2013 IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 2013 IEEE 10th International Conference on Autonomic & Trusted Computing

Privacy Preserving Path Recommendation for Moving User on Location Based Service

Yuqing Sun
Shandong University
sun_yuqing@sdu.edu.cn

Haoran Xu
Shandong University
hr_xu1990@163.com

Reynold Cheng
University of Hong Kong
ckcheng@cs.hku.hk

***Abstract*—With the increasing adoption of location based services, privacy is becoming a major concern. To hide the identity and location of a request on location based service, most methods consider a set of users in a reasonable region so as to confuse their requests. When there are not enough users, the cloaking region needs expanding to a larger area or the response needs delay. Either way degrades the quality-of-service. In this paper, we tackle the privacy problem in a predication way by recommending a privacy-preserving path for a requester.We consider the popular navigation application, where users may continuously query different location based servers during their movements. Based on a set of metrics on privacy, distance and the quality of services that a LBS requester often desires, a $secure$ path is computed for each request according to user's preference, and can be dynamically adjusted when the situation is changed. A set of experiments are performed to verify our method and the relationship between parameters are discussed in details. We also discuss how to apply our method into practical applications.**

Keywords*—*Location privacy, navigation, predication

I. Introduction

With the continued advances in mobile networks and positioning technologies, location based services (LBS) are widely adopted in daily life[8], [11]. For example, drive navigation is a very popular application [9]. In this process, a user may continuously report his current location to a service provider and query real-time road or traffic information so as to find a fast path to the destination [4]. In these applications, privacy is an important issue. An adversary may obtain unauthorized access to raw location data on LBS servers and identify the subject using the positioning device. Having knowledge of users' location related information, malicious parties can reason out more information about users' habits, and health conditions etc. for further business initiatives or political purposes.

Previous protection on location privacy is processed by LBS providers and is enforced according to user defined privacy policies [5]. Here LBS users have to trust a LBS server that their locations and identity information can be adequately protected. However, in some cases, a LBS provider may use users' information for improper purpose or leak users' query messages to other parties for some commercial reason.

To solve the problem of untrusted LBS servers, a trusted third-party, anonymity server, is introduced between LBS servers and users, as well as the channel between users and the anonymity server is assumed secure[7]. In this scenario, when a LBS query occurs, the trusted anonymity server performs some preprocessing, such as confusing the real user identity with anonym or hiding his/her exact position with a cloaking square, so as to blur the link between a query and the user's identity. Then it transmits the query to a LBS server. When the response is returned from the LBS server, the anonymity server transfers it to the intended users. In this process, LBS users can not be identified even though the location and identity information in queries are acquired by adversaries.

Based on this architecture, the $k-anonymity$ is regarded as a typical criterion to evaluate the extent of privacy protection [16], [17]. When a user requests from a LBS server which may be not credible, the trusted anonymity server computes a cloaking region that contains the user and at least $k-1$ other neighbors. The sender's accurate position is then replaced with a coarser spatial range and the sender is indistinguishable with them such that the adversary will have uncertainty in matching each exact user to a known location-identity association. Another representative method under this architecture is $mix-zone$, which is a specific region where users' accurate identities are mixed and hidden from adversaries' deduction [15]. When users need to protect their location based privacy, they enter mix-zones. This solution also requires a number of LBS users in a mix-zone.

However, the assumption in the above architecture that enough users locate in the cloaking region can not be satisfied all the time in real life. When there are less users in a considered area, location privacy of LBS users may be compromised. The current remediation extends the $k-anonymity$ criterion from both spatial and temporal aspects [7]. The spatial cloaking takes a larger geographical range so that enough users are included, while the temporal cloaking delays the response to a query for a period of time so as to wait for enough number of users. Either way degrades the quality of LBS service (QoS). Especially in the scenario of path navigation, users may continuously send LBS queries or send different queries on any point of the path. The low quality of location based services would deviate users' intentions.

In this paper, we tackle the privacy problem in the popular navigation applications in a different way. Based on a series of metrics on privacy, distance, and the quality of services that a LBS requester often desires, we introduce the notion of secure path. The proposed algorithm recommends a path for each request based on the current situation and user preferences such that on each step of the path user's privacy is preserved under the $k-anonymity$ criterion. When the situation on next step is changed insecure during user movement, the recommended secure path is dynamically adjusted. The core of our method is to provide a LBS user more privacy without degradation of QoS if he/she moves along the recommended path rather than

978-1-4799-2481-3/13 $31.00 © 2013 IEEE
DOI 10.1109/UIC-ATC.2013.16

IEEE computer society

a random movement. Although a preliminary version of this idea was presented in [10], this paper makes a comprehensive extension in four aspects: the integration of secure and insecure situation, formal verification, a series of evaluation metric and thorough experiments, and a $k-free$ discussion on how to apply our method into practical applications.

Other related works include privacy proteciton in publishing historical trajectories, which sanitize the original data and group similar trajectories for publishing[2]. For example, the local suppression method achieves a tailored privacy for trajectory data anonymization[3]. The purpose of these studies is to anonymize a specific user trajectory with other users in the same group while preserve an extent of utility. It does not solve the privacy problem in real time queries. Although the algorithm in [18] concerns the privacy in route predication, it focus on predicating a user destination by analyzing historical trajectories and notifying a user whether the next position is secure for LBS request. It does not recommend a user how to choose a privacy preserving path for continuous LBS requests. Our work also seems related to path planning. However, path planning algorithms traditionally focus on desirable paths, which are usually measured by travel distance or time[12]. They do not integrate privacy concern.

The rest of this paper is organized as follows. In Section II, we give the framework of path planning based privacy protection and the path predication algorithm. Then we present the evaluation metrics in Section III, as well as the experimental study. We then discuss how to balance the preferences on privacy and distance in section IV. Conclusions and future works are presented in section V.

II. The Privacy Preserving Model

We adopt the widely used trusted third-party architecture for preserving users' location privacy. Our work resides on the anonymity server (A-Server for short) and focuses on the popular navigation applications, which is adaptive to other related LBS with the characteristics of movement and continuous queries as well. When there is a navigation request along with a start point and a destination, the anonymity server invokes the path planning algorithm to find a privacy-preserving path for the requester. If the state of user's next step becomes insecure during movement, A-Server would find another secure path for the rest distance to his destination. In this process, a user is allowed to specify his personalized preferences on privacy and distance, such as an integer k as the $k-anonymity$ criterion or a detour rate on how far is acceptable for him to detour comparing to the geographically nearest path. In the following sections, we firstly narrate our method under the $k-anonymity$ criterion and then discuss how the k setting can be ignored by real users in practice.

A. Basic Terminology

The considered geographical region is described as a mesh or grid, denoted as $Mesh$. Each cell of the mesh is regarded as a cloak region when a user requests a location-based service. Each cell i has a pair of coordinates $c_i(x, y)$ denoting the top-left point of this cell. The $resolution$ of a mesh is the number of cells on it, denoted as $|Mesh|$. This manner of defining geographical region is often used in a geographic information system (GIS) to capture, store, manipulate, analyze, manage, and present different types of geographically referenced data [6]. It is easily reducible and enlargeable in practice. Positions from a global navigation satellite system like Global Positioning System (GPS) can be collected and then imported into a GIS. The coordinates of each cell can be mapped to the latitude and longitude in geography science. The status of a cell i refers to the number of LBS users in this region, denoted as η_i. The mesh situation refers to the distribution of the whole set of LBS users in the considered area. If there is a lake or a mountain in a map, the corresponding cells are marked as $obstacles$. A path on $Mesh$ is a sequence of cells from a start position to a destination. Formally, $path = \{c_1, c_2, \cdots, c_l\}$, where l is an integer denoting the length of the path ($\mathcal{L}(path)$), $c_i \in Mesh, i \in [i..l]$, is a cell and $c_i(x_i, y_i)$ and $c_{i+1}(x_{i+1}, y_{i+1})$ satisfy either $|x_{i+1} - x_i| = 1$ or $|y_{i+1} - y_i| = 1$.

According to the semantics of $k-anonymity$ criterion, users are regarded secure if there are at least k users in the cloak region. The purpose of an attacker is to identify a LBS requester from an observed set of LBS users in a cloak region. In essence, the privacy of each user in a cloak region is computed as the uncertainty of an adversary in linking him to a requester. Since all users look identical within the anonymous user set, the probability of successful linking is $1/\eta_i$, where η_i is the number of users in the cloak region. Since entropy is widely adopted to quantify the uncertainty[13], we use entropy to evaluate user privacy in a cloaking region.

$$H(c_i) = -\sum_{i=1}^{\eta_i}(1/\eta_i * log_2(1/\eta_i)) = log_2(\eta_i) \tag{1}$$

Intuitively, the more users in a cell, the higher entropy of the cell and user privacy. Different with a snapshot LBS query, the privacy of continuous LBS queries along with user movement requires the whole path is secure under $k-anonymity$. So, we introduce the concepts of cell privacy and path privacy.

Definition 1: [*cell k-anonymity privacy*] Given a mesh $Mesh$ and an integer k for the $k-anonymity$ criterion, the $k-anonymity$ privacy of a cell $c_i \in Mesh$ is defined as:

$$\mathcal{H}^k(c_i) = \begin{cases} log_2(k), & \eta_i \geq k \\ log_2(\eta_i), & 1 < \eta_i < k \\ 0, & \eta_i \leq 1 \end{cases} \tag{2}$$

where η_i is the number of LBS users in c_i.

The semantics of the above definition is that when the number of users in a cell equals or is larger than k, the privacy of the users in the cell remains the same, denoted as H_K. This shows the fact that under the $k-anonymity$ criterion, the users are regarded privacy preserving in such cases, and the cell is called $secure$ as well. Actually, this criteria makes a balance between privacy and quality of LBS service. Although a higher k guarantees better privacy, it generally requires a larger cloaking region to cover k LBS users, which may reduce request resolution and result in coarse results.

Definition 2: [*user request*] Given a mesh $Mesh$, a user navigation request is in the form of 4-tuple $< u_{id}, St, Des, k >$, where u_{id} is the identifier of the requester, $St \in Mesh$, and $Des \in Mesh$ respectively represent the user's current position and destination, k is an integer denoting the user's privacy preference for $k-anonymity$ criterion.

Our purpose is to find a $k-anonymity$ secure path $pt = \{c_1, c_2, \cdots, c_l\}$ for a user request such that each cell $c_i, i \in [1..l]$ is $k-anonymity$ secure. The path privacy is then defined as the overall evaluation of cell privacy on the path.

B. The Proposed Algorithm

In this section, we present the personalized privacy-preserving path planning algorithm. A LBS user is allowed to specify an integer k as his privacy preference with respect to $k - anonymity$ criterion. The path planning is a dynamic process along with the periodical update of mesh situation in real time. When solving the personalized path planning problem, we find there are quiet a few considerations similar to the $robot\ movement$ problem in a partially known environment (***RMP*** for short) [14].The partially known environment in ***RMP*** is similar to the dynamic update of mesh situation and the target of finding a sequence of points from some initial point to destination in ***RMP*** is also similar to the path planning in our problem. Since the $robot\ movement$ problem has been well studied and there are many efficient algorithms on solving it, we reduce our problem to it rather than to design a specific algorithm so as to benefit from the existing results.

The representative D^* algorithm for solving ***RMP*** consists of two stages: global planning and local planning [14]. Given a partially known map, it firstly invokes global planning according to the present map situation, namely finding a sequence of cells that a user can go through. Since the current map is partially known, the path calculated initially may not be exact and the local planning makes some necessary path adjustment. When the user moves forward along the path, the next state on the path is available. When encountering with some obstacle, the path would be re-planned according to the updated map state. Accordingly, our solution consists of the initial phase and the adjustment phase. For a LBS user request $< u_{id}, St, Des, k >$, it finds a secure path from St to Des such that each cell on the path is secure with respect to k and the total cost of the path is optimal.

However, there also exist some distinctions between our problem and the RMP problem. The most important is that we need to consider privacy besides distance. Another difference is the dynamic environment. Obstacles on a map are fixed about path planning, while in our problem cell state may change any time during user movement, i.e. from $insecure$ to $secure$ or in a reverse way. Finally, the definition of a path length is different. So, we need to make some tricks on the reduction so as to satisfy both ***RMP*** requirements and our considerations. The details of reduction are given below.

Cell Cost and Path Cost: each cell is assigned a cost according to mesh situation and is used in calculating a path in RMP. Since the core on choosing a path in our problem is the guarantee of path privacy, we select each secure cell on the path and discard insecure cells. To map to the arc cost requirements in RMP, we set a small positive cost to $k - anonymity$ secure cells and a positive infinity cost (i.e. a large enough integer) to insecure cells. An example of such setting can be as $cost(c_i) = 1$ on each secure cell c_i and $cost(c_i) = 1000$ on each insecure cell. Path Cost is the sum of cell cost on path. It is calculated by the predicate $Cost_{path}(p_i, p_j) = \sum_{x \in path(p_i,p_j)} cost(x)$, denoting the estimated cost of the path from cell p_i to p_j.

Cell Mark and Path Calculation. The considered geographic region is described as a mesh mapping to the map in RMP. Each cell on the mesh is associated with a mark belonging to the set $\{New, Closed, Open\}$ denoting whether the cell has been visited, still active or closed. If a cell is never visited, its state is set New. A visited secure cell marked with $open$ denotes that it is active and is regarded as a candidate for a secure path, while a visited but insecure cell or an obstacle cell (like a lake or a region not being allowed to enter) is marked with $Closed$ so as to prevent a loop detour. We adopt a linked list to record all the cells on a recommended path for each request $< u_{id}, St, Des, k >$. The initial estimated cost of a path $estimated_{path}(St, Des)$ is calculated according to the current mesh situation. Since the mesh situation may change any time, the cost of path via some point p_i is updated as $Cost_{path}(St, p_i, Des) = Cost_{path}(St, p_i) + estimated_{path}(p_i, Des)$, where $Cost_{path}(St, p_i)$ is the real distance between St and p_i that a user has moved through. After invoking the D^* Algorithm, we get a secure path for recommendation.

In practice, the algorithm records a table of recommended paths for recent requests and updates periodically along with mesh situation or new path generation. When a user request occurs, the algorithm checks whether there is a similar LBS request and returns the recent previously planned path. Otherwise, the algorithm would plan a new path for him. As a user moving along his recommended path, cell states may change insecure. Then the phase of dynamic path adjustment is invoked. A new secure path from the current cell to his destination is recalculated based on the current mesh situation. When the target $Destination$ is arrived, the algorithm finishes.

C. Analyses on the Proposed Method

In this subsection, we would prove the correctness of our method. To reflect user desire on privacy and short distance, we first introduce the concept of dominate relation on path.

Definition 3: [*Dominate Relation*] Given a mesh $Mesh$, a user request $req =< u_{id}, St, Des, k >$ and two paths $path_1 = \{c_1, c_2, \cdots, c_{l_1}\}$, $path_2 = \{c'_1, c'_2, \cdots, c'_{l_2}\}$ on $Mesh$ for req, we say that $path_1$ Dom $path_2$ if and only if one of the following cases holds :

- case 1: $path_1$ is $kA-secure$ while $path_2$ is insecure.
- case 2: $path_1$ and $path_2$ are $kA - secure$, $l_1 \leq l_2$.

The above definition actually considers what a user mainly desires in a navigation request. The first case compares a secure path to an insecure path while the second case compares two secure paths. We would prove that our method could find the optimal path (if exist).

Lemma 1: For a given mesh $Mesh$ and a user request $req =< u_{id}, St, Des, k >$, the proposed algorithm returns a $kA-secure$ path if there exists, and this path is not dominated by any other path connecting St and Des.

Proof: We prove this lemma from two aspects. First, assume there exist a $kA - secure$ path for req under $Mesh$. We would prove that the above solution would not return a path including insecure cells. According to the construction of $cost$ function, each secure cell is set a small integer (in our method it equals to 1) and each insecure cell is set a very larger integer (e.g.∞). Without loss of generality, we assume the length of the secure path is l, namely the path cost is $cost_{path}(st, des) = l$. Since there is no loop on path, the path length would not exceed the resolution of $Mesh$ (the total number of cells in $Mesh$). However, the cost of a path covering an insecure cell is definitely larger than the resolution and thus is larger than the cost of the secure path. Since the $D*$ algorithm in solving ***RMP*** chooses the shortest path, it would not return a path including an insecure cell.

Then we prove that if there exist more than one $kA - secure$ paths, our method returns the shortest one. According to the definition of path cost and the construction of $cost$ function, the cost of any $kA - secure$ path equals to its length. So, the cost of the short secure path is less than that of the longer secure path. The $D*$ algorithm returns the shortest $kA - secure$ path. ■

D. Handling Exception

In case there does not exist a $k - anonymity$ secure path for a user's request, say a very large k evaluates every cell insecure, the algorithm will notify the user for a smaller k or recommend a constructive path with less privacy risk. Accordingly, the concept of cell risk is introduced as a part of cell cost so as to integrate the secure state and the insecure state together in computing path.

Cell Risk. Given a mesh $Mesh$ and an integer k, cell risk is defined as the privacy difference between the insecure cell against a $k - anonymity$ secure cell, namely $Risk(c_i) = H_K - \mathcal{H}^k(\eta_i)$, where $H_K = log_2 k$ and η_i is the number LBS users in cell c_i. Cell risk describes how much information an attacker learns from the decrease of k to η_i. For normalization purpose, we introduce the risk degree of cell i:

$$\gamma_i = \frac{Risk(c_i)}{H_K} = 1 - \frac{\mathcal{H}^k(c_i)}{H_K} \tag{3}$$

Since $k > 1$, γ_i always lies within [0..1] and $\gamma_i = 0$ holds if c_i is $k - anonymity$ secure cell. The case $\gamma_i = 1$ indicates only one user in cell i and he/she can be recognized with probability 1. To overall consider distance and cell state, cell cost is extended as:

$$Cost(c_i) = \begin{cases} 1 & : \eta_i \geq k \\ \mathcal{B} + \gamma_i * RMax & : otherwise \end{cases} \tag{4}$$

where $\mathcal{B} \geq 1$ and $RMax \geq 0$ are parameters denoting how cell risk influences the path planning.

The first part $\mathcal{B}$ means a basic assignment to an insecure cell. Since cell cost is used as the distance in the path planning algorithm, insecure cell cost should be larger than secure cell cost, i.e. $\mathcal{B} \geq 1$. The second part illustrates that the cost scales with the risk of cell insecure state. The setting of these parameters would influence the final generated path. Intuitively, increasing $\mathcal{B}$ indicates a user's preference on more privacy rather than detour on path. For example, when setting $\mathcal{B} = \infty$, only secure cells are chosen as candidates on the path. In case there is not a $kA - secure$ path for a request and a path recommendation is still desired, a less risky path can be found by setting an appropriate $RMax$. A larger $RMax$ amplifies the differences between risky cells. When setting $RMax = 0$, all insecure cells are regarded same.

In practice, it is difficult for a user to choose an appropriate k for his privacy preference. Instead, he may express his preference as either a higher secure rate or overall path privacy. The algorithm can satisfies users' preferences by accommodating these parameters. We would discuss what parameter settings are appropriate for a user's preference under current mesh situation in following sections. .

III. Evaluation Metric and Experiments

A. Evaluation Metric

Success rate (*Suc_Rt*): Given a mesh $Mesh$, an integer k for the $k-anonymity$ criterion and a $path = \{c_1, c_2, \cdots, c_l\}$, where l is a positive integer, the success rate is the ratio of the number of secure cells on $path$ (denoted as $\mathcal{N}_{sec}(path)$) to the path length. Formally,

$$Suc\_Rt_{path} = \mathcal{N}_{sec}(path)/l \tag{5}$$

It describes how well the $k - anonymity\ criteria$ is satisfied on this path. We also adopt the $\mathcal{N}_{insecure}(path)$ to denote the number of insecure cells on the path.

k-anonymity path privacy (kA-Privacy). Given a mesh $Mesh$, a path $path = \{c_1, c_2, \cdots, c_l\}$ on $Mesh$ and an integer k for the $k - anonymity$ criterion, the $kA - Privacy$ of the path is the average of cell k-anonymity privacy. Formally,

$$\mathcal{H}^k(path) = \frac{1}{l} * \sum_{i=1}^{l} \mathcal{H}^k(c_i) \tag{6}$$

Comparing to the metric of success rate, $kA - Privacy$ is a more fine-grained evaluation of path privacy. Especially when there does not exist a $k - anonymity$ secure path, a user may choose a path with higher privacy.

In the process of finding a privacy-preserving path, a detour may often be encountered, which compromises the navigation purpose on shortest way. So, a tradeoff sometimes must be made between the distance and privacy. We introduce the following metric of detour rate to express the tolerance a user would like to accept on path detour.

Detour rate(*$Detour\_Rate$*). Given a mesh $Mesh$, two cells $c_1 \in Mesh$ and $c_l \in Mesh$, and a $path = \{c_1, c_2, \cdots, c_l\}$ on $Mesh$, the Detour Rate is defined as:

$$Detour\_Rate = \frac{l - \mathcal{L}_{spath}(c_1, c_l)}{\mathcal{L}_{spath}(c_1, c_l)} \tag{7}$$

where $\mathcal{L}_{spath}(c_1, c_l)$ denotes the shortest path length connecting c_1 and c_l.

Accuracy. The metric of accuracy evaluates the precision of a location based service, which is determined by the size of a cloaking region (CR). The smaller a CR, the higher quality the LBS associated with CR.

In our solution, CR size is determined by the resolution of a given map. However, from the competitive point of privacy, a higher resolution would cause fewer users in a cell, which would result in a lower probability of $k-anonymity$ satisfied. So, in the following sections, we would discuss how to make a trade-off between these metrics.

B. *Evaluation Benchmark*

A benchmark is a set of standards in order to assess our solution against the metrics. The purposes of introducing benchmark are, on one side, to evaluate how well our solution can affect the extent of privacy protection; on the other side, to understand what a LBS user desires has been sacrificed during this process. So, the selection of benchmark should consider both positive and negative aspects. In a common sense, when a user makes a navigation request, the response from a LBS server finds the shortest path for him. So, the selection of distance benchmark should randomly choose a shortest path on a map without considering privacy. Since Manhattan path is widely adopted in geographical applications to evaluate the distance between two intersections in a borough, we adopt it as the distance benchmark. For a given start point c_1 and a destination point c_2, $Manhattan\ distance$ is defined as the length of a Manhattan path connecting these two points, denoted as $\mathcal{L}_{Mpath}(p_1, p_2)$.

This Manhattan distance is actually the absolute differences of the coordinates of the start point and the destination. Since a Manhattan path resides on the rectangle determined by there two cells, we call this rectangle the *minimum shortest-path closure (*MSC *for short)*. The shortest path length in equation 7 is replaced by the Manhattan distance, namely

$$Detour\_Rate = \frac{l - \mathcal{L}_{Mpath}(p_1, p_2)}{\mathcal{L}_{Mpath}(p_1, p_2)} \tag{8}$$

Since a shortest path may randomly cover the cells in MSC, the privacy attributes of MSC should be chosen as the benchmark also.It includes the average success rate and the average $k - anonymity$ privacy, namely

$$BenSuccess = \frac{\mathcal{N}_{sec}(MSC)}{|MSC|} \tag{9}$$

$$BenPrivacy = \frac{1}{|MSC|} * \sum_{c_i \in MSC} \mathcal{H}^k(c_i) \tag{10}$$

where $\mathcal{N}_{sec}(MSC)$ is the number of secure cells in the MSC region and $|MSC|$ is the size of MSC.

We would like to mention again that the above selected benchmarks are based on randomly selected paths connecting a start and a destination. Based on the above metrics and benchmarks, we would present and evaluate the proposed location privacy protection solution in next section.

C. *Experimental Study*

The experiments were conducted on a desktop with 3.10GHz CPU, 3.16G memory and 500GB disk space installed the operating system Windows 7. All experimental results are the average values of more than ten times of program running. LBS users are randomly distributed on a map.

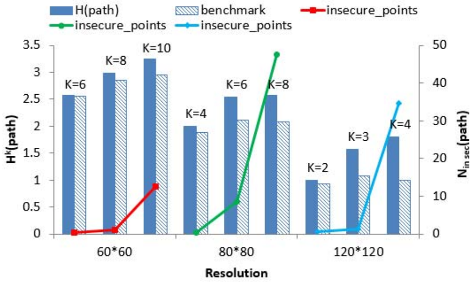
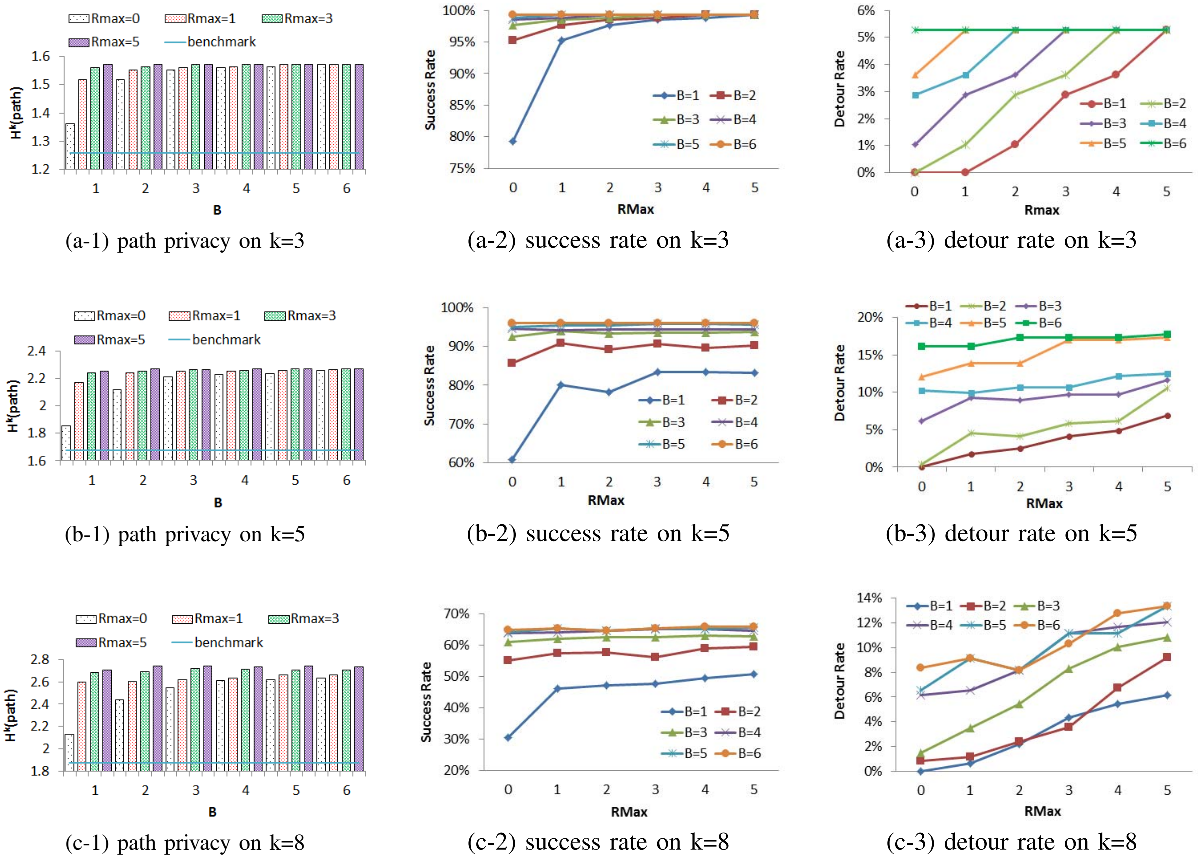


Fig. 1: The relationship between privacy and QoS

We first quantitatively investigate the relationship between the quality of LBS and privacy, as shown in Figure 1. This set of experiments are performed on the same map with a fixed number of LBS users so as to simulate a practical scenario. Intuitively, a higher resolution means a smaller cell size and results in better QoS and less users in a cell. For example, if the number of users is 10000 and the map resolution is 60*60, the average number of users on each cell is about 2.8. However, if the resolution is amplified to 120*120, this number becomes less than 1. In Figure 1, The $\mathbb{X}$ axis is the resolution of the mesh. The left $\mathbb{Y}$ axis is the privacy (in the form of entropy), while the right $\mathbb{Y}$ axis is the number of insecure cells on a path. The solid histograms denote the average privacy of generated paths and the histograms with slant lines denote the privacy benchmarks, as defined in equation 10. The broken lines are the number of insecure cells in a generated path. On each setting, we perform twenty times and get the average result. Each invocation denotes a user request under the same k but with different locations. The results show that with the increase of map resolution, the average privacy of a path decreases, while the number of insecure cells increases. We did not compare the same k under different resolutions since obviously a larger resolution results fewer users in a cell and only small k can be satisfied. This illustrates that a tradeoff must be made between the competitive targets of privacy and quality of LBS. Consider the situation of a fixed resolution, a higher k results in more insecure cells of a path under the $k - anonymity$ criterion, namely a lower success rate, but the average privacy (in entropy) of this path may increase.

Then, we evaluate how the risk parameters $\mathcal{B}$ and $RMax$ influence the attributes of a selected path. The experiments are performed on the same map with the mesh size 60*60. A set of dummy LBS users are randomly distributed on the map, who take part in the $k - anonymity$ evaluation but without any navigation request. This assumption has been well discussed in previous work [1]. The number of dummy users in each cell ranges from 0 to 9 with the same probability. When k is fixed, cell states are determined. For example, all the cells associated with less than 3 users are evaluated insecure when $k = 4$. The higher k, the more insecure cells. We analyze the relationship between the privacy and k setting. Three representative cases are selected: a lower risk situation $k = 2$, a middle risk situation $k = 5$ and a higher risk situation $k = 8$. In each case, we consider three aspects of planned paths: secure rate, cell $kA - privacy$ and detour rate, as defined in equations 2, 5 and 7 respectively. The benchmark of each attribute is calculated against the equations 8, 9 and 10, respectively. The selected $\mathcal{B}$ ranges from 1 to 20 and $RMax$ ranges from 0 to 100. The results show that after $\mathcal{B} \geq 6$ and $RMax \geq 5$, the metrics on privacy and distance remain almost the same.

The experiment results in Figure 2 show that the average path privacy and secure rate scale well with the increase of $\mathcal{B}$



(a-1) path privacy on k=3

(a-2) success rate on k=3

(a-3) detour rate on k=3

(b-1) path privacy on k=5

(b-2) success rate on k=5

(b-3) detour rate on k=5

(c-1) path privacy on k=8

(c-2) success rate on k=8

(c-3) detour rate on k=8

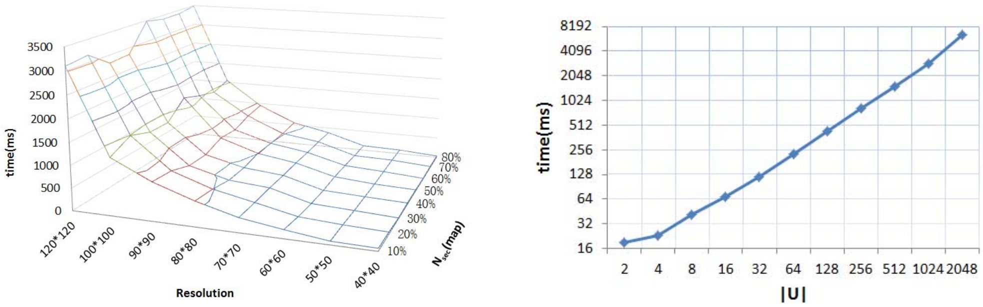
Fig. 2: Relationship between parameters

and $RMax$. They approach a stable value after the parameters exceed some thresholds, and are better than the benchmark in all cases. With the increasing k, a user has to detour more for a higher guarantee on path privacy. Comparing different mesh situations, the stable values vary with each other. In smaller k cases, shown as (a-1), (b-1) and (c-1), the stable point of privacy is smaller while the success rate is higher than in larger k cases. This is because the planning algorithm has more opportunities to find a secure path under a low k setting; while a higher k setting causes more cells insecure. Also the average path privacy scales positively with k increasing because the planning algorithm tries to find a secure or less risky cell on each step (namely more users). It is easy to understand that, in case k is fixed and mesh situation is determined, the average of path privacy is definitely increasing with the number of dummy users.

The above analysis shows that under the same mesh situation, a user's preference can be satisfied by setting different parameters. If a user prefers a lager secure rate of a path and would like to accept more detour, $\mathcal{B}$ should be set higher. When a user prefers high average privacy of a path and does not want to detour more, we should set a smaller $\mathcal{B}$ and a larger $RMax$. The differences between insecure cells, i.e different η_i, would influence path chosen. The reason of not assigning the infinity to a high risky cell is that sometimes a user has to go through an insecure cell when there is no other choice.

Next, we study the efficiency of our method. In path predication, the number of cells in a mesh influences the average length of paths and cell states determine how many tries on each step chosen. So, we evaluate how the Mesh situation influences the algorithm efficiency in two folds: the size of mesh and the percentage of secure cells for a fixed k in a mesh, denoted as $\mathcal{R}_{sec}(map)$. The settings in this experiment are $RMax = 5$, $\mathcal{B} = 6$, $k = 6$, and the simulated number of dummy LBS users on each cell ranges from 0 to 9. The generated users' requests are randomly distributed on the map. As results in Figure 3 (a), the running time scales positive with the resolution since the algorithm has to plan more steps for a user request under a high resolution mesh; while it is slightly decreases with $\mathcal{R}_{sec}(mesh)$ increase. Figure 3 (b) shows how the efficiency scales with the number of total user requests, denoted as $|U|$. The resolution in this experiment is $60 * 60$. The results show that the response time scales linearly with the increase of $|U|$ and the overall time is acceptable.

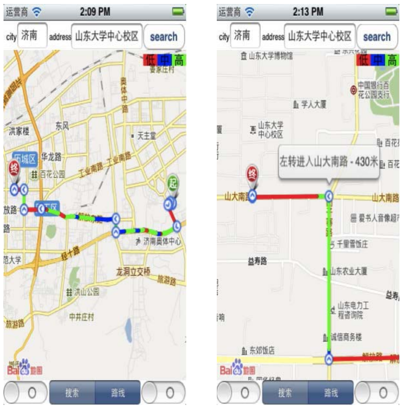
Finally, we prototype a system on Mac OS X Lion with Xcode. We adopt the Objective-C programming language to develop the system and use the application programming interface (API) on Baidu website to get map, traffic and points of interest etc. information. Figure 4 is the snapshots of



(a) running time vs. mesh situation.

(b) running time vs. user requests

Fig. 3: Efficiency of the personalized path planning algorithm



(a) example1

(b) example2

Fig. 4: Snapshot of prototype system

the prototype system with two examples of planned path for different requests. The green color denotes the high private part while red color indicates less private. Different with the simulated map, all paths should be along with the exist roads, which makes less choice. That is why some part of the path is in red color.

IV. Discussion

In the proposed predication based privacy protection solution, a user is allowed to specify k as his privacy preference for path planning. However, in practice a user himself may not have clear idea on what k is appropriate. If k is set too high, the success rate of a path is low. On the other side, although a small k could cause a high path success rate, the overall privacy may decrease. So, in this section we would discuss what is a practical choice for high guarantee on privacy.

A. What is an appropriate k setting

From the experimental results in the above section, we can see that for a given distribution of LBS users in a mesh, setting k as the average η_i seems a **golden mean** for general privacy requirements. The conclusion wholly consider multiple aspects of path privacy, secure rate and detour rate etc. For example, if η_i ranges within [0..9], $k = 5$ makes about half cells on a mesh secure and the metrics reach a good level. In practical applications, the anonymity server has the information about all LBS users' current locations. After mapping them to a mesh, the overall distribution of LBS users can be calculated, namely the number η_i of users on each cell c_i. Then k can be set the central core value of all η_i in path planning, namely the number of cells with $\eta_i > k$ approximately equals to the number of cells with $\eta_i \leq k$.

B. $k - free\ entropy$

The initial purpose of $k-anonymity$ criterion is to ensure the uncertainty of an adversary in linking a certain identity to a requester. It considers a cloaking region covering not less than k users such that the probability of successful linking is less than $1/\eta_i$, where η_i is the number of users in cell c_i. The above discussion is based on this semantics. However, when there does not exist a $kA - secure$ path for a user request, we need to revisit the semantics of $k - anonymity$ criterion and recommend an appropriate path under current mesh situation. As an alternative way, we propose the $k-free\ entropy$ criteria to select a comparatively high private path, which does not consider any specific k value. The $cost$ function is constructed as $cost(c_i) = RMax * (log_2 MaxN - log_2 \eta_i)$, where $MaxN$ is an integer larger than every η_i in the mesh and $RMax \geq 1$. Then the metrics of a generated path are the average of path privacy $\mathcal{E}(path)$ and the standard deviation of path privacy σ:

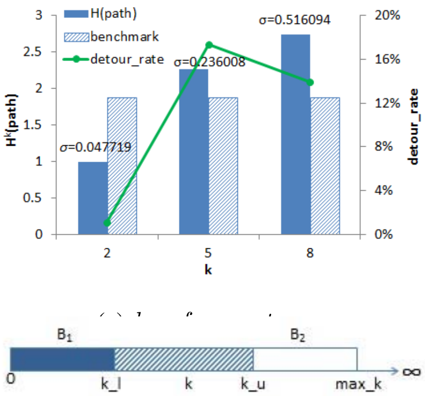
$$\mathcal{E}(path) = \frac{\sum_{c_i \in path} log_2 \eta_i}{\mathcal{L}_{path}}$$

$$\sigma = \sqrt{Exp(log_2 \eta_i - \mathcal{E}(path))^2}$$

We study the effectiveness of this method and make comparison on different k setting, shown in Figure 5 (a). From the results we can see that the $k - free\ entropy$ criteria has a better privacy without much detour rate. Comparing the cases with similar path privacy, it has a better σ.

C. Make a trade-off between privacy and communication jam

Although satisfying a higher k means more privacy, it may bring communication jam or heavy traffic. So we propose the $Moderate - Privacy$ criteria.The descriptive setting is shown in Figure 5(b). Considering a user's preferred k or a recommended $Golden\_k$ by our system, we choose a window



(b) A Moderate k setting

Fig. 5: Discussion

$[k_l, k_u]$ such that $k_l < k < k_u$, where k_l and k_u are integers, and the $cost$ function is constructed as following:

$$Cost(c_i) = \begin{cases} 1 & : \eta_i \in [k_l, k_u] \\ \mathcal{B}_1 + \gamma_i * RMax & : \eta_i \leq k_l \\ \mathcal{B}_2 & : \eta_i \geq k_u \end{cases} \tag{11}$$

$RMax$, $\mathcal{B}_1$ and $\mathcal{B}_2$ are positive numbers that satisfy $\mathcal{B}_1 \geq \mathcal{B}_2 \geq 1$. The semantics of such setting is that a moderate number of users in the anonymous set is enough. On one side, it can satisfy a user's privacy preference. On the other side, it can avoid LBS communication jam or heavy traffic.

V. Conclusion and Future Work

Privacy protection is a critical problem in location based services. Unlike previous works, we investigate this problem from the predication aspect rather than a user already in some position. This work is based on the widely adopted trusted third-party architecture. We investigate the metrics that a user often desires in continuous LBS queries, which takes an overview of path privacy rather than a snapshot LBS request at one position in previous metric. In the proposed privacy protection model, a user is allowed to specify a value k and detour rate as preferences on privacy and distance. The personalized privacy algorithm predicates an optimal path for a navigation request considering multiple considerations. Our model also integrates secure and insecure situations. We perform thorough experiments to evaluate our proposed method and the results show that, without degrading the Quality of Service, our method provides higher privacy than users' random movement. We also discuss how to apply our method to practical applications in which users generally do not know how to choose k and what a k setting exactly means, and show that our method is suitable for not only the $k-anonymity$ criterion but also a $k-free$ context.

In the future, we would integrate more considerations for path recommendation, such as traffic, sightseeing, hotels etc.We also consider to design and implement a friendly visual interface on mobile devices so that users can easily express their preference as well as to have a clear idea on the effectiveness of their choices.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61173140, 61070017) and the National Science & Technology Pillar Program (2012BAF10B03-3).

References

[1] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Proc. of the 2nd VLDB Workshop on Secure Data Management*, pages 185–199, 2005.

[2] R. Chen, B. Fung, B. C. Desai, and N. M. Sossou. Differentially private transit data publication: A case study on the montreal transportation system. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–221. ACM, 2012.

[3] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang. Privacy-preserving trajectory data publishing by local suppression. *Information Sciences*, 231(0):83–97, 2013.

[4] C.-Y. Chow and M. F. Mokbel. Trajectory privacy in location-based services and data publication. *ACM SIGKDD*, 13(1), 2011.

[5] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. Framework for security and privacy in automotive telematics. In *Proceeding of the 2nd international workshop on Mobile commerce(WMC)*, pages 25–32, 2002.

[6] K. E. Foote and M. Lynch. *Geographic Information Systems as an Integrating Technology: Context, Concepts, and Definitions*. ESRI, 2011.

[7] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[8] M. Hardt and S. Nath. Privacy-aware personalization for mobile advertising. In *Proceedings of the 2012 ACM conference on Computer and Communications Security (CCS '12)*, pages 662–673, 2012.

[9] S. Ilarri, C. Bobed, and E. Mena. An approach to process continuous location-dependent queries on moving objects with support for location granules. *Journal of Systems and Software*, 84(8):1327–1350, 2011.

[10] G. Ji, Y. Sun, and X. Ma. Path planning for privacy preserving in location based service. In *Proceeding of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 162 – 167, 2011.

[11] Y. J. Jung, K. H. Ryu, M. S. Shin, and S. Nittel. Historical index structure for reducing insertion and search cost in lbs. *Journal of Systems and Software*, 83(8):1500–1511, 2010.

[12] W. Luo, H. Tan, L. Chen, and L. M. Ni. Finding time period-based most frequent path in big trajectory data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 713–724, 2013.

[13] H. Pham, C. Shahabi, and Y. Liu. Ebm: an entropy-based model to infer social strength from spatiotemporal data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13)*, pages 265–276, 2013.

[14] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3310–3317, 1994.

[15] Y. Sun, X. Su, B. Zhao, and J. Su. Mix-zones deployment for location privacy preservation in vehicle communications. *compute and information technology (CIT 2010)*, (IEEE):2825–2830, 2010.

[16] Y. Wang, L. Wang, and B. Fung. Preserving privacy for location-based services with continuous queries. In *Proceeding of the IEEE International Conference on Communication (ICC)*, pages 1–5, 2009.

[17] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *Proceeding of The 27th IEEE Conference on Computer Communications (INFOCOM)*, pages 547–555, 2008.

[18] A. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 254–265, 2013.