Location Privacy via Differential Private Perturbation of Cloaking Area

Hoa Ngo and Jong Kim
Dept. of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH)
Pohang, Republic of Korea
Email: {hoanx, jkim}@postech.ac.kr

Abstract—The increasing use of mobile devices has triggered the development of location-based services (LBS). By providing location information to LBS, mobile users can enjoy a variety of useful applications utilizing location information, but might suffer the troubles of private information leakage. Location information of mobile users needs to be kept secret while maintaining utility to achieve desirable service quality. Existing location privacy enhancing techniques based on $K$-anonymity and Hilbert-curve cloaking area generation showed advantages in privacy protection and service quality but disadvantages due to the generation of large cloaking areas that makes query processing and communication less effective. In this paper we propose a novel location privacy preserving scheme that leverages some differential privacy based notions and mechanisms to publish the optimal size cloaking areas from multiple rotated and shifted versions of Hilbert curve. With experimental results, we show that our scheme significantly reduces the average size of cloaking areas compared to previous Hilbert curve method. We also show how to quantify adversary’s ability to perform an inference attack on user location data and how to limit adversary’s success rate under a designed threshold.

Keywords—location privacy; Hilbert curve; geo-indistinguishability; differential identifiability

I. INTRODUCTION

The development of mobile devices with embedded positioning technologies like GPS brings to mobile users various fantastical location-based applications. With a smart phone mobile users can track their routes, find nearby locations, check traffic condition or receive notifications about social events in nearby regions. Nowadays, the increasing popularity of online social networks gives even more useful location-based services like check-in, finding friends nearby or sharing places. There are a lot of location-based services and applications that have been widely used such as Foursquare, Yelp, Google Places, Loopt. In spite of the usefulness of location-based services, mobile users might suffer from some serious location privacy problems since they have to provide their location to location based services (LBS). The location information can be used to explore users’ interest places, medical issues, political views or to send spam messages.

Many approaches were proposed to protect location privacy. Policy-based effort which constrains LBS to only collect and share user location data with user one time consent was proposed [1]. However to better protect location privacy in LBS, technical approaches are very important. Several approaches based on $K$-anonymity [2], [3], [4], [5], [6], [7], [8], [9], [10], dummy locations [11], [12], [13], location perturbation [14], [15], [16], cryptography [17], [18], [19] have been proposed to protect users’ location information. These approaches can effectively protect location privacy from vulnerability of snapshot queries where user location is shared sporadically [20]. In this scenario an attacker only infers user location by observing the current result of location privacy enhancement techniques. Most of research in this field mainly focuses on adapting $K$-anonymity privacy notion.

$K$-anonymity was first proposed for privacy protection in relational database [21]. The motivation is that even the identity information such as name or social ID was removed from dataset the user anonymity can be revealed by remapping the quasi-identifier in other datasets. The key idea of this method is to hide a user identity among at least $K$ others. The technique guarantees that at least $K$ users in the dataset have the same quasi-identifiers information. Hence the probability to infer the anonymity of each user is $1/K$ even after an attacker remaps those information. In location privacy protection $K$-anonymity technique was used to hide the true identity or location of a user among $K-1$ other users. Hilbert curve method [2] is a $K$-anonymity mechanism with the consideration of preventing inference attack. However, Hilbert curve method tends to generate a significantly large cloaking area that causes a much computing overhead on LBS.

In this paper, we aim to reduce the average size of cloaking areas created by Hilbert curve method. We are motivated by fast cloaking area generation time and the ability of preventing inference attack of Hilbert curve method. To significantly solve problem of large generating cloaking area we create more than one versions of Hilbert curve by rotating and shifting the original one. After that we utilize the differential privacy [22], [23] geo-indistinguishability [15] privacy notion and a differential private mechanism to publish the optimal size cloaking area made from those multiple curves. Hence we can reduce the average size of cloaking areas while protecting the true user’s location hiding in the optimal area by differential privacy. We also quantify how much attackers know from observing cloaking result and show how to limit their probability of violating user location privacy under a desirable threshold. In short, the main contributions of our work are:

1) We release a privacy notion - area differential privacy (ADP) that utilizes differential privacy, geo-indistinguishability to protect privacy of cloaking areas perturbation mechanism.
2) We propose a mechanism to perturb the optimal

© 2015, Hoa Ngo. Under license to IEEE.
DOI 10.1109/CSF.2015.12
cloaking area made from multiple versions of Hilbert curve that guarantees ADP.

3) We quantify the probability that an attacker can learn about a user location from observing cloaking result of our perturbation mechanism and force this probability under a threshold. To guarantee this threshold, we compute the privacy parameter by area differential identifiability (ADI) - a differential identifiability [24] based mechanism.

4) Our method generates significantly smaller average size of cloaking areas compared to Hilbert curve method under different settings of ADP and ADI. It also guarantees that by observing the current cloaking result, attacker cannot absolutely conclude about the user location.

The rest of the paper is as follows: Section II surveys some related work and discusses their limitations, Section III briefly introduces backgrounds of Hilbert curve method and some privacy notions, Section IV presents the method to create cloaking areas with multiple versions of Hilbert curve, Section V is our proposed differential private approach, Section VI quantifies attacker knowledge by observing the cloaking result, Section VII shows the results of our experiments and Section VIII includes some discussions related to our work and Section IX concludes the paper.

II. RELATED WORK

Location privacy of mobile user when updating location to LBS can be protected by methods based on K-anonymity, dummy locations, location perturbation or cryptography.

A. K-anonymity and Dummy Locations Method

K-anonymity based schemes collect K-1 nearby users around a query issuer and creates a cloaking area encompassing these K users. The created cloaking area is sent to LBS as a query parameter instead of a single exact location. These schemes must guarantee that an attacker cannot determine the query issuer’s location with probability over 1/K. The cloaking area generator could be a trusted server [2], [3], [4], [5], [6] or a distributed system [7], [8], [9], [10]. K-anonymity scheme is vulnerable to the attacks to the trusted server or by misbehaviors of a peer node in the distributed system.

In the trusted server approach, there is a trusted anonymization server standing between mobile user and LBS (Fig. 1). The trusted server receives the exact location of the user and the privacy requirements (e.g. privacy level or service quality) via a secure connection, creates a cloaking area using cloaking algorithm and sends this area to LBS with the query. Since the trusted server knows all locations of mobile users, it can be a most-wanted target to be attacked. To secure K-anonymity schemes from this weakness, some distributed protocols have been proposed. In those protocols K-anonymity notion with cloaking area is guaranteed by the collaboration of mobile users without any third party.

Because almost all previous K nearest neighbors (KNN) based cloaking schemes are vulnerable for center of anonymizing spatial region (center-of-ASR) or inference attacks [2], Hilbert curve scheme [2] was proposed to completely guarantee K-anonymity. Hilbert curve scheme creates K-anonymity cloaking area by bucketing the sequence of Hilbert curve 1D indexes of 2D points in data space. It can effectively protect location data from inference attacks, has low probability of suffering from center-of-ASR attack and fast cloaking area generation time.

One variation of K-anonymity is adding dummy locations to a real location to form K users group. The main idea of dummy locations schemes [11], [12], [13] is sending some fake locations along with the real location as the query parameters to the LBS. These schemes must guarantee that LBS cannot determine the real location among the fake ones. These schemes might not require any trusted third party. In spite of the same idea of hiding the real location among others, the difference between dummy and K-anonymity schemes is that the dummy locations can be the real user locations or the random fake ones.

B. Location Perturbation

Location perturbation schemes were proposed to protect the real location of mobile users. The main idea in [14], [15], [16] is perturbing the real location to an obfuscated location. This location is sent to LBS as query parameter. The optimal strategy [14] is a zero sum Bayesian game where mobile user and adversary play in turn to maximize their benefit. This approach works based on some assumptions about adversary’s prior knowledge. Adversary knows the history of user’s LBS requesting locations and quantifies the user location privacy by a Bayesian formula. Adversary observes an obfuscated location and chooses a location to minimize the user location privacy while the user chooses appropriate obfuscated locations to maximize this value. Without any assumption about adversary’s prior knowledge, geo-indistinguishability [15], [16] uses differential privacy to perturb real user location. The probability of producing a specific obfuscated location decreases by an exponential function of its distance to the real location and the privacy parameter. Geo-indistinguishability performs the best privacy protection among all methods which do not consider adversary’s prior knowledge.

C. Cryptography

Schemes based on cryptography mostly use private information retrieval (PIR) protocol to query from LBS with hiding the exact location. These schemes [17], [18] do not require any trusted server. Location data are encrypted before being sent to LBS. With PIR based protocols LBS can execute a location query directly on the encrypted data and send back an encrypted result. The mobile user decrypts the encrypted result to get the true answer. However, PIR-based schemes only secure the privacy of user location. The scheme in [19]
uses both oblivious transfer based protocol and PIR protocol to
guarantee privacy for both user location and LBS location data. It
guarantees that LBS cannot infer the user location and the
user cannot infer more data in LBS than what he is allowed.

D. Limitations of Previous Mechanisms

Location perturbation schemes are based on obfuscating the
real user location. Hence they are safe until they have to
create a bounding area to perform the query execution process.
Because the user cannot control the number of locations in this
area it is likely to disclose the real location if there are very few
locations in there. Removing real location from the bounding
area [15] can partially solve this problem but it degrades the
service quality. In addition, the location perturbation alone
might make LBS cannot provide exact answers for location
queries. The cryptography schemes provide very secure pro-
tocols to protect user location with some strong assumptions
about encryption and decryption of messages. However they
require high computation power and cause networking over-
heads to perform cryptography-based secure protocols. For
example, PIR protocol in [19] takes significantly long time
to initialize a query and the response. This drawback makes
those methods impractical to be implemented in handheld
devices. The dummy location schemes cause both processing
and communication overheads to generate dummies on mobile
devices.

The $K$-anonymity based methods can reduce the computa-
tional cost for users’ mobile devices; users can easily control
the privacy degree; the cloaking area generation time is very
fast. However IC [4], Casper [5], PrivacyGrid [6] and NNC
[2] suffer from inference attacks because they do not have
reciprocity property [2]. This attack is successful if the users
associated with a certain locations in a cloaking area yield
different cloaking areas when applying the same cloaking
algorithm. When an attacker observes the reported cloaking
area for a long time, he can uniquely determine the exact real
location. Hilbert curve scheme [2] can prevent the inference
attack, however it generates significantly large cloaking areas
that causes computational overheads on LBS and communica-
tion overheads for networking. Therefore reducing the size of
cloaking areas in Hilbert curve scheme is needed for practical
use.

Um et. al [3] proposed a scheme that reduces the size of
cloaking areas made of Hilbert curve scheme using a data
structure to store needed information of all the grid cells in data
space. The scheme considers to extend cloaking area to the
nearest unconnected cells along with connected cells in Hilbert
curve direction. A priority queue is used to firstly consider the
unconnected cells with higher number of users or closer Hilbert
values with the query cell. By doing this way, the scheme can
reduce the average size of cloaking areas up to 40% compared
to the original Hilbert method. However, this method again
cannot protect the real location from inference attack.

Our goal is to significantly reduce the average size of
cloaking areas made by Hilbert curve method and also to
effectively prevent adversary from performing inference attack
on our method. Hence adversary should be prevented from
inferring the real user’s location by just observing the cloaking
results. By this way, we can solve the critical problems of the
previous cloaking area schemes.

III. Backgrounds

In this section we present Hilbert curve cloaking area
construction scheme, definitions of differential privacy, geo-
indistinguishability and differential identifiability.

A. Cloaking Area Construction with Hilbert Curve

Cloaking area construction with Hilbert curve [2] has a
relatively small computation time on the area generation in
anonymity server and has an ability to prevent inference attack.
Hilbert curve scheme simplifies the cloaking area construction
in anonymization server by 1D indexing of 2D data. However
due to the dimensional reduction this scheme tends to generate
a large cloaking area to preserve a locality. The large size
of cloaking area results in computation overhead in LBS and
communication overhead in network. In LBS the disk-based
data structure is usually used to store large amount of data so
that it spends much time for I/O operations. A large cloaking
area means that LBS needs to spend more time to calculate
on it. It also means that there are many possible candidates
for a location query like nearest neighbors query or range
query. This large number of candidates is a big burden for
networking communication. Therefore an efficient approach
that can maintain the advantages of Hilbert curve scheme but
generates a smaller cloaking area is needed.

Fig. 2. Original Hilbert curve and cloaking areas

The Hilbert curve scheme [2] generates cloaking areas
by sorting and splitting Hilbert values of all locations into
buckets of $K$ users (or locations). In Fig. 2 there are eight
users associated with their locations that are divided into two
cloaking areas with privacy order of 3-anonymity. The first
cloaking area of $U_1$, $U_2$, $U_3$ covers 16 cells in the
$8 \times 8$-cells data space, the second cloaking area covers 49 cells
of the other five users; the average size of the two cloaking
areas is 32.5 cells. In the original Hilbert curve some adjacent
cells in the data space are not connected together. Instead
a certain cell could be associated with a far neighbor cell
following the curve direction. It is clear that if $U_2$ connects
with $U_8$, $U_7$, $U_6$, and $U_5$, which means $U_1$ connects with
$U_3$ and $U_4$, the average size of the two cloaking areas will
decrease significantly compared to the original case (19 cells
vs. 32.5 cells). Previous work showed that even Hilbert curve
is one of the best locality preserving scheme while mapping
multidimensional data points into 1D data points [25], it causes
the increase of cloaking areas more than 50% compared to NNC scheme [2].

B. Differential Privacy

Differential privacy was first introduced to protect individual data in statistical databases. Differential privacy guarantees that the outcome of a data analysis cannot be changed significantly by the presence of any specific individual. Hence without any consideration of adversary knowledge differential privacy can prevent the presence of any individual in database from the inference attack when the adversary is able to observe query outcomes. For the case of numerical outcome, the most popular mechanism to guarantee differential privacy is to add controlled random noise drawn from Laplace distribution to the query output [22]. The amount of noise is linked to the sensitivity of the query function. If the sensitivity is high more noise needs to be added to provide the same level of privacy defined by a given parameter $\epsilon$.

Definition 1. (Differential Privacy [22]). For any two data sets $D_1$ and $D_2$ differing at most one element, a randomized function $K$ provides $\epsilon$-differential privacy if for all $S \subseteq \text{Range}(K)$,

$$\Pr(K(D_1) \in S) \leq e^\epsilon \Pr(K(D_2) \in S)$$

Differential privacy now becomes a gold standard from its birth to protect individual privacy in tabular database. Recently, applying differential privacy to protect location privacy is the target of some valuable research [15], [16], [26], [27]. Most of them were proposed for location perturbation.

C. Geo-Indistinguishability

Geo-indistinguishability motivated from [28] is a generalized variant of differential privacy with arbitrary metric between secrets [15]. To protect location privacy in location perturbation scheme geo-indistinguishability constrains the probability of transformation from one location to another with a differential privacy like formula. Adversary can observe the obfuscated location but cannot determine the real location with high accuracy depending on privacy parameter. The basic mechanism to gain geo-indistinguishability is to add 2D Laplace noise directly to the location data, but an optimal mechanism [16] has been proposed that can provide better location privacy given the same privacy parameter.

Definition 2. (Geo-Indistinguishability [15]). For any two locations $x$ and $x'$ belong to an input location set $X$, given a reported location $z$ belongs to an output location set $Z$, a mechanism $K$ provides $\epsilon$-geo-indistinguishability if

$$K(x)(z) \leq e^{\epsilon d(x,x')} K(x')(z), \text{ where } d(\cdot, \cdot) \text{ is a metric to measure distance between two locations such as Euclidean or Hamming distance metric and } K(x)(z) \text{ is the probability that } z \text{ is the perturbed output of } x \text{ through a mechanism } K.$$  

D. Differential Identifiability

Differential identifiability proposes a privacy policy to quantify the privacy parameter $\epsilon$ of differential privacy against the identifiability risk [24]. An adversary in this case knows the domain of all elements in the dataset. He also knows all values in a dataset but one as the assumption of differential privacy. Because the adversary knows the input domain he can rebuild all possible worlds, each world contains all known records of the dataset and a possible value of the missing element. After observing the query outcome the adversary computes the probability that this outcome comes from each possible world. He compares the probabilities and concludes the world that includes the missing value. Therefore he can determine the missing value of the dataset with some probability. To limit the success probability of the adversary, differential identifiability requires the difference in probabilities that a result comes from all different possible worlds which differ at most one element is smaller than a threshold $\rho$.

Definition 3. (Differential Identifiability [24]). Given a database $D$ with records drawn from the universe $U$, each record corresponding to the identity of an individual and a query function $f$ releases result $r$ on $D$. $I(i)$ is the record $i$ of a database, $I_D$ is the set of records in $D$, $I_D = \{I(i)| i \in D\}$. A randomized mechanism $M$ satisfies $\rho$-differential identifiability if for $\forall D' = D - i^*$ and $\forall i \in U - D'$:

$$\Pr(I(i) \in I_D | M(D) = r, D') \leq \rho$$

The threshold $\rho$ constrains the probability that adversary infers the presence of an individual in $D$ by comparing the probabilities of this individual in all possible world. It guarantees that a strong adversary who knows the domain of database records and also all the records in a database but one cannot infer the missing value with a probability higher than $\rho$. To achieve differential identifiability the privacy parameter $\epsilon$ must be smaller than a specific value.

IV. MULTIPLE HILBERT CURVES AND CLOAKING AREA

In this section we explain the way to generate cloaking areas with multiple versions of Hilbert curve including the rotated and shifted versions.

A. Rotated and Shifted Hilbert Curves

By rotating the original Hilbert curve 90° clockwise and counterclockwise and rotating 180° around the center point of the data space we have four versions of rotated curve. Since the original version of Hilbert curve can be shifted to every different directions, in this paper we only consider four versions of shifted curve following top-left, top-right, bottom-left, bottom-right directions. Rotated and shifted versions of Hilbert curve bring different connections among grid cells in comparison with the original version. The work in [29] leverages this characteristic to perform more efficient range query search.

A spatial range query is used to identify certain objects in a given area such as "find the nearest gas station at most 300 m far away from my location or find all hospitals in the city". To perform a range query with the Hilbert curve indexing, LBS searches all objects that have index values covered by the query window. However Hilbert curve cannot preserve closeness of nearby positions while indexing 2D data into 1D. So, Hilbert curve may separate data space into some discrete clusters for the given query window. Rotated and shifted versions of Hilbert curve can reduce the number of clusters in a certain query window so the I/O time to access data in the data structure like B+ tree would be reduced significantly. If the
query window covers only one cluster, data in B+ tree can be accessed consecutively. This will save much I/O time to travel on the tree especially in case of using a disk-based data structure for big data analysis in LBS.

**B. Cloaking Areas Construction with Multiple Hilbert Curves**

Rotated and shifted versions of Hilbert curve might reconnect the close disconnected grid cells in the original version together. That opens a chance to reduce size of cloaking areas with a certain privacy parameter $K$. For example in Fig. 4 we create another version of Hilbert curve by shifting the original to the bottom left direction. This version generates two cloaking areas with smaller size than that in the original version. It connects $U_1$, $U_3$, $U_4$ together while those close points are disconnected in the original version. Fig. 3 shows another version of Hilbert curve by rotating the original one by $90^\circ$ clockwise around the center point of the data space (note that the starting point of the curve now is in the top left corner of the data space). Two cloaking areas are created, the second area is even smaller than that in the bottom left shifted version.

Nonetheless a certain version of curve can provide smaller cloaking areas for certain number of points than that in the original version, it also can increase the size of the others. For example, rotated and shifted versions in Fig. 3 and Fig. 4 reduce the size of cloaking areas of $U_4$, $U_6$, $U_7$ or $U_8$. However, they increase the size of cloaking area of $U_2$. The shifted versions can disconnect some points from Hilbert curve. So that if a user requests a query from $U_8$ the bottom-left version cannot return a cloaking area. To solve the challenge of disconnected points in shifted versions, we could assign the cloaking areas of the original version to those points.

Intuitively the simplest scheme to find the best cloaking result for a given location is to firstly find all cloaking areas made by different Hilbert curve versions and after that choose the smallest one. By this way we can find a close optimal cloaking area which is similar to the result of a KNN based method. We will show this in the evaluation section. However the scheme might be vulnerable to the inference attack because different points in a certain cloaking area may give different optimal areas. For example, $U_8$ chooses the cloaking area $c$ from $90^\circ$ clockwise rotated Hilbert curve, $c \equiv (U_8, U_2, U_3, U_4, U_1)$, as the optimal cloaking area with the smallest size. Adversary observes $c$ and concludes that this area is generated by $U_8$ because all other points in $c$ do not take $c$ as the smallest area. To reduce the average size of cloaking areas while protecting the real point hiding in the optimal one, we perturb the smallest cloaking area with satisfying a geo-indistinguishability based privacy notion.

**V. AREA DIFFERENTIAL PRIVACY**

In this section we demonstrate how to publish efficiently cloaking areas. We show our geo-indistinguishability [15] based privacy notion, the mechanism to publish the optimal size cloaking area and security analysis of this mechanism.

**A. Definition**

Given a set of cloaking areas $C$, the possible reported cloaking areas are in the set $Z$. We denote the probability to transform a cloaking area $c \in C$ to a cloaking area $z \in Z$ is $K(c)(z)$ where $K$ is a randomized mechanism.

**Definition 4.** (Area Differential Privacy (ADP)). Given any two cloaking areas $c, c' \in C$ and a reported cloaking area $z \in Z$, a randomized mechanism $K$ satisfies $\epsilon$-area differential privacy if,

$$K(c)(z) \leq e^{\epsilon d(c, c')}K(c')(z),$$

where $d(\cdot, \cdot)$ is a metric to measure the distance between two cloaking areas.

The $d(\cdot, \cdot)$ can be the difference in size between two areas, Euclidean distance between two area centers, etc. To efficiently reduce the average size of Hilbert curve cloaking areas we design a metric showing in the publishing mechanism section.

ADP constrains the probabilities $K(c)(z)$ and $K(c')(z)$ with a geo-indistinguishability like privacy notion. The difference is that geo-indistinguishability definition is for location points and ADP is for cloaking areas. From the characteristics of geo-indistinguishability we can state that by observing the reported cloaking area an adversary may only have a limited increase in his knowledge about the real cloaking area. The adversary can absolutely know the set of possible input cloaking areas and he can calculate all probabilities of the perturbation. However he cannot conclude the real cloaking area with high
accuracy. The knowledge the adversary gains after observing the reported area depends on the privacy parameter $\epsilon$. This privacy parameter controls the trade-off between privacy and utility in all differential privacy based approaches.

### B. Optimal Cloaking Area Publishing Mechanism

By using multiple versions of Hilbert curve we get multiple cloaking areas for a given location point. To preserve privacy and significantly reduce the size of cloaking area for a location we perturb the smallest cloaking area in a set of areas generated from multiple versions of Hilbert curve. The perturbation mechanism must satisfy ADP definition. To perform the perturbation mechanism we firstly build a score function to assign score for each output. We assume that the data space is divided into $m \times m$ cells to draw Hilbert curves. We give score for each cloaking area depending on its size that is the number of cells it covers in data space.

**Definition 5. (Score Function).** Given a set $D$ of $n$ cloaking areas of a location generated from $n$ versions of Hilbert curve, the cloaking areas in $D$ are numbered in the ascending order of their sizes, the one with smallest size is $c_1$ and that with biggest size is $c_n$. The score function $f : D \rightarrow \mathbb{R}$ will give a $c_k \in D$ the score $q(D, c_k)$ as

$$f(c_k) = q(D, c_k) = k - 1.$$ $\blacksquare$

For example let us assume that $D$ contains three cloaking areas $(c_1, c_2, c_3)$ in the ascending order of their sizes. We have $q(D, c_1) = 0, q(D, c_2) = 1, q(D, c_3) = 2$. The score $q(D, c_k)$ describes the sequence distance from the cloaking area $c_k \in D$ to the smallest one in $D$. The bigger size $c_k$ is the higher score it gets in $D$. The score function is simple but it can keep the maximum score distance of every set $D$ of every location constant. The real size distance of $c_k$ and the smallest one can be used but we cannot keep the same maximum distance for every set $D$ that hurts our ability to quantify and limit adversary’s inference attack later. By this score function we build the mechanism that gives higher probabilities for smaller cloaking areas in a set of cloaking areas.

**Definition 6. (Probability Function).** Given the privacy parameter $\epsilon$, the set $D$ of cloaking areas of a location point created by multiple versions of Hilbert curve, an output cloaking area $c \in D$, the probability function $g : D \rightarrow \mathbb{R}$ will give $c$ the probability $Pr(D, c)$ as

$$g(c) = Pr(D, c) = \frac{e^{-\epsilon q(D, c)}}{\sum_{c' \in D} e^{-\epsilon q(D, c')}}$$

where $q(D, c)$ is the score of $c$ in the set $D$. $\blacksquare$

Intuitively this probability function gives higher probabilities to smaller cloaking areas to make them the reported areas. This characteristic brings a chance to reduce the average size of cloaking areas generated from multiple versions of Hilbert curve method. Procedure 1 shows the steps to generate the optimal cloaking area with ADP.

Any mechanism following Procedure 1 perturbs the optimal cloaking area of a location that satisfies the ADP definition. To prove that we define a metric $d(.)$ to measure the distance between two cloaking areas, for examples $c$ and $c'$, $c$ belongs to the set $D$ of cloaking areas generated from multiple Hilbert curve versions of a specific location and $c'$ belongs to the set $D'$ of another location. $D$ and $D'$ must intersect in at least one common cloaking area. We define the distance between two cloaking areas $c$ and $c'$ belonging to $D$ as $d(c, c') = |q(D, c) - q(D, c')|$. The distance between $c$ and $c'$ is computed by Definition 7.

**Definition 7. (Cloaking Areas Distance Metric).** Given a reported cloaking area $c_r$, define $d : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ for two cloaking areas $c$ and $c'$, $D$ is a set of cloaking areas generated from multiple versions of Hilbert curve of a location; $D'$ is the set of cloaking areas of another point. $D$ contains $c_r$ and $c$, $c'$ contains $c_{r'}$ and $c'$, i.e.,

$$d(c, c') = |d(c, c_r) - d(c', c_r')|$$ $\blacksquare$

Note that adversary only observes the reported cloaking area. Let us assume that the adversary observes the reported cloaking area $c_r$ containing $k$ locations. He calculates a set $S$ containing $k$ optimal cloaking areas corresponding to $k$ locations. For any two optimal cloaking areas $c_{op}$ and $c'_{op}$ corresponding to two locations $p$ and $p'$, $c_r$ is published by a mechanism $K$ following Procedure 1. $D$ is the set of cloaking areas generated from multiple Hilbert curves of $p$, $D'$ is the set of cloaking areas of $p'$, we have

$$\frac{K(c_{op})(c_r)}{K(c'_{op})(c_r)} = \frac{\sum_{c \in D} e^{-\epsilon q(D, c)}}{\sum_{c' \in D'} e^{-\epsilon q(D', c')}}$$

(1)

Because

$$\sum_{c \in D} e^{-\epsilon q(D, c)} = \sum_{c' \in D'} e^{-\epsilon q(D', c')},$$

due to the score function, we have,

$$\frac{K(c_{op})(c_r)}{K(c'_{op})(c_r)} = \frac{e^{-\epsilon q(D, c_r)}}{e^{-\epsilon q(D', c_r)}}$$

(2)

Because $q(D, c_{op}) = q(D', c'_{op}) = 0$, we can deduce that

$$\frac{K(c_{op})(c_r)}{K(c'_{op})(c_r)} \leq e^{\epsilon d(c_{op}, c_r) - d(c'_{op}, c_{r'})} = e^{\epsilon d(c_{op}, c_r) - d(c'_{op}, c_{r'})}$$

(4)

$$e^{\epsilon d(c_{op}, c_r) - d(c'_{op}, c_{r'})}$$

(5)

$$e^{\epsilon d(c_{op}, c_r)}$$

(6)
After all, (6) indicates that with the metric in Definition 7, Procedure 1 perturbs the optimal cloaking area of a location point that satisfies the ADP definition.

C. Security Analysis

The critical drawback of effort in [3] is that it cannot protect the user location privacy when adversary performs the algorithm again and gets different results for different locations. Therefore by only observing the cloaking area from a snapshot query adversary can determine exactly the real location. The advantage of our method is that if a cloaking area contains two different locations $a$ and $b$, it can be the result of cloaking area perturbation process from either $a$ or $b$. Adversary observes the cloaking result and computes the probability that the result was generated from each location in this area but he cannot definitely conclude the query point. However it does not mean that the perturbation process can prevent the information leakage perfectly. Because adversary can compute the probabilities, he can compare them to perform an inference attack. Note that our scheme confuses adversary about the real location however it might not guarantee the $K$-anonymity. $K$-anonymity notion forces all $K$ members in a cluster to have the same $1/K$ probability of being disclosed the identity. However, in our perturbation method the probability distribution is not identical, given a cloaking area containing locations $a$ and $b$ adversary can estimate $p_a$ and $p_b$ that are the probabilities that $a$ and $b$ yield the query respectively. If $p_a \gg p_b$ the adversary can conclude with high probability that $a$ is the real location of querying user.

This attack shows that our scheme is hard to fully achieve $K$-anonymity. We can get close to $K$-anonymity by decreasing the privacy parameter $\epsilon$ to be very small so that all the cloning areas from different versions of Hilbert curve have very close probabilities. However $\epsilon$ degradation also increases the average size of cloning areas because the perturbation results could be far away from the optimal value. Hence we need to set the privacy parameter $\epsilon$ to protect location privacy well from inference attack while remaining utility for the differential scheme.

VI. Area Differential Identifiability

We now discuss how to prevent adversary from performing an effective inference attack on our differential privacy based method. To do that we need a privacy notion and mechanism to set the value of parameter $\epsilon$ so that we can control the success rate of the adversary. We use a differential identifiability [24] based mechanism to bound the adversary success rate. We firstly define a privacy definition.

A. Definition

The adversary observes the reported cloaking area and we assume that he observes cloaking area $c$ of $K$ location points. He starts the attack with the knowledge of $K$ different sets $D_0, D_1, \ldots, D_{K-1}$ of cloning areas of $K$ different points in $c$. Every set is generated from the same versions of Hilbert curve.

**Definition 8. (Area Differential Identifiability).** A randomized mechanism $M$ on a function $f$ publishes the optimal cloning area of a location following Procedure 1 is said to satisfy $\rho$-area differential identifiability (ADI) if the inference attack success probability is below $\rho$ even when an adversary knows $K$ different sets of cloning areas generated from $K$ points in $c$. I.e., $\forall i, 0 \leq i < K, \forall D \in \{D_0, \ldots, D_{K-1}\}$,

$$\Pr(c \in D_i | M_f(D) = c, D_0, D_1, \ldots, D_{K-1}) \leq \rho.$$  

ADI guarantees that the probability a cloning result belonging to the set of cloning areas of any specific point in this result must be lower than the threshold $\rho$. Hence by only comparing probabilities adversary cannot identify the real location with higher probability than $\rho$. The threshold $\rho$ in ADI is similar to the one in differential identifiability. It bounds the confidence of adversary about the world containing a specific element. The difference is that in differential identifiability attacker wants to identify the missing value in a database among all possible values while in ADI attacker wants to identify the real location among $K$ points in a cloning area which publishes this area by Procedure 1. Differential identifiability only considers two adjacent databases so all the possible worlds differ exactly one element while in our method the adversary recovers all possible worlds that might differ more than one cloning areas.

Similar to differential identifiability, ADI is proposed to protect a differential privacy based method from an inference attack where an attacker observes the query output and computes probabilities of all possible input candidates. The question is how to choose $\epsilon$ that is enough to provide an identifiability level $\rho$. The focus of the differential private method is shifted to controlling the parameter $\epsilon$. However the
value of \( \rho \) that guarantees robustness against any inference attacks is an open question.

B. Mechanism

We now show how to adjust the privacy parameter \( \epsilon \) to achieve ADI. There exists an upper bound of \( \epsilon \) where we achieve ADI and guarantee a privacy level, therefore we cannot increase \( \epsilon \) to very high value to achieve more utility. To find this upper bound, we assume that different versions of Hilbert curve yield different cloaking areas for a specific location.

Let \( P(c) \) be the probability an adversary believes that the cloaking result \( c \) belongs to a set of cloaking areas \( D_i \) created from multiple Hilbert curves of a point \( i \). Cloaking area \( c \) contains \( K \) points, each point generates a set of cloaking areas from multiple versions of Hilbert curve. Intuitively the adversary knows all those sets \( D_0, D_1, \ldots, D_{K-1} \). We illustrate the way to find the upper bound of \( \epsilon \) given \( \rho \) that is similar to the differential identifiability work [24]. However our work is for discrete domain. Without loss of generality we assume that adversary concerns to the set \( D_0 \). With \( \forall D \in \{D_0, D_1, \ldots, D_{K-1}\} \) we have the probability of the adversary’s confidence \( P(c) \) as follow:

\[
P(c) = \frac{\sum_{0 \leq i < K} Pr(D_i, c)}{Pr(D_0, c)}
\]

We apply (14) and (15) to (11)

\[
P(c) \leq \frac{1}{1 + \sum_{1 \leq i < K} (e^{-\epsilon S} \times 1)}
\]

We have to find \( \epsilon \) that

\[
\frac{1}{1 + (K-1)e^{-cS}} \leq \frac{1}{\rho(K-1)}
\]

\[
\epsilon S \geq \frac{1 - \rho}{\rho(K-1)}
\]

\[
\epsilon \geq \frac{1}{K}
\]
TABLE I. CLOAKING AREA METHODS WITH SOME CHARACTERISTICS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Guaranteed Privacy Notion</th>
<th>Mechanism</th>
<th>Vulnerability to Inference Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNC</td>
<td>K-anonymity</td>
<td>KNN</td>
<td>Yes</td>
</tr>
<tr>
<td>Smallest HC</td>
<td>K-anonymity, ADP</td>
<td>Procedure 1; $\epsilon \rightarrow +\infty$</td>
<td>Yes</td>
</tr>
<tr>
<td>HCs and DP</td>
<td>K-anonymity, ADP</td>
<td>Procedure 1; Vary $\epsilon$</td>
<td>Depend on $\epsilon$</td>
</tr>
<tr>
<td>Original HC</td>
<td>K-anonymity</td>
<td>Hilbert Curve</td>
<td>No</td>
</tr>
</tbody>
</table>

Theorem 1 shows the relation between $\epsilon$ and $\rho$, when $\rho$ decreases $\epsilon$ might also be decreased under a threshold. To achieve more privacy, we reduce the value of $\rho$. When $\rho$ is very close to $1/K$, all cloaking areas have very close probabilities. However in this case the average size of cloaking areas might be increased so much that can destroy the utility of our method. We can denote parameter $\rho$ as a trade-off between utility and privacy of our cloaking areas publishing mechanism. Low value of $\rho$ means high privacy and low utility and high value of $\rho$ means low privacy but high utility.

VII. EVALUATION

To perform the evaluation we used a real dataset of California points of interest [30] that contains 104,770 locations in California. We implemented our algorithm in C++ and executed in Intel Core Quad 2.83-GHz CPU, 8-GB RAM and Linux OS. In all experiments we considered 8 versions of Hilbert curve including the original, negative 90°, positive 90°, 180° rotated, top-left, top-right, bottom-left and bottom-right shifted version. For experiments we used the trusted server architecture in Fig. 1, which has an anonymization server between mobile users and LBS. With a specific Hilbert curve version, we make a red black binary search tree to store coordinates of locations and their Hilbert values. To efficiently compute the rank of a location by its Hilbert value we use an order statistic tree. When a user sends a query to anonymization server the server computes Hilbert value of the location and finds the location’s rank in the order statistic tree. After finding the rank, it finds the bucket containing the location in the red black tree and denotes this bucket as a cloaking area. By applying those data structures all operations would be executed in $O(log N)$ that will be helpful when we provide services to a large number of mobile users. We also implemented NNC algorithm [2] to make some comparisons. In NNC method a location firstly finds its $K−1$ nearest neighbors and makes the set $N$. After that it randomly chooses a location in $N$ and finds $K−1$ nearest neighbors of this location. Those $K−1$ neighbors are grouped in the set $N'$. Finally, the real location is grouped with $N'$ to make a cloaking area. NNC is based on KNN search that can be performed efficiently by a R* tree. We used an open source of in memory R* tree in [31]. The performance of our algorithm is measured in CPU time of a cloaking area construction in anonymization server and the average size of cloaking areas.

We compare the average size of cloaking areas of four methods (NNC, Smallest HC, HCs and DP, and Original HC) in Figs. 5, 6, and 7. NNC method uses a KNN based mechanism to generate cloaking areas. "Smallest HC" makes a cloaking area by choosing the smallest area among those made by multiple versions of Hilbert curve. Smallest HC publishes cloaking areas by Procedure 1 with $\epsilon$ comes to infinite. HCs and DP follows Procedure 1 with a specific value of $\epsilon$. Finally "Original HC" is the Hilbert curve method [2] with only one formal version of curve. A quick comparison in some characteristics of those methods are showed in TABLE I.

![Fig. 5. Average size of cloaking areas (% of data space). Varying test, $N = 104,770$, $K = 80$, $Q = 2,000$, $\epsilon = 0.01$.](image)

![Fig. 6. Average size of cloaking areas (% of data space). Varying test, $N = 104,770$, $K = 80$, $Q = 2,000$, $\rho = 1/50$.](image)

![Fig. 7. Average size of cloaking areas of the first test (% of data space). Varying $K$, $N = 104,770$, $Q = 2,000$, $\epsilon = 0.01$.](image)

In the experiments which resulted in Figs. 5 and 6, we generated 2,000 queries from random locations. The number of locations in a cloaking area is 80; we measured the average size of cloaking areas in the percentage it covers in data space. This average size is computed by $S_A = \frac{Q}{Q} \sum_{i=1}^{M} \sum_{j=1}^{p_{ij}} x_{ij}$, where $Q$ is the number of queries, $M$ is the number of Hilbert curve
versions, $S_{ij}$ is the size of cloaking area generated from query $i$ and version $j$ of Hilbert curve, $p_{ij}$ is the probability of the cloaking area with size $S_{ij}$ and $\sum_{j=1}^{M} p_{ij} = 1, \forall i$. The results vary by tests so we generated ten tests for the experiments. Fig. 5 shows that when differential privacy parameter $\epsilon$ equals to 0.01, a quite small value, the average size of cloaking area generated by our method, HCs and DP is smaller than one generated by the original version of Hilbert curve method. Original HC up to 36%. If we only choose the smallest area from multiple versions of Hilbert curve the results are very close to NNC method. In Fig. 6, the experiment sets the ADI threshold $\rho$ equals to 1/50. The threshold is very small and the average size increases significantly. However cloaking results still remain better than the original version up to 33%. However, in some tests the results are only better than the original version about 16%.

In Fig. 7, we use the first test that is used in experiments of Figs. 5 and 6. Fig. 7 describes the average size of a cloaking area when we vary the number of users in a cloaking area of a location in the first test with differential privacy parameter $\epsilon$ equals to 0.01. Our method shows advantage in comparison with the original version that the average size is smaller from 10% to 23%. Note that the privacy parameter $\epsilon$ controls result of our method, when we increase $\epsilon$ we get better result; in the optimal case it would be close to cloaking result of NNC method.

Fig. 8 shows the comparison between our method and the original Hilbert curve method when $\epsilon$ and $K$ are varied. The percentage of average size degradation is computed by $p = \frac{100 \times \sum_{i=1}^{Q} (S_i - \sum_{j=1}^{M} p_{ij} \times S_{ij})}{\sum_{i=1}^{Q} S_i}$, where $Q$ is the number of queries, $M$ is the number of Hilbert curve versions, $S_{ij}$ is the size of cloaking area generated from query $i$ and version $j$ of Hilbert curve, $S_i$ is the size of cloaking area generated from query $i$ and the original version of Hilbert curve, $p_{ij}$ is the probability of the cloaking area with size $S_{ij}$ and $\sum_{j=1}^{M} p_{ij} = 1, \forall i$. In all cases, the average size of cloaking areas created by our method is smaller than that in the original Hilbert curve method. The utility increases when $\epsilon$ increases. When $\epsilon = 1$ the cloaking results are close to NNC method.

In Fig. 9, we only choose $K=80$ to perform the experiment with the smallest $\rho$ equals to 1/70 because the threshold $\rho$ has to be bigger than 1/$K$. From this figure the utility decreases when ADI threshold $\rho$ decreases. Hence the average size of cloaking areas in our method gradually increases. However it is still smaller than cloaking results of the original Hilbert curve method from 16% to 43%. Even when $\rho$ is very small (e.g., 1/70), our method still shows advantages compared to the original Hilbert curve method, it indicates that our method still remains utility while preserving privacy well.

Finally, we measured the CPU time to generate a cloaking area in our method compared to NNC method. We only measured CPU time because of using in memory data structures. In Fig. 10, even spatial data are indexed by a R* tree, NNC method lasts longer than our method to create a cloaking area because of costly KNN queries. In this experiment we measured the cloaking area generation time with a given value of $\epsilon$ or $\rho$ and we have to query 8 versions of Hilbert curve. Because each query in a different version of curve can be executed separately so the performance of our method could be increased by parallel programming.

VIII. DISCUSSION

Our work requires more computation to create a cloaking area but it is worth to reduce the cloaking areas size. The LBS usually maintains a huge amount of data and uses disk-based data structure [2], so executing a location based query with a large cloaking area is a big burden for LBS. The time to create a cloaking area in anonymization server can be measured in milliseconds (Fig. 10) but the time to answer a query in LBS is measured in seconds [2].
Our method is based on geo-indistinguishability privacy notion that has a proposed optimal mechanism [16]. The optimization of location perturbation method is a good direction for research motivated from [32]. By making an optimal mechanism with a differentially private approach we can achieve maximum utility for a given level of privacy or maximum privacy for a given level of utility while satisfying the privacy notion [16], [33]. Our procedure to achieve ADP is similar to adding Laplace noise in discrete domain that might be not the optimal mechanism. Finding the optimal one with ADP is a very promising future work.

IX. CONCLUSION

In this paper, we proposed a privacy definition that is based on geo-indistinguishability, a generalized variant of differential privacy, to publish the smallest cloaking area generated from multiple versions of Hilbert curve. Our method can both reduce the average size of cloaking areas compared to the original Hilbert curve method and guarantee location privacy against inference attack. We used the different versions of rotated or shifted Hilbert curve to reduce cloaking area size and used geo-indistinguishability to protect against inference attack. Our work is the first approach that uses differential privacy based method to publish the optimal size cloaking areas made of multiple versions of Hilbert curve. The experiments show many advantages of our method in comparison with previous methods. Our method outperforms the original Hilbert curve method by generating smaller average size of cloaking areas and outperforms the NNC method in cloaking area generation time. We believe that our work can motivate other research in applying differential privacy to protect location privacy or in applying differential privacy based mechanisms to achieve close-optimal results.

ACKNOWLEDGMENT

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1403-04.

REFERENCES