Peer-to-Peer Netw. Appl.
DOI 10.1007/s12083-017-0545-9

AGENT: an adaptive geo-indistinguishable mechanism for continuous location-based service

Xindi Ma[1,2] · Jianfeng Ma[1,2] · Hui Li[1] · Qi Jiang[1,3] · Sheng Gao[4]

Received: 25 September 2016 / Accepted: 24 January 2017

Abstract With the widespread use of Location-base Services(LBSs), the problem of location privacy has drawn significant attention from the research community. To protect the user's exact location, a new notion of privacy, named geo-indistinguishability, that adapts differential privacy has been proposed for LBSs, recently. However, the obfuscation mechanism satisfying this privacy notion only works well in the case of snapshot LBS, which would not apply to the case of continuous LBSs due to the quick loss of privacy caused by the correlation between locations in the trace. In this paper, we propose a novel mechanism, namely AGENT, to protect the user's location privacy in continuous LBSs. In AGENT, a R-tree is introduced to realize the reusable of generated sanitized locations, which achieves the notion of geo-indistinguishability with less consumption of privacy budget. Finally, empirical results over real-world dataset demonstrate that with the same utility, our mechanism consumes less privacy budget to obfuscate the same trace.

Keywords Location-based service · Privacy preservation · Differential privacy · Geo-indistinguishability

1 Introduction

With the advancement of positioning technology and the wildly usage of mobile devices, location-based services(LBSs) have got tremendous development, which range from searching point of interest(POI) to location-based games and location-based commerce [1]. The main challenge for LBSs is how to protect location privacy with high quality of services [2]. Generally, users have to provide LBSs providers(LBSP) with their locations for services. But the disclosure of locations has seriously threatened users' privacy while they request the LBSs. Since digital traces of users' whereabouts contain some sensitive information, the attackers can easily infer users' home, their health status, their habits, and their truthfulness. Therefore, it is crucial to protect user's location privacy when requesting the LBSs.

To address the problem, a variant of differential privacy [3], which is called "geo-inistinguishability" [4], was recently introduced to protect users' location privacy. However, a user rarely performs a single location-based query. For each query, although we can simply generate a new obfuscated location and report it to the service provider, referred to as independent mechanism, it is easy to see that privacy is degraded as the number of queries increases, due to the correlation between the locations. Intuitively, in the extreme case, the reported locations are centred around the real one when the user never moves, completely revealing it as the number of queries increases. Additionally, the independent mechanism applying ϵ-geo-indistinguishable noise to n locations can be shown to satisfy $n\epsilon$-geo-indistinguishability. This is typical in the area of differential

✉ Xindi Ma
xdma1989@gmail.com

[1] School of Cyber Engineering, Xidian University, Xi'an 710071, China

[2] School of Computer Science and Technology, Xidian University, Xi'an 710071, China

[3] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China

[4] School of Information, Central University of Finance and Economics, Beijing 102206, China

Published online: 13 March 2017

privacy, in which ϵ is thought as a privacy budget, consumed by each query. As a result, the linear increase makes the privacy budget exhausted prematurely and the remainder locations have to be reported exactly. Hence, the independent mechanism is applicable only when the number of queries remains small.

In this paper, to solve the problems described above, we propose an adaptive geo-indistinguishable mechanism for continuous location-based service, named AGENT. In AGENT, we introduce a R-tree to store and reuse the generated sanitized locations, which obfuscate the trace with less privacy budget. The main contributions of this paper are as follows:

- We propose an adaptive geo-indistinguishable mechanism (AGENT), which protects user's location privacy when requesting LBSs continuously.
- Differential privacy is adopted to sanitize the locations. In order to reduce the privacy budget consumption, we introduce a test mechanism and a R-tree to achieve the reusing of generated sanitized locations. Through AGENT, if a sanitized location is found that satisfies the condition of test, we will report it instead of generating a new one.
- To achieve the adaptive geo-indistinguishability, we also introduce a parameter k in test mechanism. When travelling at different speed, users can select different parameter k to test the sanitized locations adaptively. In this manner, users can make more reasonable utilization of the privacy budget.
- We conduct an analysis of AGENT in terms of theory and practice. The results indicate that AGENT achieves $\epsilon(\Gamma)$-geo-indistinguishability and has a lower consumption of budget than independent mechanism.

The rest of this paper is organized as follows. Section 2 presents some related works. In Section 3, we present some preliminaries. The system overview and problem statement are presented in Section 4, followed in Section 5 by the details of the AGENT. Section 6 presents the analysis of privacy and utility of AGENT. In Section 7, we evaluate the effectiveness of our proposal. Finally, we provide some concluding remarks regarding this paper in Section 8.

2 Related work

There have been many excellent works and surveys [5–7] that summarize the different threats, methods, and guarantees in the context of location privacy. After analysis, we find that the privacy-preserving mechanism can be classified into three categories: anonymization, cryptography, and differential privacy. Then, we will present some related works for the three technologies and analyze the advantages and disadvantages of them.

Based on the mix-zones model, Gao et al. [8] proposed a trajectory privacy-preserving framework to protect users' privacy during the publication of location information. Niu et al. [9] proposed a caching-based solution to protect location privacy in LBSs. Analyzing the location-based functionality of each app, Fawaz et al. [10] designed LP-Doctor to anonymize locations when the app accessed them. In vehicular networks, Ying et al. [11] proposed MPSVLP which is based on dynamical mix zone to encourage vehicles to cooperate in privacy preservation. Rios et al. [12] presented HISP-NC which includes a perturbation mechanism to protect the location of the base station in wireless sensor networks. The authors of [13–16] also proposed some mechanisms which are based on anonymity model to protect users' location privacy when requesting LBSs. Although the above mechanisms are diversiform, each of them assumes the adversaries own specific prior knowledge and also needs a trusted third-party, such as anonymity proxies, to help to achieve the privacy preservation.

Another technique which is usually used to protect location privacy is cryptography. Multi-factor authentication schemes [17–19] could be deployed to prevent unauthorized user from using the location service. Besides, searchable encryption schemes [20–24] could be employed to protect location data privacy from the server. Furthermore, in order to decrease the computational complexity and communication overhead, Wang et al. [25] proposed an efficient optimal private meeting location determination protocol. Similarly, Bilogrevic et al. [26] proposed privacy-preserving algorithms, based on homomorphic encryption method, to determine an optimal meeting location service for a group of users. Without adding uncertainty into query results, Puttaswamy et al. [27] applied secure user-specific, distance-preserving coordinate transformations to all location data shared with servers. Shen et al. [28] and Yi et al. [29] also presented two mechanisms based on additive homomorphic encryption to protect location privacy in spatial crowdsourcing and kNN queries, respectively. The authors of [30] and [31] introduced the public-key encryption and attribute-based encryption to protect the request information from being leaked to the untrusted service provider. However, these mechanisms require to change the original architecture of LBSs and also consume much more computational overhead to encrypt and decrypt the privacy information. As a result, these methods usually obtain a low efficiency.

In the last few years, differential privacy has got a rapid development and several variants or generalizations of that have been studied. However, applying differential privacy for location protection has not been investigated in depth. Andrés et al. [4] presented a mechanism for achieving geo-indistinguishability by adding controlled random noise to user's location. To protect the location privacy of workers participating in spatial crowdsourcing (SC), To et al. [32]

proposed a mechanism based on differential privacy and geo-casting that achieves effective SC services while offering privacy guarantees to workers. However, these similar methods are only suitable for sporadic LBSs, but not for continuous using, which will lead to a quick loss of privacy. Although the authors of [33] proposed a predictive mechanism to protect users' location privacy continuously, it is only suitable for the low-speed transport model, just like walking. Additionally, Xiao et al. [1] and He et al. [34] also extended differential privacy in a new setting of trajectory sharing and publishing. But unfortunately, both of them only suit for trajectory sharing and cannot be used to requeste POIs in continuous LBSs. So, until now there have been limited effective works which utilize the differential privacy to protect location privacy continuously. However, our AGENT can solve the above predicament perfectly and obfuscate the same trace with less privacy budget.

3 Preliminaries

For your convenience, we present some preliminaries that serve as the basis of AGENT in this section.

3.1 Differential privacy and geo-indistinguishability

Differential privacy [3] is a notion of privacy from the area of statistical database. Its goal is to protect an individual's data while publishing aggregate information about the database. Differential privacy requests that modifying a single user's data should have a negligible effect on the query outcome. The privacy definition used in our AGENT is based on a generalized variant of differential privacy that can be defined on an arbitrary set of secrets χ, equipped with a metric d_χ [35]. The distance $d_\chi(x, x')$ expresses the distinguishability level between the secrets x and x', modeling the privacy notion that we want to achieve. A small value denotes that the secrets should remain indistinguishable, while a large one means that we allow the adversary to distinguish them.

Let $\mathcal{Z}$ be a set of values reported to service providers and let $\mathcal{P}(\mathcal{Z})$ denote the set of probability measures over $\mathcal{Z}$. The multiplicative distance $d_\mathcal{P}$ on $\mathcal{P}(\mathcal{Z})$ is defined as:

$$d_\mathcal{P}(\mu_1, \mu_2) = sup_{Z \in \mathcal{Z}} \left| \ln \frac{\mu_1(Z)}{\mu_2(Z)} \right|$$

where $\mu_1(Z)$ and $\mu_2(Z)$ are the posterior probabilities that the reported locations belong to the set $Z \in \mathcal{Z}$ when users' locations are x and x'. Intuitively, $d_\mathcal{P}(\mu_1, \mu_2)$ is small if μ_1, μ_2 assign similar probabilities to each reported value.

A mechanism is designed as a probabilistic function $K : \chi \to \mathcal{P}(\mathcal{Z})$, assigning to each secret x a probability distribution $K(x)$ over the reported values $\mathcal{Z}$. The generalized variant of differential privacy, called d_χ-privacy, is defined as follows [4]:

Definition 1 (d_χ-privacy). A mechanism $K : \chi \to \mathcal{P}(\mathcal{Z})$ satisfies d_χ-privacy if:

$$d_\mathcal{P}(K(x), K(x')) \leq d_\mathcal{X}(x, x'), \forall x, x' \in \mathcal{X}$$

or equivalently $K(x)(Z) \leq e^{d_\mathcal{X}(x,x')} K(x')(Z), \forall x, x' \in \mathcal{X}$, $Z \subseteq \mathcal{Z}$.

Different choices of d_χ give rise to different privacy notions, it is also common to scale this metric of interest by a privacy parameter ϵ which is called privacy budget (note that ϵd_χ is itself a metric).

However, the main motivation of this paper is the location privacy. In this case, the secrets χ as well as the reported values $\mathcal{Z}$ are all the sets of locations, while K is an obfuscation mechanism. Using the Euclidean metric d_2, we obtain ϵd_2-privacy, a natural notion of location privacy called geo-indistinguishability in [4]. If for any radius d_2, a mechanism makes the user enjoy ϵd_2-privacy within d_2, we claim that the mechanism satisfies ϵ-geo-indistinguishability. As the definition, the closer (geographically) two locations are, the more similar the probability of producing the same reported location z should be. Through the mechanism K, the service provider cannot infer the user's location accurately, but he can obtain approximate information required to provide the service.

While protecting the location traces, we denote a trace as $\Gamma = [x_1, \ldots, x_{|\Gamma|}]$. In order to sanitize Γ, the geo-indistinguishable mechanism, expressed as $N(\epsilon_N)$, can be simply applied to each secret x_i. We assume that a family of obfuscated mechanisms $N(\epsilon_N) : \chi \to \mathcal{P}(\mathcal{Z})$ are available, parametrized by $\epsilon_N > 0$, where each mechanism $N(\epsilon_N)$ satisfies ϵ_N-privacy. However, the simple application may bring a serious problem which is explained in the introduction, the entire obfuscated mechanism is $n\epsilon_N d_2$-private, that is, the privacy budget consumed increases linearly with n.

3.2 Utility

The goal of a privacy mechanism is to hide the privacy and disclose enough useful information for the service. Typically, these two aspects go in opposite directions: a stronger privacy level requires more noise which results in a lower utility.

To measure the utility, we first define a notion of error which is a distance d_{err} between the secret trace Γ and a sanitized trace Z. In LBSs, a location is wanted to report as close as possible to the original one. So a natural choice

is to define the error as the average geographical distance between the locations in the trace:

$$d_{err}(\Gamma, Z) = \frac{1}{|\Gamma|} \sum_i d(x_i, z_i)$$

To find the minimum error, we consider the probability that it achieves, commonly expressed in the form of $\alpha(\delta)$-useful [36]. If a mechanism K is $\alpha(\delta)$-useful for all $x \in \Gamma$, then $\Pr[d_{err}(\Gamma, Z) \leq \alpha] \geq \delta$.

When requesting the service continuously through AGENT, we design a test mechanism $\Lambda(l, \epsilon_\theta, k)$ to guarantee the utility as similar with [33]. The test mechanism takes as input the secret x_i and reports whether the sanitized location z_i is acceptable or not for this secret. If the test is successful, then the sanitized z_i will be used instead of generating a new one. When searching the sanitized locations, if the distance between z_j and secret x_i satisfies the following condition:

$$d(x_i, z_j) \leq l + Lap(\varepsilon_\theta)$$

location z_j will be accepted as the sanitized location for x_i. If not, we have to spend some privacy budget to generate a new one.

Since the test is accessing the secret x_i, it should be private itself and added Laplace noise to the threshold l, where ϵ_θ is the budget that is allowed to be spent for test. Additionally, we also introduce a $|\Gamma| \times 2$ matrix $\mathbf{B}$ to store the results of tests. For example, if z_j is accepted as the sanitized location for secret x_i with k_i tests, we will set $b_i[0] = k_i$, $b_i[1] = 0, b_i \in \mathbf{B}$, $k_i \leq k$, which k is a threshold to limit the number of test for x_i. And if z_j is rejected with k_i tests, $b_i[0], b_i[1]$ will be set to k_i and 1, respectively. It is to be noted that $k * \epsilon_\theta < \epsilon_N$, the reason is that the sanitized mechanism is always more expensive than the test.

4 System overview and problem statement

As discussed above, we designed AGENT to help users get an adaptive location-privacy preservation with less privacy budget consumption in continuous LBSs. In this section, we present the system model of AGENT and then emphasize the privacy problem with the disclosure of users' locations.

4.1 System model

Before describing the details, we show the system model of AGENT as Fig. 1.

From Fig. 1, we derive three basic components of our adaptive geo-indistinguishability system: GPS module, AGENT, and service provider. As a part of the service

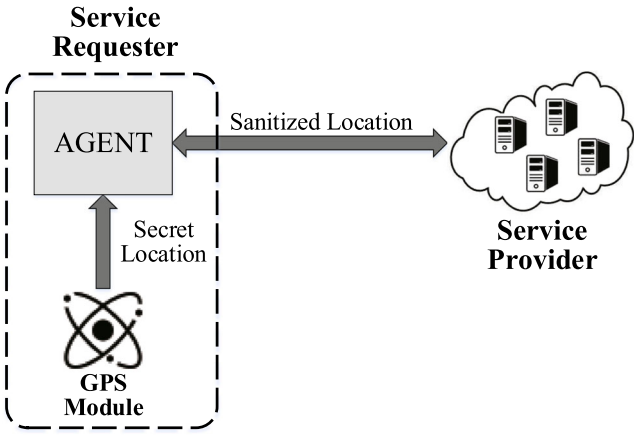


Fig. 1 The system model of AGENT

requester, GPS module provides secret locations for requester when he asks for LBSs. As shown in Fig. 1, the system works as follows:

- While asking for LBSs, the requester first generates a secret location by the GPS module and then sends it to the AGENT.
- After that, the AGENT obfuscates the secret location. In AGENT, if there is an existed sanitized location finding that meets the condition of test, the requester will select it as the obfuscated location instead of generating a new one.
- Finally, the requester sends the sanitized location to service provider and get the service results.

Notably, during the procedure, the service requester always does not disclosure his secret locations to the service provider, and only the sanitized ones will be sent. So, the location privacy is not leaked when he requests the service.

4.2 Problem statement

In LBSs, service providers are curious-but-honest, who are interested in requesters' privacy information. When requesting the service, requesters will send their locations to service providers. Since the location information that attached to the request information are considered as the privacy of requesters, it will be leaked if it is not obfuscated. If their secret locations are sent to the service providers, the requesters may be tracked by the adversary and face the dangerous conditions. Moreover, a Laplace-based obfuscation mechanism satisfying the geo-indistinguishability works well in the case of a sporadic use, under repeated use, however, independently applying noise leads to a quick loss of privacy due to the correlation between the locations in the trace.

4.3 Design Goals

As an adaptive geo-indistinguishable mechanism, AGENT should fulfill the following requirements.

- **Quality of Service(QoS).** The proposed mechanism should disclose enough useful information for the LBSs to guarantee the utility while protecting privacy.
- **Privacy Preservation.** The proposed mechanism should protect user's location privacy. When requesters request the LBSs, the AGENT should prevent users' exact locations from leaking to the service provider and other adversaries.

5 The adaptive geo-indistinguishability

While requesting the LBSs repeatedly, independently applying differential privacy mechanism leads to a quick loss of privacy due to the correlation between the locations in the trace. To solve the problem above, Chatzikokolakis et al. [33] proposed a prediction mechanism that tried to guess the new sanitized location based on the previously reported locations. However, their mechanism could not solve the problem perfectly, especially when requesters travel at a high speed. Under this circumstance, their mechanism will be disabled and an example is shown in Fig. 2.

As shown in Fig. 2, when asking for the service at location x_1 and x_2, the requester will send the sanitized location z_1 to service provider. Then, while he requests the service at location x_3, the sanitized location z_1 cannot satisfy the requirements of privacy and utility as in the prediction mechanism. So he needs to generate a new one z_2 and sends it to service provider. Next, when he asks for the service at location x_4, location z_2 does not satisfy the privacy requirement for location x_4 and another sanitized location z_3 must be generated and sent to service provider. However, as shown in Fig. 2, although z_2 does not satisfy the conditions of test mechanism in [33], location z_1 meets the condition for x_4. So the mechanism in [33] may be disabled in this case.

To solve the problem, we design an adaptive geo-indistinguishable mechanism, named AGENT, in which a R-tree is introduced to realize the reuse of sanitized locations and a parameter k is also introduced in test mechanism to achieve the adaptive privacy preservation with different transportation models. In this section, we present the detail of AGENT and the overall procedure is shown in Fig. 3.

5.1 Initializing R-tree and searching sanitized location

Before obfuscating the locations by AGENT, a sanitized R-tree should be first initialized. Then, when asking for the LBSs, the requester will search the R-tree to get an available sanitized location.

5.1.1 Initializing the R-tree

In AGENT, a R-tree spatial decomposition technique is introduced to store and index the sanitized locations. Each sanitized location which is generated by the obfuscation mechanism will be stored in the R-tree. We take the area of interest as a minimum bounding rectangle (MBR) in R-tree. The spatial decomposition mechanism is defined as $\Theta(n) : S \rightarrow RT(\mathcal{Z})$, parameterized by n. In the mechanism, n represents the maximum number of sanitized locations which are allowed to store in each MBR , S represents the plane in coordinate system, and $RT(\mathcal{Z})$ represents the R-tree which

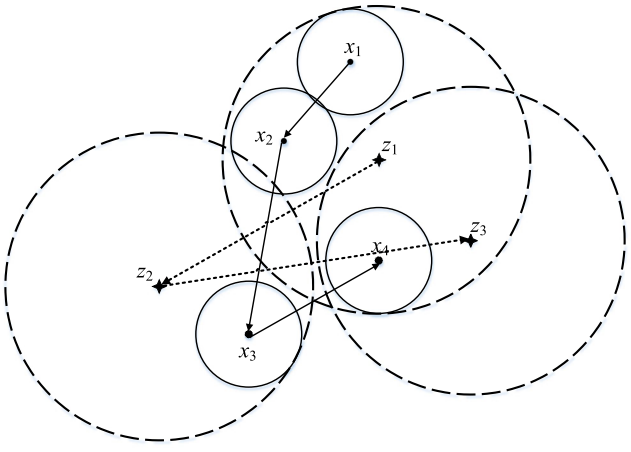


Fig. 2 The prediction mechanism for continuous LBS

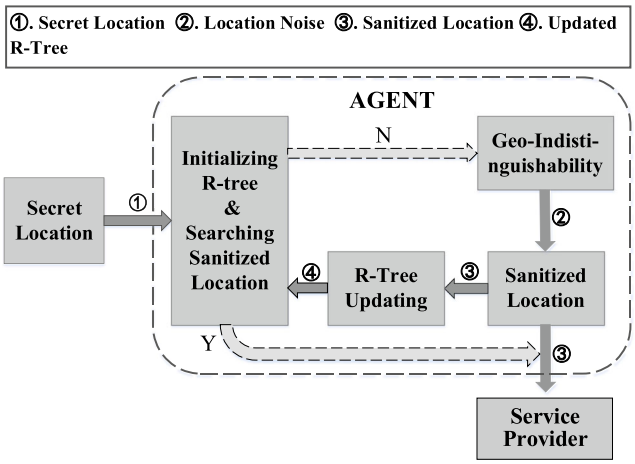


Fig. 3 Overall AGENT procedure

stores the sanitized locations $\mathcal{Z}$. In the sanitized R-tree, the root MBR represents the entire area of interest, where all the sanitized and secret locations locate.

In the R-tree, the head of each node stores the coordinate of corresponding MBR, such as $< x_{i\_begin}, x_{i\_end}, y_{i\_begin}, y_{i\_end} >$, where x_{i_begin} represents the begin coordinate of x-axis for the i-th MBR, x_{i_end} represents the end coordinate of x-axis for the i-th MBR, y_{i_begin} and y_{i_end} are the begin and end coordinates of y-axis for i-th MBR, respectively. If the node is a leaf node, it will store the grid coordinates and the coordinates of sanitized locations. If not, it will store the coordinates and the pointers of corresponding child nodes.

While some existed sanitized locations need to be stored into the R-tree, we should first obtain the corresponding leaf nodes which the sanitized locations lie in. Then, if there is enough space to store them, they will be inserted into the leaf nodes. In AGENT, we set threshold n as the maximum locations which each leaf node can store. If there is not enough space to insert a sanitized location, we need to split the node into two child nodes and set them as the left-child one and right-child one, respectively. After splitting, all the sanitized locations which are stored in the father node and the new inserted location will be reinserted into the new child nodes according their coordinates. We split the node according the rules in the following:

- Comparing the size of intervals of x-axis and y-axis
- If the intervals satisfy the following condition:

$$|y_{i\_end} - y_{i\_begin}| \leq |x_{i\_end} - x_{i\_begin}|,$$

  we split the node on x-axis equally and get the grid coordinates of left-child and right-child nodes as follows: $< x_{i\_begin}, (x_{i\_begin} + x_{i\_end})/2, y_{i\_begin}, y_{i\_end} >$ and $< (x_{i\_begin} + x_{i\_end})/2, x_{i\_end}, y_{i\_begin}, y_{i\_end} >$
- If the intervals satisfy the following condition:

$$|y_{i\_end} - y_{i\_begin}| \geq |x_{i\_end} - x_{i\_begin}|,$$

  we split the node on y-axis equally and get the grid coordinates of left-child and right-child nodes as follows: $< x_{i\_begin}, x_{i\_end}, y_{i\_begin}, (y_{i\_begin} + y_{i\_end})/2 >$ and $< x_{i\_begin}, x_{i\_end}, (y_{i\_begin} + y_{i\_end})/2, y_{i\_end} >$

Finally, as an example, we can get an initialized R-tree which is shown in Fig. 4. In Fig. 4, the root node R_0 represents the entire area of interest, in which there are four sanitized locations: z_1, z_2, z_3, and z_4. If the parameter n is 3, the root node cannot store all the sanitized locations and it must split into two child nodes: R_1 and R_2. After that, sanitized location z_4 is stored in node R_2 and other locations z_1, z_2, and z_3 stay in node R_1.

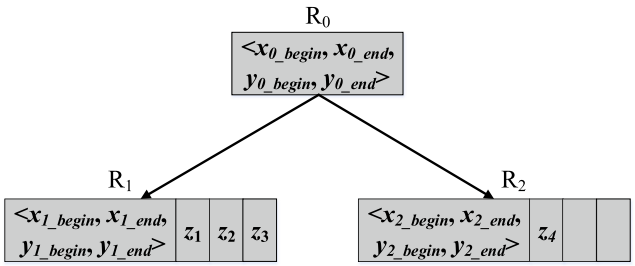


Fig. 4 An initialized sanitized R-tree

5.1.2 Searching the sanitized location

While asking for LBSs, the requester first traverses the R-tree to search an available sanitized location. If such a location is sought, the requester will select it as the sanitized location instead of generating a new one. For example, if the requester asks for the service at position $x :< x_{r\_x}, y_{r\_x} >$, he will search the location as following steps:

Step-1 traversing the sanitized R-tree and finding out a leaf node R_i which the secret x locates in: $x_{i\_begin} \leq x_{r\_x} \leq x_{i\_end}$ and $y_{i\_begin} \leq y_{r\_x} \leq y_{i\_end}$.

Step-2 sorting the sanitized locations in R_i with the following rules: if $|y_{i\_end} - y_{i\_begin}| \leq |x_{i\_end} - x_{i\_begin}|$, we sort the locations according the x coordinate. If $|y_{i\_end} - y_{i\_begin}| \geq |x_{i\_end} - x_{i\_begin}|$, we sort the locations according the y coordinate.

Step-3 searching the sanitized location by binary search. We also introduce a threshold k to limit the number of test and it can be changed according the transportation model adaptively. If a sanitized location z_i is accepted with k_i times, $k_i \leq k$, which satisfies $d(x, z_i) \leq l + Lap(\varepsilon_\theta)$, we will return z_i and set $b_i[0] = k_i$, $b_i[1] = 0$, otherwise return null and set $b_i[0] = k_i$, $b_i[1] = 1$.

Step-4 if there is an existed sanitized location z_i satisfying the test condition for secret x, the requester will accept it as the sanitized location for x. If not, the requester must take some extra privacy budget to generate a new one and insert it into the sanitized R-tree.

Step-5 after obtaining the sanitized location, the requester sends it to the service provider and filters out the points which he needs from the feedbacks.

5.2 Generating sanitized locations

While there is not an existed sanitized location for secret x, the requester must generate a new one with some privacy

budget. In AGENT, we adopt the geo-indistinguishable mechanism [4] as the main obfuscation method. Given a secret x and privacy budget ε_N, the requester generates a sanitized location as following steps:

Step-1 transforming the plane coordinate to polar coordinate.

Step-2 drawing θ uniformly in $[0, 2\pi]$.

Step-3 drawing p uniformly in [0,1] and set

$$r = -\frac{1}{\varepsilon_N}(W_{-1}(\frac{p-1}{e}) + 1)$$

where W_{-1} is the Lambert W function (the -1 branch).

Step-4 computing the sanitized location as follows:

$$z = x + < r\cos(\theta), r\sin(\theta) > .$$

5.3 Updating the sanitized *R*-tree

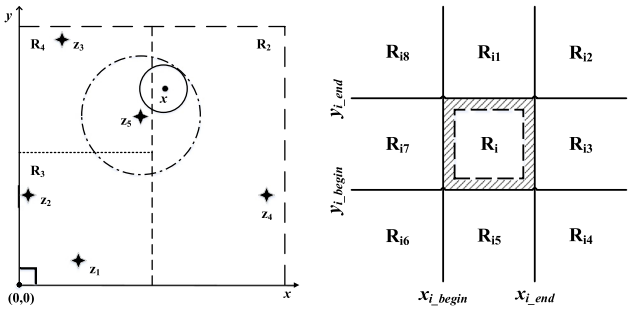
When a new sanitized location is generated, we need to insert it into the constructed R-tree and to update its structure. If there is a new generated sanitized location z_5, we will insert it into the initialized R-tree in Fig. 4 to show how to update the existed structure. We assume that the location z_5 lies in the node R_1. However, R_1 has not enough space to store a new location. Hence, we split the node R_1 into two child nodes: R_3 and R_4. While splitting R_1, we assume $|y_{1\_end} - y_{1\_begin}| \geq |x_{1\_end} - x_{1\_begin}|$, so the node R_1 will be split on y-axis and the coordinates of child nodes are as follows:

$$R_3 : \begin{cases} x_{3\_begin} = x_{1\_begin} \\ x_{3\_end} = x_{1\_end} \\ y_{3\_begin} = y_{1\_begin} \\ y_{3\_end} = (y_{1\_begin} + y_{1\_end})/2 \end{cases}$$

$$R_4 : \begin{cases} x_{4\_begin} = x_{1\_begin} \\ x_{4\_end} = x_{1\_end} \\ y_{4\_begin} = (y_{1\_begin} + y_{1\_end})/2 \\ y_{4\_end} = y_{1\_end} \end{cases}$$

Then, sanitized locations z_1 and z_2 lie in node R_3, and z_3 and z_5 lie in node R_4. Obviously, the two nodes have enough space to store the locations. However, to make full use of the existed sanitized locations, some specific conditions must be considered. If the requester asks for the service at the border of a MBR, the sanitized location which locates in the adjacent MBR may be available and an example is shown in Fig. 5a.

As shown in Fig. 5a, if the requester asks for the service at location x and searches the sanitized location with the R-tree which is constructed above, he will find that none of the existed sanitized locations in node R_2 satisfy the requirements of utility. However, it does exist a sanitized location z_5 that can be used to instead of generating a new one. To solve the above problem, we introduce a parameter $\eta(0 < \eta < 1)$ which represents the proportion of mutual coverage of the adjacent MBRs. For example, in Fig 5a, we assume that the coordinate of z_5 is $< x_{z_5}, y_{z_5} >$ and $x_{z_5} < x_{4\_end}$, but it also satisfies the following condition:



(a) An available location in adjacent MBR (b) The adjacent MBRs of R_i

Fig. 5 The adjustment of location in adjacent MBRs

$$x_{z_5} \geq x_{4\_end} - \eta(x_{4\_end} - x_{4\_begin}).$$

We will insert z_5 into its adjacent MBR R_2. So we redefine the R-tree spatial decomposition mechanism as $\Theta(\eta, n) : S \rightarrow RT(\mathcal{Z})$, parameterized by η and n, and modify the rules of update as follows:

- Traversing the R-tree to search a leaf node to store the new generated sanitized location z_i. If the leaf node has enough space to store it, we will insert it into the node directly. If not, we will split the node and insert z_i into its corresponding child node. We assume that the node which contains z_i is R_i.
- Generally, we assume that R_i has 8 adjacent MBRs and their positions are shown in Fig. 5b.
- Computing the following equations:

$$\begin{cases} x_{l\_bound} = x_{i\_begin} + \eta(x_{i\_end} - x_{i\_begin}) \\ x_{u\_bound} = x_{i\_end} - \eta(x_{i\_end} - x_{i\_begin}) \\ y_{l\_bound} = y_{i\_begin} + \eta(y_{i\_end} - y_{i\_begin}) \\ y_{u\_bound} = y_{i\_end} - \eta(y_{i\_end} - y_{i\_begin}) \end{cases}$$

- According to the following rules, z_i will be inserted into different nodes:
  - If $x_{l\_bound} \leq x_{z_i} \leq x_{u\_bound}$ and $y_{z_i} \geq y_{u\_bound}$, the location z_i will also be inserted into R_{i1};
  - If $x_{z_i} \geq x_{u\_bound}$ and $y_{z_i} \geq y_{u\_bound}$, the location z_i will also be inserted into R_{i1}, R_{i2}, and R_{i3};

- If $x_{z_i} \geq x_{u\_bound}$ and $y_{l\_bound} \leq y_{z_i} \leq y_{u\_bound}$, the location z_i will also be inserted into R_{i3};
- If $x_{z_i} \geq x_{u\_bound}$ and $y_{z_i} \leq y_{l\_bound}$, the location z_i will also be inserted into R_{i3}, R_{i4}, and R_{i5};
- If $x_{l\_bound} \leq x_{z_i} \leq x_{u\_bound}$ and $y_{z_i} \leq y_{l\_bound}$, the location z_i will also be inserted into R_{i5};
- If $x_{z_i} \leq x_{l\_bound}$ and $y_{z_i} \leq y_{l\_bound}$, the location z_i will also be inserted into R_{i5}, R_{i6}, and R_{i7};
- If $x_{z_i} \leq x_{l\_bound}$ and $y_{l\_bound} \leq y_{z_i} \leq y_{u\_bound}$, the location z_i will also be inserted into R_{i7};
- If $x_{z_i} \leq x_{l\_bound}$ and $y_{z_i} \geq y_{u\_bound}$, the location z_i will also be inserted into R_{i7}, R_{i8}, and R_{i1};

According to the new rules, after being inserted the new generated location z_5, the sanitized tree in Fig. 4 can be updated as shown in Fig. 6. The details of the process are listed in Algorithm 1.

6 Privacy and utility analysis

In this section, we theoretically analyze the privacy preservation and utility which AGENT achieves.

6.1 Privacy of AGENT

We now proceed to show that our AGENT described above is d_χ-private, which depends on the privacy of its components. In the following, we assume that the test and sanitized mechanisms are both d_χ-private with the corresponding privacy budget:

$$\forall l, \varepsilon_\theta, k.\Lambda(l, \varepsilon_\theta, k) \text{ is } \varepsilon_\theta d_\chi - private$$
$$\forall \varepsilon_N.N(\varepsilon_N) \text{ is } \varepsilon_N d_\chi - private$$

We can show that the test mechanism $\Lambda(l, \epsilon_\theta, k)$, equipped with a Laplacian noise generation function *Lap* scaled by ϵ_θ, is indeed $\epsilon_\theta d_\chi$-private, independently of the metric or threshold used.

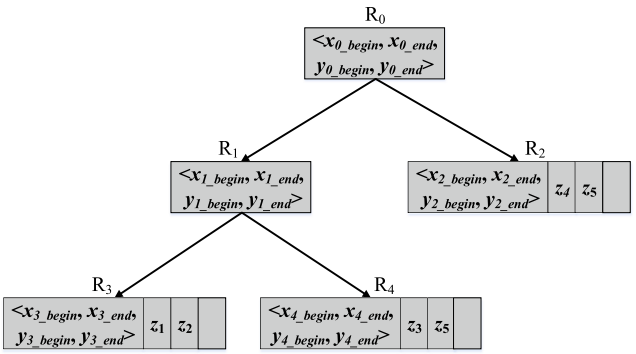


Fig. 6 An updated sanitized R-tree

Algorithm 1 Process of AGENT for Continuous LBSs

Require: Current location of requester $x :< x_{r\_x}, y_{r\_x} >$, the privacy budget for obfuscation and test mechanism: ϵ_N and ϵ_θ, parameters η, n for spatial decomposition mechanism, threshold l and k.

Ensure: The sanitized location z_i for input x.

Initialize the sanitized R-tree
while $true$ **do**
  Requester asks for the LBS with location x
  Find out the leaf node R_i that cover the secret x
  Sort the sanitized locations in node R_i
  Search a sanitized location z_i that satisfies the following condition within k times:

$$d(x, z_i) \leq l + Lap(\varepsilon_\theta)$$

  if z_i is not $null$ **then**
    Send z_i to service provider
  else
    Generate a location noise $< r, \theta >$ from polar Laplacian with privacy budget ϵ_N
    Compute the sanitized location $z_i = x + < r\cos(\theta), r\sin(\theta) >$
    if R_i has enough space to store z_i **then**
      Insert z_i into node R_i
    else
      Split node R_i into two child nodes
      Insert z_i into corresponding child node
    end if
    Insert z_i into adjacent nodes
    Send z_i to service provider
  end if
end while

The global privacy budget for a certain trace Γ is defined as:

$$\varepsilon(\Gamma) = \begin{cases} 0 & if|\Gamma| = 0 \\ \varepsilon_\theta b[0] + \varepsilon_N b[1] + \varepsilon(tail(\Gamma)) o.w. \end{cases}$$

Building on the privacy properties of its components, we obtain that AGENT satisfies a property similar to d_χ-privacy, with a parameter ϵ that depends on the trace.

Theorem 1 *Under the assumptions, for the test and sanitized mechanisms, the mechanism of AGENT Ψ satisfies*

$$\Psi(x)(\Gamma) \leq e^{\varepsilon(\Gamma)d(x,x')}\Psi(x')(\Gamma) \qquad \forall \Gamma, x, x'$$

Proof To prove the privacy preservation of AGENT, we just to prove that:

$$\forall x, x'. \qquad P[\Gamma|x] \leq e^{\epsilon(\Gamma)d(x,x')}P[\Gamma|x']$$

Analyzing the single step, we have a binary choice between the successful test, which is deterministic, and the failed case, which is probabilistic. Given a trace Γ, we reorganize the indexes of its steps in two groups, the successful $I_S = \{i|b_i[1] = 0\}$ and the failed steps $I_F = \{i|b_i[1] = 1\}$. After having the assumptions for the test and sanitized mechanisms, we regroup the exponents as follows:

Let Γ_n represents the trace after n steps, x_n represents the n-th secret in trace Γ_n, so we obtain the following equations:

$$\begin{aligned}\forall x\ P[\Gamma_n|x_n] &= P[\Gamma_n|\Gamma_{n-1}, x_n] * P[\Gamma_{n-1}|x_{n-1}] \\ &= \prod_{i=1}^{n} P[\Gamma_i|\Gamma_{i-1}, x_i]\end{aligned}$$

When generating the trace Γ_i from Γ_{i-1}, we need to select or generate a sanitized location z_i to form the i-th step. Since b_i is related to the formation of z_i, we evolve the above equation into following:

$$\begin{aligned}\forall x\ P[\Gamma_n|x_n] &= \prod_{i=1}^{n} P[\Gamma_i|\Gamma_{i-1}, x_i] \\ &= \prod_{i=1}^{n} P[z_i|\mathrm{Z}_{i-1}, b_i, x_i]P[b_i|\Gamma_{i-1}, x_i] \\ &= \prod_{i\in I_S(\Gamma)} P[z_i|\mathrm{Z}_{i-1}, b_i, x_i]P[b_i|\Gamma_{i-1}, x_i] * \\ &\quad \prod_{i\in I_F(\Gamma)} P[z_i|\mathrm{Z}_{i-1}, b_i, x_i]P[b_i|\Gamma_{i-1}, x_i]\end{aligned}$$

If the i-th step belongs to the successful group, we obtain the sanitized location z_i with probability 1. If it belongs to the failed one, z_i can only be obtained by the sanitized mechanism with probability $P[z_i|x_i]$. So we evolve the above equation continuously:

$$\begin{aligned}\forall x, x'\ P[\Gamma_n|x_n] &= \prod_{i\in I_S(\Gamma)} P[z_i|\mathrm{Z}_{i-1}, b_i, x_i]P[b_i|\Gamma_{i-1}, x_i] * \\ &\quad \prod_{i\in I_F(\Gamma)} P[z_i|\mathrm{Z}_{i-1}, b_i, x_i]P[b_i|\Gamma_{i-1}, x_i] \\ &= \prod_{i\in I_S(\Gamma)} 1 * P[b_i[1] = 0|\Gamma_{i-1}, x_i] * \\ &\quad \prod_{i\in I_F(\Gamma)} P[z_i|x_i]P[b_i[1] = 1|\Gamma_{i-1}, x_i] \\ &= \prod_{i\in I_S(\Gamma)} e^{b_i[0]\varepsilon_\theta d(x_i,x_i')} * P[b_i[1] = 0|\Gamma_{i-1}, x_i'] \\ &\quad * \prod_{i\in I_F(\Gamma)} \begin{matrix} e^{\varepsilon_N d(x_i,x_i')}P[z_i|x_i']e^{b_i[0]\varepsilon_\theta d(x_i,x_i')} * \\ P[b_i[1] = 1|\Gamma_{i-1}, x_i'] \end{matrix} \\ &= e^{\varepsilon(\Gamma)} \prod_{i\in I_S(\Gamma)} P[b_i[1] = 0|\Gamma_{i-1}, x_i'] * \\ &\quad \prod_{i\in I_F(\Gamma)} P[z_i|x_i']P[b_i[1] = 1|\Gamma_{i-1}, x_i'] \\ &= e^{\varepsilon(\Gamma)}P[\Gamma_n|x_n']\end{aligned}$$

With a global exponent for AGENT:

$$\varepsilon(\Gamma) = (\sum_{i\in I(\Gamma)} b_i[0]\varepsilon_\theta + \sum_{i\in I_F(\Gamma)} \varepsilon_N) * d(x, x')$$

□

This result shows that there is a difference between the budget spent on a "good" trace, where the input has a considerable correlation, and that spent on a "bad" trace, where the input has uncorrelated secrets. For a "good" trace, AGENT performs well and the majority tests are successful. Conversely, for an uncorrelated trace, AGENT is useless and all tests are failing. In this case, it is clear that AGENT wastes part of its budget on the tests that always fail, performing worse than an independent mechanism.

6.2 Utility analysis

Next, we analyze the utility achieved by AGENT. In AGENT, we just want to show that it satisfies $\alpha(\delta)$-useful, which is introduced in Section 3. According the utility properties of its components, we show that it depends on the utility of noise mechanism, as well as the test condition when searching the sanitized locations. Therefore, we can derive a result about the utility of a single step for AGENT.

Proposition 1 (Utility) *Let* Γ *be a trace and let* $\alpha_N(\delta)$, $\alpha_\theta(\delta)$ *be the utility of* $N(\epsilon_N)$ *and* $Lap(\epsilon_\theta)$, *respectively. Then, we get the utility of Step*(Γ) *is* $\alpha(\delta) = max\{\alpha_N(\delta), l + \alpha_\theta(\delta)\}$.

This result gives a bound for the utility of AGENT at each step when requesting the service continuously. And the bound depends on the privacy budget ϵ_N, ϵ_θ and threshold l. In our mechanism, we give both noise mechanism and test mechanism the same utility: $\alpha_N(\delta) = l + \alpha_\theta(\delta)$. In this manner, if the requester gets an existed sanitized location z_i from the sanitized tree, the utility of this step is $l+\alpha_\theta(\delta)$ and location z_i satisfies $d(x, z_i) \leq l + Lap(\varepsilon_\theta)$ with probability 1.

7 Experimental evaluation

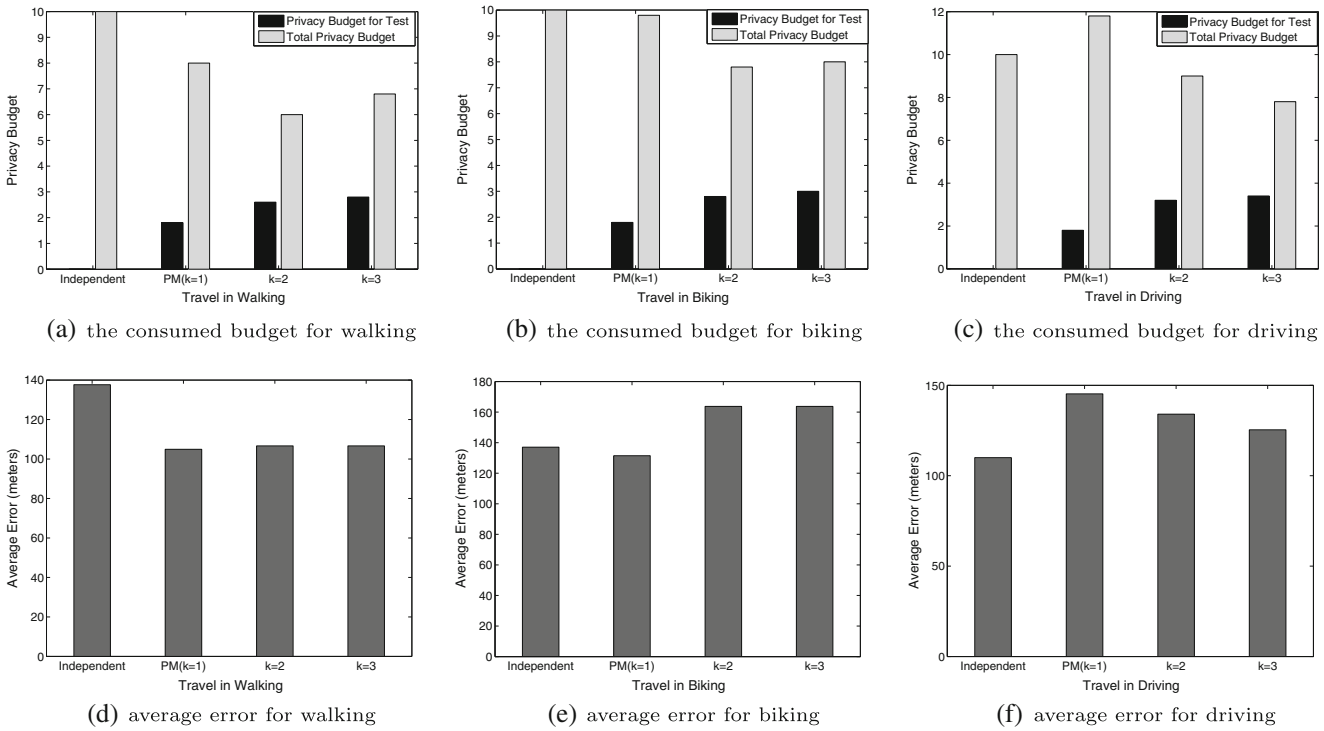
In this section, we present series of empirical results of AGENT conducted over the real-word dataset which is well known as GeoLife [37]. To evaluate the effectiveness of AGENT, we obfuscate three traces which travel in different transportation modes with the same utility and compare the privacy budgets which they consume for test and generating noise.

7.1 Evaluation setup

In order to configure the geo-indistinguishable application, we first define a radius $r^\star$ where we wish to be protected, that we assume is 100 meters, and then the privacy budget for generating noise, $\epsilon_N^\star$, to be $\ln 6$. This means that taken two points on the radius of 100 meters, their probability of being the observables of the same secret differ at most by 6, and even less the more we take them closer to the secret. To ensure $k * \epsilon_\theta^\star < \epsilon_N^\star$ while k is changing, we set the privacy budget for each test, $\epsilon_\theta^\star$, as $\ln 6/5$. During the simulation, we also set the parameters $\eta = 0.1$, $n = 3$, and the threshold l which is used to fix the utility in test mechanism as 100 meters. When requesting a service, we choose a large confidence factor for utility, say, 0.95.

7.2 Evaluation results

To evaluate the effectiveness of AGENT, we assume that the user performs several activities while moving around the city throughout a day, possibly using different means of transport. Firstly, we select three different traces from GeoLife in different transportation modes: walking, biking, and driving. Then, we evaluate the consumed privacy budget



(a) the consumed budget for walking (b) the consumed budget for biking (c) the consumed budget for driving

(d) average error for walking (e) average error for biking (f) average error for driving

Fig. 7 the evaluation results for three different traces

(including the privacy budget for testing the sanitized location z_i and the privacy budget for test and generating new sanitized locations) and the average errors of AGENT by changing the parameter k in different transportation models. After that, given the same utility, we also compare the evaluation results with the independent mechanism which generates a new sanitized location for each secret location and the predicted mechanism [33] which is the condition of $k = 1$ in our AGENT. The simulation results are shown in Fig. 7.

While the user travels at a low speed, such as walking, the simulation results in Fig. 7a show that the PM consumes less privacy budget than the independent one, and that is also indicated in [33]. However, when using the AGENT and set parameter $k = 2$ and 3, the simulation results show that our mechanism consumes less privacy budget. So, given some budget, the user can translate more points by AGENT when requesting the service. In Fig. 7a, the results also show that although our AGENT takes more privacy budget to test the availability of sanitized locations, we will consume less that to generate new ones. Finally, compared with the independent mechanism and PM, our mechanism will spend less budget for the same trace to obtain the same utility.

When the user travels at a medium speed, such as biking, the PM does not perform any advantages compared with the independent mechanism. The simulation results in Fig. 7b show that PM almost spends same privacy budget with the independent one. When obfuscating the trace with AGENT and setting parameter $k = 2$ and 3, we spend less budget with the same utility. As is also shown in Fig. 7b, we also spend more budget to test in exchange for less generation of new sanitized locations.

For the last transportation model, when the user travels at a high speed, such as driving, the simulation results in Fig. 7c show that the PM performs a worse performance than the independent mechanism. The reason is that the user wastes some budget for tests and they always fail. So the user must spend some additional budget on generating new sanitized locations. Compared with the independent mechanism and PM, the simulation results show that given the same utility, our AGENT consumes less privacy budget when the user travels at such a high speed, especially set parameter $k = 3$.

In summary, we can get the conclusion that with the same utility, AGENT consumes less budget than PM and independent mechanism on the same trace. Especially, when traveling at a higher speed, users may need to choose a bigger parameter k to get more chance to test adaptively.

After that, we also test the average error of AGENT while k is changing. As shown in Fig. 7d, e, f, the simulation results indicate that the average errors of AGENT and independent mechanism are the same magnitude and only have little difference. The reason is that we set the same utility between the noise and test mechanisms. If we set the test mechanism to own a higher utility than the noise, our AGENT will obtain less error than the independent mechanism. But that will also consume much more privacy budget on test.

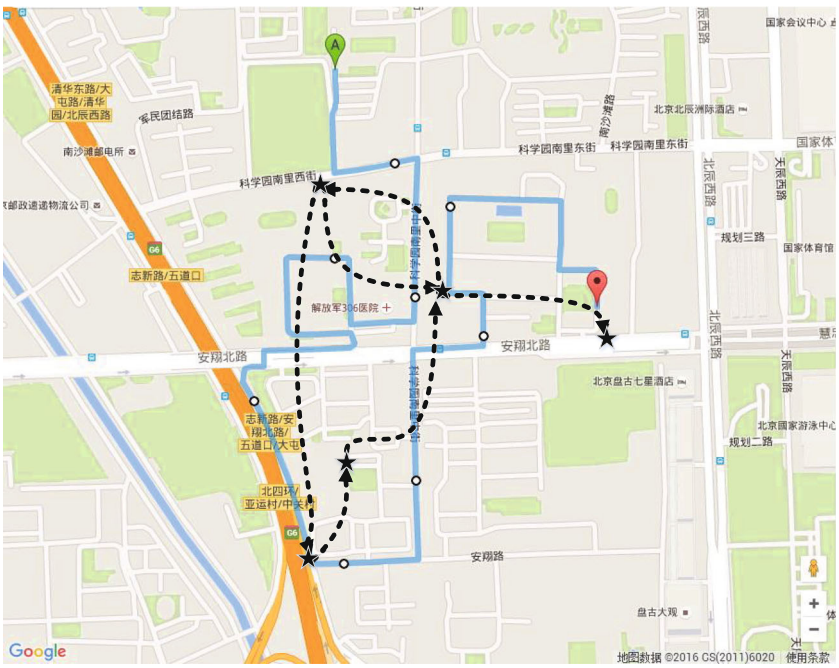


Fig. 8 Original trace (*full line*), secret locations (*circle point*), and sanitized locations (*star point*)

Figure 8 displays one of Geolife trajectories sanitized with AGENT. The original trace, in full line, starts point *A* with traveling in biking. During the traveling, we assume the user launches 10 requests at the secret locations, including the start and end points. Finally, we have the reported trace, in dashed line, with only 5 sanitized locations, in star point.

8 Conclusion

The disclosure of user locations seriously threatens users' personal privacy when requesting the LBSs. In this paper, we present a novel solution, called AGENT, to address the privacy preservation in continuous LBSs. For AGENT, we introduce the test mechanism and *R*-tree to reuse the generated sanitized locations, which achieve the notion of geo-indistinguishability with less consumption of privacy budget. As the experiments show, with the same utility, the reuse of sanitized locations allows users to sanitize the same trace with less privacy budget.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. U1405255, 61672413, 61672408, 61502368, 61602537, 61602357, 61303221, U1509214), National High Technology Research and Development Program (863 Program) (Grant Nos. 2015AA016007, 2015AA017203), China Postdoctoral Science Foundation Funded Project (Grant No.2016M592762), Shaanxi Science & Technology Coordination & Innovation Project (Grant No.2016TZC-G-6-3), Shaanxi Provincial Natural Science Foundation (Grant Nos. 2015JQ6227, 2016JM6005), China 111 Project (Grant No. B16037), Beijing Municipal Social Science Foundation(Grant No. 16XCC023), Fundamental Research Funds for the Central Universities (Grant Nos. JB150308, JB150309, JB161501, JBG161511).

References

1. Xiao Y, Xiong L (2015) Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, p 1298–1309
2. Beresford AR, Stajano F (2003) Location privacy in pervasive computing. IEEE Pervasive Comput 2(1):46–55
3. Dwork C Differential privacy. In: Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, p 1–12, vol 2006
4. Andrés ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C (2013) Geo-indistinguishability: Differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, p 901–914
5. Ghinita G (2013) Privacy for location-based services. Synthesis Lectures on Information Security Privacy, & Trust 4(1):1–85
6. Guo M, Jin X, Pissinou N, Zanlongo S, Carbunar B, Iyengar SS (2015) In-network trajectory privacy preservation. ACM Comput Surv 48(2):23
7. Terrovitis M (2011) Privacy preservation in the dissemination of location data. ACM SIGKDD Explorations Newsletter 13(1):6–18
8. Gao S, Ma J, Shi W, Zhan G, Sun C (2013) Trpf: A trajectory privacy-preserving framework for participatory sensing. IEEE Trans Inf Forensics Secur 8(6):874–887
9. Niu B, Li Q, Zhu X, Cao G, Li H (2015) Enhancing privacy through caching in location-based services. In: 2015 IEEE Conference on Computer Communications (INFOCOM), p 1017–1025
10. Fawaz K, Feng H, Shin KG (2015) Anatomization and protection of mobile apps' location privacy threats. In: 24th USENIX Security Symposium, Washington, D.C., USA., p. 753–768
11. Ying B, Makrakis D, Hou Z (2015) Motivation for protecting selfish vehicles' location privacy in vehicular networks. IEEE Trans Veh Technol 64(12):5631–5641
12. Rios R, Cuèllar J, Lopez J (2015) Probabilistic receiver-location privacy protection in wireless sensor networks. Inf Sci 321:205–223
13. Ercument Cicek A, Nergiz ME, Saygin Y (2014) Ensuring location diversity in privacy-preserving spatio-temporal data publishing. The VLDB J 23(4):609–625
14. Ma T, Zhou J, Tang M, Tian Y, Al-Dhelaan A, Al-Rodhaan M, Lee S (2015) Social network and tag sources based augmenting collaborative recommender system. IEICE Trans 98-D(4):902–910
15. Gao S, Ma J, Shi W, Zhan G (2015) LTPPM: a location and trajectory privacy protection mechanism in participatory sensing. Wirel Commun Mob Comput 15(1):155–169
16. Gong X, Chen X, Xing K, Shin D-H, Zhang M, Zhang J (2015) Personalized location privacy in mobile networks: A social group utility approach. In: 2015 IEEE Conference on Computer Communications (INFOCOM), pp 1008–1016
17. Qi J, Wei F, Shuai F, Ma J, Li G, Alelaiwi A (2015) Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. Nonlinear Dyn 83(4):2085–2101
18. Qi J, Khan MK, Xiang L, Ma J, He D (2016) A privacy preserving three-factor authentication protocol for e-health clouds. The Journal of Supercomputing, p 1–24
19. Qi J, Ma J, Wei F (2016) On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services. IEEE Systems Journal
20. Zhangjie F, Fengxiao H, Xingming S, Vasilakos A, Yang C-N (2016) Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Transactions on Services Computing
21. Fu Z, Sun X, Ji S, Xie G (2016) Towards efficient content-aware search over encrypted outsourced data in cloud. In: IEEE Conference on Computer Communications, pp 1–9
22. Zhangjie F, Ren K, Shu J, Sun X, Huang F (2016) Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 27(9):2546–2559
23. Zhangjie F, Xinle W, Guan C, Sun X, Ren K (2016) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716
24. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352
25. Wang X, Yi M, Chen R (2016) One-round privacy-preserving meeting location determination for smartphone applications. IEEE Trans Inf Forensics Secur 11(8):1712–1721

26. Bilogrevic I, Jadliwala M, Joneja V, Kalkan K, Hubaux J-P, Aad I (2014) Privacy-preserving optimal meeting location determination on mobile devices. IEEE Trans Inf Forensics Secur 9(7):1141–1156
27. Puttaswamy KPN, Wang S, Steinbauer T, Agrawal D, Abbadi AE, Kruegel C, Zhao BY (2014) Preserving location privacy in geosocial applications. IEEE Trans Mob Comput 13(1):159–173
28. Shen Y, Huang L, Li L, Xiaorong L, Wang S, Yang W (2015) Towards preserving worker location privacy in spatial crowdsourcing. In: 2015 IEEE Global Communications Conference, San Diego, CA, USA, p 1–6
29. Yi X, Paulet R, Bertino E, Varadharajan V (2016) Practical approximate k nearest neighbor queries with location and query privacy. IEEE Trans Knowl Data Eng 28 (6):1546–1559
30. Schlegel R, Chow C-Y, Huang Q, Wong DS (2015) User-defined privacy grid system for continuous location-based services. IEEE Trans Mob Comput 14(10):2158–2172
31. Shao J, Lu R, Lin X (2014) Fine: A fine-grained privacy-preserving location-based service framework for mobile devices. In: IEEE Conference on Computer Communications, p 244–252
32. To H, Ghinita G, Shahabi C (2014) A framework for protecting worker location privacy in spatial crowdsourcing. Proceedings of the VLDB Endowment 7(10):919–930
33. Chatzikokolakis K, Palamidessi C, Stronati M (2014) A predictive differentially-private mechanism for mobility traces. International Symposium on Privacy Enhancing Technologies Symposium, pp 21–41
34. Xi H, Cormode G, Machanavajjhala A, Procopiuc CM, Srivastava D (2015) DPT: differentially private trajectory synthesis using hierarchical reference systems. PVLDB 8(11):1154–1165
35. Reed J, Pierce BC (2010) Distance makes the types grow stronger: a calculus for differential privacy. In: ACM Sigplan Notices, volume 45, p 157–168
36. Roth A, Roughgarden T (2010) Interactive privacy via the median mechanism. In: Proceedings of the forty-second ACM symposium on Theory of computing, pp 765–774. ACM
37. Zheng Y, Xie X, Geolife W-YM (2010) A collaborative social networking service among user, location and trajectory. IEEE Data Eng Bull 33(2):32–39

Xindi Ma received the B.S. degree in school of computer science and technology from Xidian University, China in 2013. He is currently working toward the Ph.D. degree at the School of Cyber Engineering, Xidian University, China. His current research interests include database security, location-based service and recommender system with focus on security and privacy issues.

Jianfeng Ma received the B.S. degree in computer science from Shaanxi Normal University in 1982, and M. S. degree in computer science from Xidian University in 1992, and the Ph. D. degree in computer science from Xidian University in 1995. Currently he is a Professor at School of Computer Science and Technology, Xidian University. His research interests include information security, cryptography, and network security.

Hui Li received the B.Eng from Harbin Institute of Technology in 2005 and Ph.D. degree from Nanyang Technological University, Singapore in 2012, respectively. He is an Associate Professor in School of Cyber Engineering, Xidian University, China. His research interests include data mining, knowledge management and discovery, privacyp-reserving query and analysis in big data.

Qi Jiang received the B.S. degree in Computer Science from Shaanxi Normal University in 2005 and Ph.D. degree in Computer Science from Xidian University in 2011. He is now an associate professor at School of Cyber Engineering, Xidian University. His research interests include security protocols and wireless network security, cloud seurity, etc.

Sheng Gao is an Assistant Professor in the School of Information at Central University of Finance and Economics. He received the B.S. degree in information and computation science from Xi'an University of Posts and Telecommunications, in 2009, and the Ph.D. degree in computer science and technology from Xidian University, in 2014. His current research interests include finance information security and privacy computing.