IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Location Privacy Against Long-Term Observation Attacks

Suganya[1]

[1]*Computer Science Engineering,Madha Engineering College,Tamilnadu ,India.*

Abstract *—Mechanisms built upon geo-indistinguishability render location privacy, where a user can submit obfuscated locations to Location-Based Service providers but still be able to correctly utilize services. However, these mechanisms are vulnerable under inference attacks. Particularly, with background knowledge of a user's obfuscated locations, an attacker can infer actual locations by carrying out long-term observation attacks. Unfortunately, how to defend long-term observation attacks in the field of differential location privacy remains open. In this paper, we first demonstrate the vulnerabilities of existing mechanisms under long-term observation attacks. In light of these vulnerabilities, we devise a novel mechanism, referred to as Eclipse, which bridges the gap between location protection and usability of services. Specifically, we harness geo-indistinguishability and k-anonymity to obfuscate locations and hide each location based on an anonymity set. As a result, our mechanism effectively perturbs the distribution of locations and suppresses leakage under long-term observation attacks. Moreover, the set of possible outputs is utilized to minimize the impacts to usability and correctness. We formally define and rigorously prove the security of the proposed mechanism by leveraging differential privacy. Moreover, we implement the proposed mechanism and conduct a series of experiments on real-world datasets to demonstrate its efficacy and efficiency.*

Keywords *- Location Privacy, Long-term Observation, Geo-indistinguishability, K-anonymity, Expected Inference Error.*

1. Introduction

Although Location-Based Services (LBSs) render great convenience to users, these services also raise critical privacy concerns as users' private locations could be easily revealed to the public. For instance, to search near by pharmacies on Yelp, a user has to provide her actual location. Many location privacy-preserving mechanisms (LPPMs) have been proposed to promote users' location privacy. For example, geo-indistinguishability , which is generalized based on differential privacy , can output obfuscated locations within a radius to hide a user's actual location but still releases approximate location information for desired services. Existing geo-indistinguishable mechanisms guarantee location privacy against attacks with no background or prior information. However,

recent studies have shown that, given prior information, an attacker can reveal locations protected by these mechanisms with inference attacks. For instance, investigated privacy leakage under inference attacks, where an attacker has a snapshot of one obfuscated location and prior information, which can be called short-term observation attacks in this paper. They demonstrate that the leakage under short-term attacks is severe and devise a dynamic differential location privacy mechanism with personalized error bounds named PIVE to mitigate the leakage.

Unfortunately, an attacker can defeat this newly proposed defense by performing our proposed long-term observation attacks. Long-term observation attacks indicate that a user's behavior could be gathered and stored over a period of time, and such cumulative information might be exploited by an adversary performing inference attacks to obtain some sensitive information. In our attacks, the behavior refers to the query requests sent to the server, and each request contains the user’s obfuscated location generated by existing geo-indistinguish ability based mechanisms. Once an attacker obtains a series of obfuscated locations produced by the same actual location (e.g., home), he could uncover the actual location with other prior information. Millions of users have daily routines with same locations (e.g., homes, offices, schools). An attacker can take advantage of long-term observation attacks, easily uncover sensitive locations, and further users interests and activities. It disturbs the privacy of millions of consumers at large, and ultimately thwarts Internet freedom. The consequences of this kind of attacks are severe, and effectively mitigating the attacks is challenging. Arbitrarily perturbing users locations can simply suppress the attacks, however, it would completely affect the usability and correctness of location-based services. In other words, how can we bridge the gap between privacy protection and usability of services

In this paper, with focus on solving such problem under the Point of Interests (POIs) searching scenarios (e.g., Yelp and Foursquare), we propose a novel location privacy preserving mechanism, referred to as Eclipse. We integrate geo-indistinguish ability, k-anonymity, and the expected inference error to tackle the problem. Specifically, our proposed mechanism harnesses geo-indistinguish ability and k- anonymity to obfuscate the distribution of locations and promotes privacy against the long-term observation attacks. An obfuscated location is hidden within an anonymity set chosen based on k-anonymity. Moreover, the set of possible outputs is generated by considering the restriction of Quality of Services (QOSS). As a result, obfuscated locations produced by Eclipse still provide approximate location information for desired services. In other words, our proposed mechanism strengthens privacy against long-term observation attacks with minimal impacts to the usability and correctness of LBSs. The major contributions of this paper are summarized as below:

_ We develop a long-term observation attack on existing mechanisms built upon geo-indistinguish ability. We identify the privacy limitations of these solutions under long-term observation attacks.

_ We devise a novel mechanism, named Eclipse, to advance privacy protection of a user’s location. Eclipse integrates k-anonymity, geo-indistinguish ability and the expected inference error, and overcomes the limitation in existing studies. It can effectively perturb the distribution of obfuscated locations while introducing minimal impacts to services. To the best of our knowledge, our mechanism represents the first defense against long-term observation attacks in the field of differential location privacy.

_ We formally define and rigorously prove the security of Eclipse with differential location privacy. To demonstrate the efficacy and efficiency of our mechanism, we implement Eclipse and conduct a series of experiments on two real-world datasets (including Brightkite1 dataset and Gowalla2 dataset) with millions of location check-ins. Our results suggest that the proposed scheme achieves stronger privacy than previous solutions.

2. RELATED WORKS

Anonymity. The main idea of *k*-anonymity is to hide a user’s location among a number of *k* locations. Gruster *et al.* first introduced the definition of *k*-anonymity in location privacy. Specifically, *k*-anonymity can hide a user’s location into a spatio-temporal cloaking box,which contains at least *k* users. In addition to spatio-temporal cloaking, another effective way to achieve *k*-anonymity is to apply dummy locations. Kido *et al.* implemented a random walk model to generate dummy locations. The anonymity degree of this approach could be weaken if an adversary has prior information. To address this limitation, Niu *et al.*proposed a new scheme to improve the generation of dummy locations. Liu *et al.* selected dummy locations by evaluating the spatio-temporal correlation from three aspects, including time reachability, direction similarity and in degree/out-degree. However, Zhang *et al.* argued that a significant part of users may concern about their location privacy and therefore may not be interested in participating in an anonymity set. To solve this problem, they designed a set of auction-based mechanisms and proved that these mechanisms are truthful. Some anonymity-based solutions have also been proposed in continuous LBS queries Zhao *et al.* proposed a privacy preservation against location injection attacks. In the meantime, Jiang *et al.*presented RobLoP, a robust privacy preserving algorithm against location dependent attacks. Tang *et al.* studied the long-term location privacy protection, and proposed a set of

1.https://snap.stanford.edu/data/locbrightkithtml
2.https://snap.stanford.edu/data/locgowalla.html

novel dummy trajectories generation algorithms by considering both real geographical information and long-term consistency. To resist long-term observation attack, Beresford and Stajano presented another method by changing pseudonyms for each user frequently. However, an adversary would link together all the pseudonyms attached to requests for a single user's data, when a user's preference data were stored on a server. One important technique for increasing location privacy is to obfuscate the location data, possibly by adding noise or reporting regions instead of points. Therefore, we study long-term observation attacks on those mechanisms built upon geo-indistinguishability.

Differential Privacy. The definition of differential privacy is derived from the area of statistic databases. It can preserve the privacy of each individual's tuple when publishing aggregated information of a database.The major advantage of differential privacy is that it can preserve privacy against attackers with arbitrary background knowledge. In differential privacy was first introduced in the context of location privacy. It can publish securely statistical information with regard to the commuting patterns of users without compromising individual privacy. Ho *et al.* proposed a quadtree spatial decomposition technique, which is utilized to ensure differential privacy in a database with the capability of location pattern mining. Although differential privacy can be easily applied to cases where information of multiple users is aggregated, it is not suitable for applications where only a single user is engaged. To tackle this limitation, Dewri proposed a scheme by combining differential privacy and *k*-anonymity. It also suggested that the probability of reporting the same obfuscated location from any of *k* locations shall be similar. Similarly, Zhao *et al.*proposed a privacy-preserving paradigm-driven framework for indoor localization (*P*3 -LOC), which employs specially designed *k*-anonymity and differential privacy techniques to achieve the provable privacy preservation. Moving a step forward, Andres *et al.* introduced the notion of geo-indistinguishability, which is a generalized notion of differential privacy. A Planar Laplace mechanism was developed to achieve geo-indistinguishability by adding noises to actual locations. Bordenabe *et al.* devised an optimal geo-indistinguishable mechanism to minimize the service quality loss. Then, presented several challenges in applying differential privacy in the setting of continual location sharing. Therefore, they proposed δ-location set based differential privacy to protect a user's actual location under temporal correlations. Similarly, quantified the risk of differential privacy under temporal correlations, and proposed a mechanism that converts any existing DP mechanism into one defending against temporal privacy leakage.proposed AdaTrace, which generates traces through a four-phase synthesis process consisting of feature extraction, synopsis learning, noise injection, and generation of differentially private synthetic location traces. studied the problem of using a single privacy metric while finding an optimal mechanism to preserve user's location privacy. Recently, differential location privacy is also widely used in mobile crowd sensing .

Expected Inference Error. The expected inference error leverages the resilience to the prior information of an adversary as a formal metric to assess privacy loss of a location obfuscation mechanism which aims to obfuscate an adversary by slightly modifying reported locations of two nearby users. designed a method, which can be used to build an optimal location obfuscation mechanism. It formulates the problem as a linear program problem where the constraints are determined by QoSs. More recently, Ahmad *et al.* developed an effective intent-aware query obfuscation solution to maintain Bayes optimal privacy in the web searching environment. Geo-indistinguishability guarantees location privacy with respect to the information leakage, but they do not consider the prior information. On the contrary, the mechanisms with expected inference error are based on the assumption of prior information, but without consideration of constraint on the posterior information gain. To absorb the advantages of these two techniques, was the first mechanism integrating the expected inference error and geo-indistinguishability to improve location privacy. Based on this work, Yu *et al.* further suggested that a specific user may have different privacy/utility preferences for different locations, time and services. Thus, they proposed a new scheme PIVE, which can support customized privacy requirements for users. However, we argue that existing geo-indistinguishability-based mechanisms do not offer privacy protection against our proposed long-term observation attacks. To solve this problem, we propose Eclipse, which can resist the long-term observation attacks by integrating *k*-anonymity, geo-indistinguishability and the expected inference error.

3. BACKGROUND

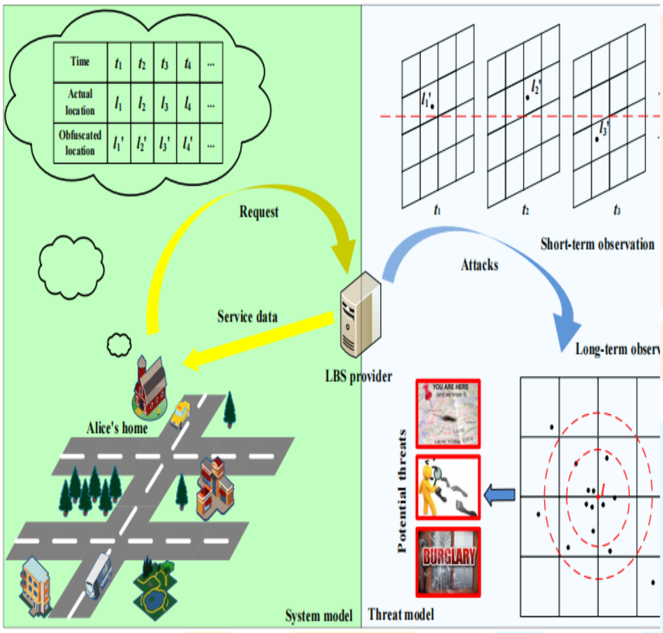
3.1 Definitions

3.1.1 Differential Privacy

Differential Privacy (DP) renders correct statistic information of a dataset without revealing each individual's privacy.Specifically, DP ensures that the presence of a single tuple from an individual does not significantly alter the outcome of a query.

3.1.2 Geo-indistinguishability

Geo-indistinguishability renders privacy preservation for locations that are geographically close. Specifically, for any radius r > 0, a user achieves r-privacy within radius r. A user can customize her privacy requirements by a tuple (l, r), where r is the radius she is concerned with and l is the privacy level she wishes for that radius. In this case, it is sufficient to require geo-indistinguishability for E = l/r.



Time	t_1	t_2	t_3	t_4	...
Actual location	l_1	l_2	l_3	l_4	...
Obfuscated location	l_1'	l_2'	l_3'	l_4'	...

3.1.3 Expected Inference Error

For the adversary, we assume that he knows the distribution information M(l0 |l) of the obfuscation mechanism, where l0 is observed from the output of the obfuscation mechanism. It is also easy for him to get the prior information f(·). With such information, he can calculate the posterior distribution P r(l|l0) to infer the actual location l of a specific user. Therefore, the adversary's goal is to choose an estimated location ˆl to minimize the user's conditional expected privacy. The user's conditional expected privacy [35] for an arbitrary ˆl can be calculated by P l P r(l|l0) deuc(ˆl, l). Thus, minimizing l can be described as:

min ˆl X l P r(l|l0)deuc(ˆl, l).

3.2 System and Threat Models

The system model, as illustrated in the left front of Figure 1, includes mobile users and a LBS provider. In our work, we focus on the Point of Interests (PoIs) searching service, which is one of the most popular services in LBSs, i.e., finding nearby pharmacies. Assume user Alice periodically submits service requests at her home, and receives service data from the LBS provider. Each request normally consists of information like her identifier, query time, current location, queried PoI and query range. To preserve such private location relative information, we assume that Alice applies location obfuscation mechanisms to hide her actual location. In the threat model, as illustrated in the right side of Figure 1, we assume the LBS provider is an *honest-but-curious* adversary. Specifically, it would like to learn users' actual locations by launching kinds of attacks. As Alice frequently submits obfuscated locations from an actual location (e.g., Alice's home), the LBS provider is able to collect these historical obfuscated locations *l*10 , *l*02 , *l*03 , *l*04 , ... According to these obfuscated locations, the LBS provider could infer Alice's actual location *l* with high probability. Once the location of Alice's home is revealed, she may suffer kinds of potential threats such as tracking and burglary. In short, upon possessing obfuscated locations, the untrusted LBS provider could bypass the privacy protection of location obfuscation mechanisms and infer a user's actual location by performing two types of attacks:

- Short-Term Observation Attacks. In a short-term observation attack, the adversary can infer a user's actual location based on a snapshot of one obfuscated location and additional prior information. The prior information for a specific user can be obtained in multiple ways, such as using the population density or using the user's historical access information of a location-based service.

- Long-Term Observation Attacks. In a long-term observation attack, besides all the prior information mentioned above, the adversary can infer a user's actual location based on a sequence of obfuscated locations of a user collected by an adversary over a period of time.

Leveraging geo-indistinguishability alone is vulnerable under both short-term observation and long-term observation attacks. A recent study [10] has proposed a solution to defend against short-term observation attacks by integrating geo-indistinguishability and expected inference error.

However, the above mentioned solution is vulnerable to our proposed long-term observation attacks. Therefore, how to suppress long-term observation attacks remains open. The short-term observation attacks and the long-term observation attacks are different in the observations on a particular user. The observations constitute the inputs for the inference attacks. Since the inputs for inference attacks are not the same, the attack process will also be different.

Next, we formally illustrate the details of the observations and attack processes of the two attacks. Short-Term Observation Attacks: In this setting, the adversary's observation is limited to one snapshot, namely one obfuscated location in this paper. Based on an obfuscated location, the adversary could calculate the posterior probability distribution. Let L be the set of all the possible locations, the posterior probability distribution P r(l|l0) for l ∈ L can be computed by:

P r(l|l0) = f(l)M(l0 |l) P l∈ L f(l)M(l0 |l),

where l0 denotes an obfuscated location, f(l) denotes the prior information of an actual location l, M(l0 |l) represents the distribution of the location obfuscation mechanism. Upon obtaining the posterior distribution, the adversary can estimate the actual location as ˆl by two different ways:

ˆl = arg max l∈ L P r(l|l0),

l = arg min ˆl∈ L X l∈ L P r(l|l0)deuc(ˆl, l).

Short-term observation attacks based on Formula are referred to as bayesian inference attacks. The ones based on Formula are referred to as optimal inference attacks. Long-Term Observation Attacks: In this setting, the attacker obtains a user's requests over a period of time, namely a series of obfuscated locations produced by the same actual location (e.g., home) in this paper. Given these obfuscated locations of a user, the adversary can obtain a set O={o1, · · · ,oi, · · · on}, where ol denotes frequency of location l ∈ L to be considered as the obfuscated location.

Based on the set *O*, this adversary could estimate a user's actual location as:

ˆ*l* = *arg* max *l*∈L *ol*. (9)

Long-term observation attacks are particularly effective on users who often submit location check-ins at the same location. Even a user has applied existing mechanisms built upon geo-indistinguishability, the frequency of the user's actual location very likely remains the highest among all the obfuscated locations. Since existing geo-indistinguishability-based LPPMs always generate noises with the mean of zero, and add such noises to the actual location, thus the actual location is easily identified under long-term observation attacks. Note that it is not appropriate to add noises with non-zero mean value to withstand the above attacks. One reason is that the noises used by mechanisms to achieve differential privacy (i.e., the Laplace and Gaussian mechanisms) are all with the mean of zero. In addition, adding noises with non-zero mean value could not defense the short-term observation attack, since it cannot guarantee the lower bound of the expected inference error.

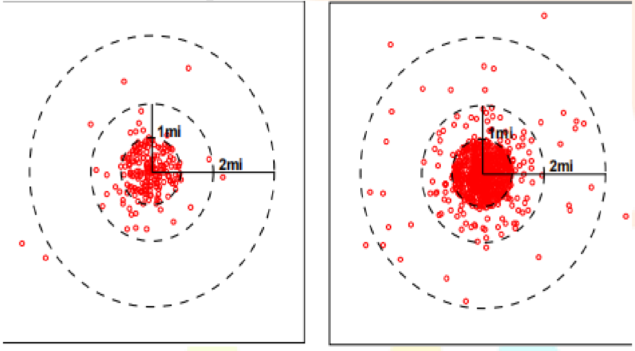
Note that, the short-term observation attacks work well when the LPPM does not consider the prior information. Once a LPPM is conducted based on the assumption of prior information, the short-term observation attacks would be ineffective. On the contrary, the long-term observation attacks utilize the fact that existing geo-indistinguishability based mechanisms always add noise with a mean value of 0 to the actual location. That is, the reasons for the success of the two types of attacks are different.

Discussion. A limitation of our long-term observation attacks is that it works under the condition of cell-based discretization. Without discretization, observing multiple identical locations is highly unlikely, considering the precision of GPS locations, thus our attack (9) may fail. That is, our long-term observation attacks may not work well in the continuous location space without discretization. Next, we leverage a concrete example to verify that previous location obfuscation mechanisms fail to defend against long-term observation attacks.

3.3 Motivating Experiment

In current LBSs, a user utilizes service data from the LBS provider by submitting a query, which always includes information like her identifier, query time, current location, queried PoI, query range. To preserve users' location privacy, one approach is to apply geo-indistinguishability .

However, one critical limitation of this approach is that when multiple queries are submitted from the same location, i.e., a user may frequently request nearby pharmacies every at home, user's location privacy would be significantly decreased. In short, such mechanisms build upon geo -indistinguishability fail to resist long-term observation attacks. Next, we would like to leverage one typical mechanism build upon geo-indistinguishability to further elaborate our motivation. We assume the obfuscated locations in this example are generated by a recent method called PIVE [10]. The reason why we only choose PIVE is that it is representative of geo-indistinguishability-based mechanisms.Besides, compared with other geo-indistinguishability-based mechanisms, PIVE can resist various inference attacks, namely the short-term observation attacks in this paper.Therefore, we can reveal the limitations of mechanisms build upon geo-indistinguishability by analyzing PIVE. In our example, the local map is divided into a grid of 32×32 cells, suppose a user is located at the central cell. We use a real-world dataset Brightkite3 to generate the query probability for each cell as the prior information.



details of Brightkite are introduced in Section 5. PIVE first determines a set of obfuscated locations by considering the user-defined inference error threshold, and calculates the sensitivity of differential mechanism over the determined. With such sensitivity, PIVE generates an obfuscated location in a differentially private way. To demonstrate the consequence of a long-term observation attack, we produce 200 and 800 obfuscated locations based on one actual location, and plot these obfuscated locations on four maps. The results are presented in Figure

2. We can observe that although each obfuscated location does not give away the actual location directly, an attacker can infer the actual location by analyzing the distribution of these obfuscated locations. For example, they could refer to the centers of those maps as the actual locations. The more obfuscated locations, the more accurate the distribution, thus, it is easier to reveal the actual location. Figure 2 also illustrates the relationship between the distance from the actual location and the frequency of occurrence as the obfuscated locations. It is obvious that obfuscated locations with shorter distance to the actual location are with higher frequency of occurrence. It means that the long-term observation attacks can be performed by sorting the historical obfuscated locations based on the frequency of occurrence. Besides, we also find that PIVE may generate obfuscated locations far away from the actual location, i.e., 2mi, which may cause very poor QoS. Therefore, our design objective is to propose an obfuscation mechanism, which can resist long-term observation attacks while satisfying the QoS requirement.

3.4 Problem Statement

We assume that mobile users are within an area, which is divided into $n \times n$ disjoint cells, denoted as $C = \{c1, c2. \cdot\cdot\cdot cn\times n\}$. A static (or mostly static) user may periodically sends requests from a same (or several same) actual location. Based on the observations of a user's requests,there are two types of attacks: the short-term observation attacks and the long-term observation attacks. In short-term setting, the adversary's observation is limited to one snapshot of an obfuscated location $l0$. Based on $l0$ and some other prior information, the adversary can infer the actual location l in two ways: the bayesian inference attack and the optimal inference attack (8). As for long-term observation attacks, the attacker obtains a series of obfuscated locations of a particular user generated from the same actual location. Given these obfuscated locations, the attacker can build $O = \{o1, o2, \cdot\cdot\cdot , on\times n\}$ to record the number of occurrence times of each location to be considered as the obfuscated location.Adversaries who observe a user for a long time period can easily obtain O, which can be used to perform long-term observation attacks.Therefore, our problem statements can be summarized from three dimensions. First, we need to resist long-term observation attacks while being able to withstand shortterm observation attacks. Second, we need to guarantee the QoS requirement while improving the degree of privacy. Third, we need to support customizable privacy/QoS requirement of mobile users. This motivates the design and implementation of Eclipse.

4 OUR PROPOSED ECLIPSE

4.1 Framework of Eclipse

The main purpose of our mechanism is to effectively resist both the short-term observation and the long-term observation attacks while satisfying user's QoS requirement. Specifically, Eclipse first filters out a set of locations from the possible outputs according to the user's QoS requirement. Then it chooses an anonymity set to bound the expected inference error in order to resist the short-term observation attack. Finally, Eclipse produces an obfuscated location against long-term observation attacks in a differential and anonymous way based on the obtained anonymity set and the set of possible outputs. Figure 3 shows the three-phase framework of our Eclipse. Note that, our mechanism is a client-based solution, that all the phases are executed in the smart device in user's hand.

Eclipse provides three privacy and one QoS control knobs: the size of the anonymity set *k*, the minimum inference error *Em*, the budget of differential privacy and the user's QoS requirement *Q*. By combining these parameters, Eclipse allows users to define their desired privacy preference and QoS requirement under different circumstances. Specifically, the parameter *k* is used to represent the privacy degree against long-term observation attacks. As a result, the adversary with performing long-term observation attacks cannot distinguish user's actual location from a candidate set with *k* locations. The parameter *Em* aims to bound the expected inference error in the worst case to resist the short-term observation attacks. The parameter allows users to constrain the posterior information leakage via the provisioning of differential privacy. The parameter *Q* guarantees the minimum QoS level that Eclipse can provide.

4.2 Set of Possible Outputs Determination

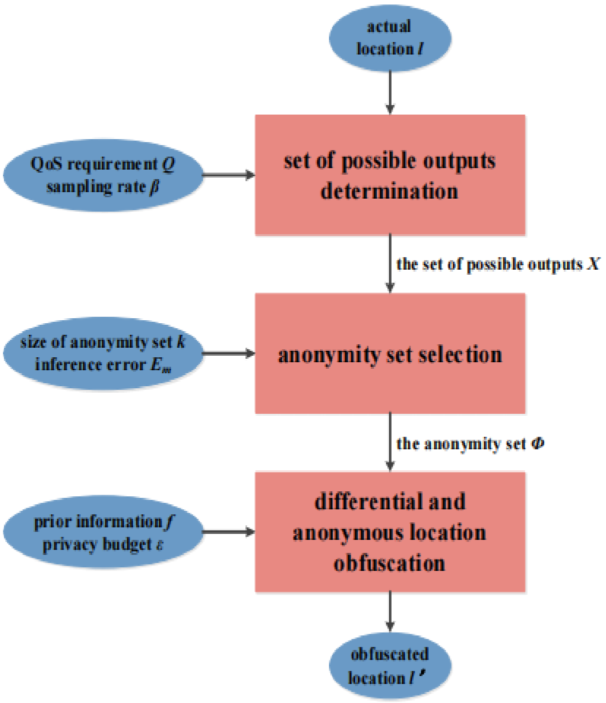

actual location l
QoS requirement Q sampling rate β
set of possible outputs determination
the set of possible outputs X
size of anonymity set k inference error E_m
anonymity set selection
the anonymity set Φ
prior information f privacy budget ε
differential and anonymous location obfuscation
obfuscated location l'

Fig. 3. The framework of Eclipse

Given a specific user's actual location *l*, the first problem is how to determine the set of possible outputs, such that each location output as the obfuscated location from the set can satisfy the user's QoS requirement.

We first discuss how to calculate the QoS. The QoS is related to the service data provided by the LBS provider. The service data is calculated based on the location, the query range and queried PoI provided by a user. When a user submits her actual location, she could obtain the service data that she needs. On the contrary,once a user submits an obfuscated location, the user would obtain two types of service data: data that meets her needs, namely the **correct data** and data that is irrelevant to her needs, namely the **redundant data**. In our mechanism, in order to meet user's QoS requirement, it may change the original query range *ro* into the submitted query range *rs*. Generally, larger submitted query range usually brings more correct data, but leads to more redundant data as well. Therefore, the QoS of the obfuscated location *l0* with submitted query range *rs*
could be calculated by:

$QoS\ (l0\ ,\ rs) = A(l0\ ,\ rs) \cap A(l,\ ro)\ /\ A(l,\ ro) - \omega \cdot A(l0\ ,\ rs) - A(l,\ ro)\ /\ A(l0\ ,rs)$,

where $A(l0\ ,\ rs)$ denotes the area covering all the obtained service data, $A(l,\ ro)$ is the area covering all the service data the user needs. $\omega \in [0,\ 1]$ denotes the proportion of the redundant data when calculating QoS. Note that, the formula of QoS could be changed according to special circumstances. According to Formula , we then introduce how to determine the set of possible outputs. Eclipse first produces a candidate set C, then conducts sampling from C to form the set of possible outputs X.

Specifically, we use *d* to denote the diameter of the candidate set when pinpointing them onto the local map. Once *d* is determined,the farthest location from the actual location *l* could be obtained, which is denoted as *ld*. Then, Eclipse sets the submitted query range *rs* to calculate the corresponding QoS. The minimum value of the submitted query range *rs* is equal to the value of the original query range *ro*. Therefore, it uses the original query range as the submitted query range to calculate the minimum value of d, which is described as below:

$$dmin = arg\ max\ d\ Qos(ld, ro) \geq Q.$$

A large value of d can increase the degree of privacy protection due to more candidates could be output as the obfuscated locations. However, too large values of d may violate the QoS requirement. Eclipse seeks for dmax by increasing d, which starts from dmin, until it cannot find a corresponding rs to satisfy user's QoS requirement. Then, it obtains the candidate set using dmax. The candidate set can be considered as a circular centered at the actual location, and with the diameter of dmax. Although all locations in the candidate set C can meet the user's QoS requirement, they cannot be chosen into the set of possible outputs directly, due to the privacy issue.Once the candidate set is used as the output set, the adversary can collect all the obfuscated locations through the long-term observation, then determines a circular covering all these obfuscated locations. Finally, the center of the circular would be inferred as the actual location. In order to avoid such privacy issue while ensuring user's QoS requirement, we design a sampling-based solution which samples some element from the candidate set C and forms the set of possible outputs X. The sampling probability β can be determined by user. Note that, the set of possible outputs remains unchanged if the QoS requirement is the same as before. The detail can be found in Algorithm 1.

4.3 Anonymity Set Selection

With the determined set of possible outputs in hand, another problem is how to efficiently determine the anonymity set with guaranteeing the expected inference error. According to , the expected inference error via the anonymity set Φ can be calculated by:

E(Φ) = min ˆl∈ Φ X l∈ Φ f(l) P l∈ Φ f(l) deuc(ˆl,l),

where f is the distribution of query probabilities over the set of possible locations as the prior information. To ensure a lower bound for the expected inference error, it is sufficient that: E(Φ) ≥ e Em. Then, the next problem is how to effectively search over the plane for the anonymity set, which can satisfy the above formula. Inspired by PIVE, we propose a Hilbert curve-based searching algorithm to improve the searching efficiency. Our searching algorithm differs from the PIVE's Hilbert curve algorithm in two ways. First, we should search an anonymity set based on the size of anonymity set k. Second, we add randomness to the process of selecting the anonymity set. Hilbert curve provides a mapping from a data point in a 2-D space to a point in 1-D space that preserves the proximity of data. Hilbert curve maps a location point in a plane to a 1-D value, which is denoted as H(·), we call H(·) as the Hilbert value of this location. Given a user's actual location l, the size of the anonymity set k and the expected inference error Em, our algorithm searches the neighborhood of l along the Hilbert curve to find an anonymity set Φ, which satisfies |Φ| = k, l ∈ Φ and E(Φ) ≥ Em. Specifically, let l−m, l−m+1,..., l0(= l), l1,..., ln be the sequence of locations in the searching neighborhood of l along the Hilbert curve, sorted by their Hilbert values. Our algorithm then calculates the inference error for every interval [li , li+k−1] for 1-k ≤ i ≤ 0 by Formula (12). Once an interval that satisfies E([li , li+k−1]) ≥ e Em, it can be added to the candidate set. If all intervals cannot meet the threshold Em, Eclipse increases the size of the interval k, until the candidate set has at least one interval. Finally, it randomly chooses a set from the candidate set as the anonymity set. Note that, for a given actual location, when the privacy requirements are the same as before, the anonymity set remains unchanged. The detail can be found in Algorithm 2.

4.4 Differential and Anonymous Location

Obfuscation Given the anonymity set Φ and the set of possible outputs X, Eclipse achieves differential privacy by employing the exponential mechanism since we require the obfuscated location could be chosen from X.

In exponential mechanism, we should set a function to measure the quality score [39]. In this paper, the quality score of the obfuscated location l0 is measured by the Euclidean distance between l0 and user's actual location l,

denoted as deuc(l, l0), as well as the average quality loss, denoted as loss(l0). Smaller distance and smaller average
quality loss could lead to a higher score, where the quality score can be calculated by:
$q(l, l0) = -(deuc(l, l0) + loss(l0))$,
where
loss(l0) = X l∈ Φ f(l) P l∈ Φ f(l)deuc(l, l0).
As the anonymity set indicates the“neighbouring” locations to the user's location, the sensitivity of the utility function q is:
4 q = max l0 ∈ L max li,lj∈ Φ | − deuc(li, l0) + deuc(lj , l0)|.
Obviously, according to triangle inequality, for any li, lj ∈ Φ, |deuc(li , l0) − deuc(lj , l0)| ≤ deuc(li , lj) < D(Φ), so 4 q = D(Φ), where D(Φ) denotes the diameter of Φ.
Exponential mechanism ME : Given a user's location l, the anonymity set Φ and the set of possible outputs X, the exponential mechanism ME selects and outputs a location l0 ∈ X with probability proportional to exp(q(l,l0) 24 q).
When a user issues queries for multiple times, she can obtain differential privacy guarantees by employing the exponential mechanism independently to generate obfuscated locations before sending each query. However, the user performs n queries via the above exponential mechanism ME providing -differential privacy, then enjoys n-differential privacy. The privacy degree decreases sharply with the increase of the number of queries n. That is to say, the adversary can perform the long-term observation attacks by collecting and analyzing the obfuscated locations produced in historical queries. To make the exponential mechanism resist the longterm observation attacks, we propose the following Eclipse mechanism ML, and build a new function as:

$$ML(z|l) = \alpha 1 \cdot ME(z|l1) + ... + \alpha k \cdot ME(z|lk),$$

where li(i ∈ [1, k]) is a particular location in the anonymity set Φ, and ME (z|l) means that we use exponential mechanism on location l, and αi denotes the probability that the specific exponential mechanism ME (z|li) can be chosen.
Note that, the probability can be determined by the user, such O can obey a certain distribution.
In brief, the Eclipse mechanism ML can randomly choose a location.in the
anonymity set to produce the obfuscated location.

4.5 Security Analysis

In this subsection, we focus on analyzing the privacy issues of the users against possible inference attacks performed by the untrusted LBS provider. We prove that our Eclipse can
effectively resist the long-term observation attack, as well as another two kinds of inference attacks: the bayesian inference attack and the optimal inference attack.
Theorem 1. The resistance to the long-term observation attacks is achieved with the ratio of the probability that outputs the same obfuscated location on two different location in the same anonymity set.
Proof: Since our Eclipse ML adds the noise to a
location in the anonymity set randomly, instead of adding the noise to the actual location directly. It means that the locations in the same anonymity set can produce obfuscated location with the same probability
Theorem 2. The protection of -differential location privacy is achieved with the upper bound of the ratio of the probability that outputs the same obfuscated location on two different location in the same anonymity set.
Proof: We prove above conclusion by computing the ratio of the probability that the exponential mechanism outputs the same obfuscated location l0 on two different location li , lj , which are in the same anonymity set Φ.
-differential location privacy is achieved with the upper bound of the ratio of the probability that outputs the same obfuscated location on
two different location in the same anonymity set.
Theorem 3. The resistance to the Bayesian inference attacks is achieved with the upper bound of the posterior probability.
Proof: Through Formula (16), we can find that ML is composed of ME , so we only need to prove that ME can resist the Bayesian inference attacks. According to Formula , the upper bound of posterior distribution P r(l|l0) can indicate the capability of mechanism for defending against the Bayesian inference attacks.The upper bound of the posterior probability implies that no matter what prior information the adversary has, differential privacy can constraint the posterior probability P r(x|x0). That is to say, it can limit the posterior information gain of the adversary.
Theorem 4. The resistance to the optimal inference attacks is achieved with the lower bound of expected inference error.
Proof: Similar to the proof above, we only need to prove that ME can resist the optimal inference attacks. According to Formula (8), we can see the lower bound of inference error can imply the capability of mechanism for defending against the optimal inference attacks. The expected inference error can be represented as

min ˆl∈ X X l∈ X P r(l|l0)deuc(ˆl, l)

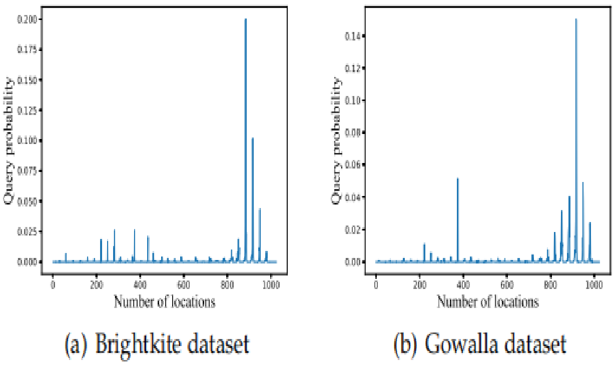
5 PERFORMANCE EVALUATION

5.1 Settings

To further verify the strength of Eclipse, we compare it with two recent LPPMs:

- Lap [8]: To generate the radius of the Laplace noise, this mechanism first samples p uniformly in the interval [0, 1). Then, it sets r = 1 (W−1 p−1 e + 1), where W−1 is the -1 branch of the Lambert W function, and samples θ uniformly in the interval [0, π) at the same time. Finally, it uses r and θ to generate the obfuscated locations.
- PIVE [10]: A dynamic differential location privacy mechanism with personalized error bounds. There are two parameters and Em (default, Em = 1).

Datasets: We use two real-world datasets Brightkite4 and Gowalla5 to conduct our experiments. Both of them are location-based social networking service providers where users share their check-in data. Table 2 shows other details of datasets. To simplify our experiments, we restrict the PoIs to a infinite region of San Francisco area between the latitude coordinates (37.5 and 37.81) and longitude (−122.6 and −122.29). Because it contains many PoIs and lots of user's check-in data. We divide the region into 32 × 32 cells firstly, then calculate the query probability by counting how many user's check-in data on each cell and normalizing the resulting histogram for each dataset. The obtained probabilities of two datasets are shown in Figure 4.



(a) Brightkite dataset (b) Gowalla dataset

Fig. 4. Query probability for two datasets

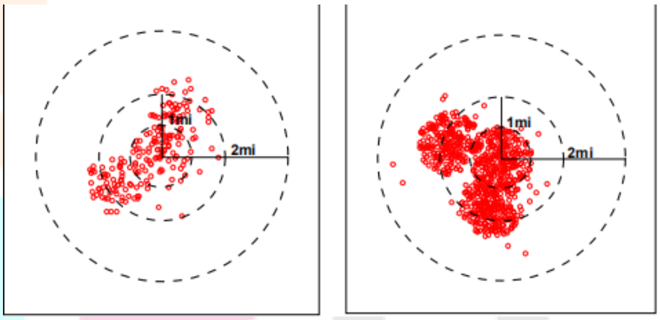
Privacy Metrics: We mainly use long-term privacy metric to measure the privacy, and use the short-term privacy metric as the supplement in some experiments.

- Long-term privacy metric: We collect these obfuscated locations, which are derived from the historical queries. Then we pick the three most frequently occurring locations, more formally:

arg max l1,l2,l3∈ L ol1 + ol2 + ol3 ,

Short-term privacy metric: We use unconditional expected inference error (5) as the short-term privacy metric. It is measured by the expected inference error of the Bayesian adversary averaged over all possible locations in X, under the inference attack launched by the Bayesian adversary.

Utility Metrics: In they provided two utility metrics for different applications. In some applications, service quality degrades linearly as the obfuscated location moves away from the actual location, in such cases the Euclidean distance deuc provides a reasonable way to measure utility. Other applications tolerate a noise up to a certain threshold with almost no effect on the service, but the quality drops sharply beyond the threshold. We focus on the Point of Interests (PoIs) searching service, which belongs to the latter. Therefore, we use the following distance metric:



(a) 200 times observation
(b) 800 times observation

Fig. 5. The distribution of the output locations by using Eclipse

In the following experiments, we set t = ro, where ro denotes the original query range. Based on the above distance metric, we measure utility by running the mechanism 1000 times, then calculate the average distance. We refer to the average distance as the utility metric. The value is between 0 and 1. Larger value means the better utility the mechanism can achieve.

5.2 A Concrete Example

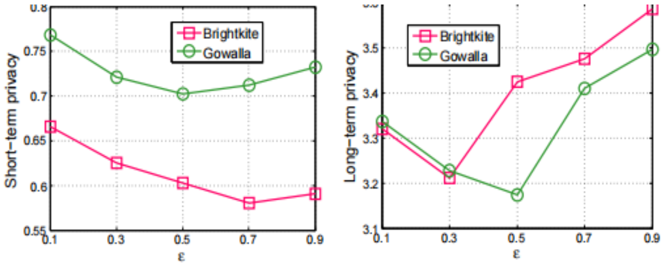
To verify the effectiveness of our proposed Eclipse, we first perform a long-term observation attack under the experiment setting in Subsection

3.3. Since we use the anonymity set to produce the obfuscated locations, instead of one actual location. Figure 5 shows the outputs by using our Eclipse, they form multiple hotspots (the size of anonymity set k = 3
in this specific case) compared to the single one in Figure
2. Thus, the adversary cannot easily distinguish the actual location by long-term observation.

5.3 Effects of Parameters on Eclipse

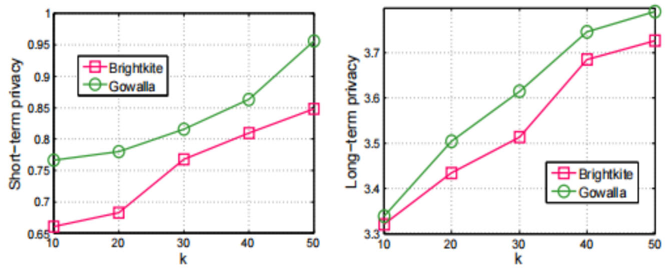
We evaluate the impact of differential privacy parameter ,inference error threshold Em, the size of anonymity set k and the requirement of QoS Q on location privacy. In the meantime, we evaluate the effect of two parameters the size of anonymity set k and the requirement of QoS Q on the overhead of our proposed Eclipse.

The effect of on privacy. Figure 6 shows the trends of the short-term privacy and long-term privacy when varies from 0.1 to 0.9. Obviously, larger means the less noise generated, and leads to privacy decrease. However, e Em increases linearly with Em with a small factor e, and the larger is, the larger value of sensitivity will be. That is to say, as becomes larger, the impact of sensitivity changes on the privacy degree becomes larger compared with that of privacy budget . Note that, for the Brightkite dataset, the increasing of short-term privacy occurs when > 0.7, but in long-term privacy case, the privacy growth occurs when > 0.3. That is because the long-term privacy is more For long-term privacy, it influences not only the sensitivity of noise for differential privacy, but also the size of the anonymity set. The size of the anonymity set would increase when all sets with size k cannot satisfy the expected inference error eEm. Note that, eEm has almost the same effect on long-term and short-term privacy when the increasing eEm does not cause an increase in the size of the anonymity set. The effect of k on privacy. Figure 7 shows that both of the short-term privacy and long-term privacy increase



(a) Short-term privacy comparison(b) Long-term privacy comparison

Fig. 6. Privacy comparison when ϵ changes



(a) Short-term privacy comparison(b) Long-term privacy comparison

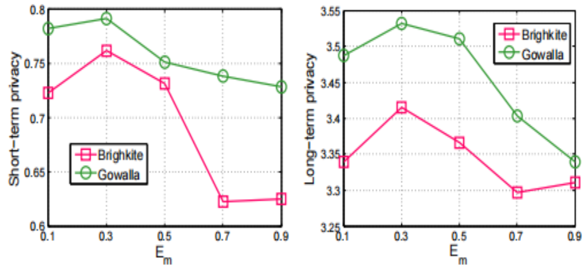
Fig. 7. Privacy comparison when k changes

sensitive to e Em.

as k increases from 10 to 50. k denotes the size of the anonymity set, which can influence the value of sensitivity used in differential mechanism. Larger k means larger value of sensitivity we should set, thus, the more noises can be generated. We can make a conclusion that the larger k can provide stronger privacy degree for both short-term and long-term privacy metrics. However, k cannot be set too larger, due to the data utility and the extra computational overhead. The user can set the value of k according to the specific environment.

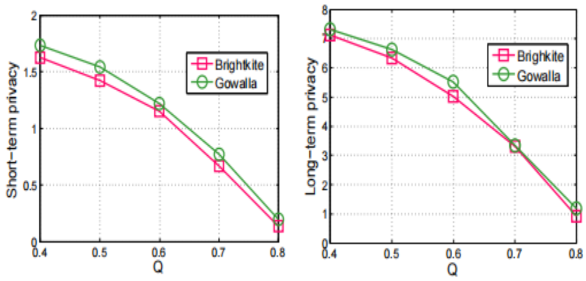
The effect of Em on privacy. Figure 8 shows the shortterm privacy and long-term privacy fluctuate with Em, varying from 0.1 to 0.9. Since Em is closely related to the lower bound of inference error, for a definite k, the larger
Em can discard more unqualified sets, which can decrease the privacy degree. However, when Em continues growing, all sets with size of definite k may not satisfy the threshold eEm, then, we increase the size of the anonymity set, thus, the privacy degree increases. We should know that the process is cycled as Em changes.

The effect of Q on privacy. Figure 9 shows the shortterm privacy and long-term privacy decrease as Q changes from 0.4 to 0.8. Since the higher QoS requirement leads to the smaller size of the set of possible outputs, which degrades the privacy degree in terms of both privacy metrics. From the result, we can see that the privacy decrease sharply when the QoS requirement becomes higher.



(a) Short-term privacy comparison(b) Long-term privacy comparison

Fig. 8. Privacy comparison when E_m changes



(a) Short-term privacy comparison(b) Long-term privacy comparison

Fig. 9. Privacy comparison when Q changes

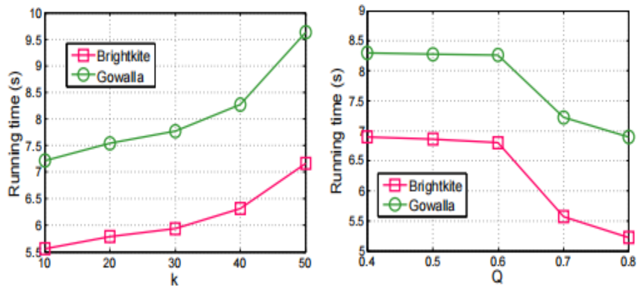
The effect of k on overhead. Figure 10(a) shows the running time increases when k increases from 10 to 50. k denotes the size of the anonymity set. Larger k means larger candidate sets we should calculate when we determine the anonymity set, thus, the more time we need. Thus k cannot be set too larger, due to the computational overhead.

The effect of Q on overhead. Figure 10(b) shows the trends of running time when Q increases from 0.4 to 0.8. Since the higher QoS requirement leads to the smaller size of the set of possible outputs, which degrades the running time when obtaining the obfuscated location using the exponential mechanism. From the result, we can see that the running time is stable at first and decreases sharply when the QoS requirement becomes higher.

5.4 Performance Comparison with Other Mechanisms

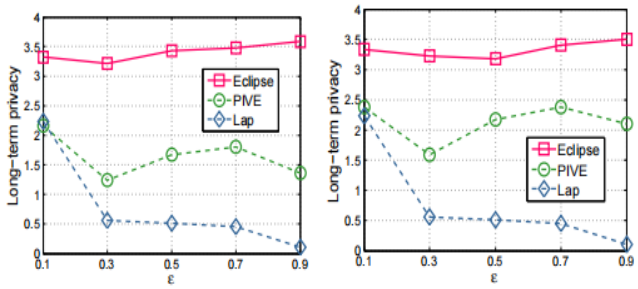
In this subsection, we compare Eclipse with the aforementioned two typical mechanisms to verify the advantage of our proposed Eclipse. It is worth to note that Eclipse allows the user to specify a set of different parameters like , k, Em, Q. Given that other mechanisms do not have so many parameters, we set k = 10, Em = 1, Q = 0.7 as the default value, and compare these mechanisms with changing . **Comparison on long-term privacy.** Figure 11 shows the long-term privacy comparison between Eclipse, PIVE and Lap [8] when changes from 0.1 to 0.9. Compared with that of Eclipse, the long-term privacy degree of PIVE and Lap are much smaller. It indicates that the two mechanisms fail to preserve user's location privacy under long-term observation attacks. We can see that our Eclipse mechanism is robust as the changes. Hence, our mechanism can resist the long-term observation attacks, no matter what value of takes.

Comparison on short-term privacy. Figure 12 shows the short-term privacy comparison between Eclipse, PIVE and Lap when changes. The Lap mechanism does not consider



(a) k (b) Q

Fig. 10. Comparison of running time



(a) Brightkite dataset (b) Gowalla dataset

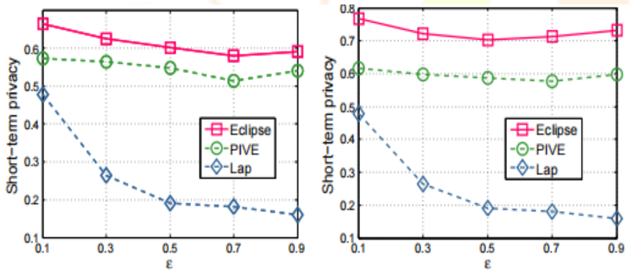
Fig. 11. Long-term privacy comparison

the expected inference error, thus performs worst, and the short-term privacy decreases sharply as increases. On the contrary, the short-term privacy of our Eclipse and PIVE decrease first then increase, since both of them use e Em to guarantee the expected inference error, thus also influences the sensitivity used in differential privacy.

Comparison on utility. Figure 13 shows the utility comparison between Eclipse, PIVE and Lap when changes from 0.1 to 0.9. We can see that when is smaller than 0.3, the utility of PIVE and Lap are much smaller than Eclipse. However, when becomes larger, the utility of Lap increases sharply and outperforms Eclipse and PIVE. The results show that both Eclipse and PIVE mechanisms obtain stronger privacy protection at the cost of partial utility. Moreover, Eclipse can not only provide stronger privacy protection than PIVE, but also outperform PIVE in terms of utility. This is because our Eclipse considers the QoS requirement, which can improve the utility to some extent.

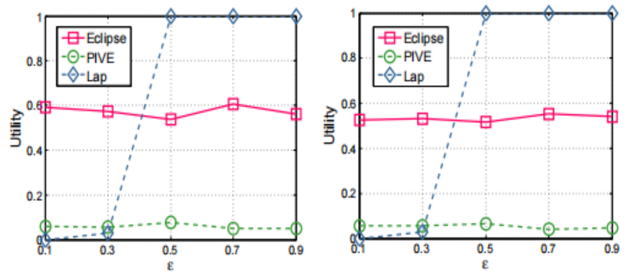
6 CONCLUSION

To effectively prevent mobile user's location privacy from the long-term observation attacks, we proposed Eclipse, which is a three-phase differential location privacy-preserving mechanism. Eclipse combines geo-indistinguishability, k-anonymity and expected inference error together. Specifically, the set of possible outputs determination phase first determines the set of possible outputs by ensuring the user's QoS requirement. Then, the anonymity set selection phase determines an anonymity set, which guarantees the expected inference error bound. Finally, the differential and anonymous location obfuscation phase generates an obfuscated location differentially and anonymously. The evaluation results on two real-world datasets show the efficacy and efficiency of our Eclipse.



(a) Brightkite dataset (b) Gowalla dataset

Fig. 12. Short-term privacy comparison



(a) Brightkite dataset (b) Gowalla dataset

Fig. 13. Utility comparison

REFERENCES

[1] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of
location sharing in mobile social networks: Attacks and defense,"
IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4,
pp. 646–660, 2018.

[2] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in ACM
WWW, 2009.

[3] H. Li, Q. Chen, H. Zhu, D. Ma, H. Wen, and X. S. Shen, "Privacy
leakage via de-anonymization and aggregation in heterogeneous
social networks," IEEE Transactions on Dependable and Secure Com
puting, 2017.

[4] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Loca
tion privacy in usage-based automotive insurance: Attacks and
countermeasures," IEEE Transactions on Information Forensics and
Security, vol. 14, no. 1, pp. 196–211, 2018.

[5] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection
of mobile apps location privacy threats," in USENIX Security, 2015.

[6] K. Fawaz and K. G. Shin, "Location privacy protection for smart
phone users," in ACM CCS, 2014.

[7] Z. Montazeri, A. Houmansadr, and H. Pishro-Nik, "Achieving
perfect location privacy in wireless devices using anonymization,"
IEEE Transactions on Information Forensics and Security, vol. 12,
no. 11, pp. 2683–2698, 2017.

[8] M. E. Andr
´
es, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for
location-based systems," in ACM CCS, 2013.

[9] C. Dwork, "Differential privacy," in Springer ICALP, 2006.

[10] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy
with personalized error bounds," in ISOC NDSS, 2017.

[11] M. Gruteser and D. Grunwald, "Anonymous usage of location
based services through spatial and temporal cloaking," in ACM
MobiSys, 2003.

[12] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous commu
nication technique using dummies for location-based services," in
IEEE ICPS, 2005.

[13] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity

in privacy-aware location-based services," in IEEE INFOCOM, 2014.

[14] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation aware dummy-based privacy protection scheme for location based services," in IEEE INFOCOM, 2017.

[15] Y. Zhang, W. Tong, and S. Zhong, "On designing satisfaction-ratio aware truthful incentive mechanisms for k-anonymity location privacy," IEEE Transactions on Information Forensics and Security, vol. 11, no. 11, pp. 2528–2541, 2016.