國立交通大學

資訊科學與工程研究所

碩士論文

行動計算環境中的隱私權保護機制

Protecting Moving Trajectories with Dummies

研 究 生：游敦皓

指導教授：彭文志　教授

中華民國 九十六 年六月

行動計算環境中的隱私權保護機制
Protecting Moving Trajectories with Dummies

研究生：游敦皓 Student：Tun-Hao You

指導教授：彭文志 Advisor：Wen-Chin Peng

國立交通大學
資訊科學與工程研究所
碩士論文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

摘　　要

在本篇論文中探討了利用假人的技術(Dummy-based anonymization techniques)來保護在行動環境中的使用者位置隱私權。利用製造移動方式接近真人的假人可以保護到在行動環境中的使用者位置隱私權。然而，透過監控使用者的長期的移動行為，他的移動路徑仍然會被暴露出來。我們認為當使用者的移動路徑被暴露出來之後，他的位置也同時被暴露出來了。因此，為了保護用戶位置隱私，保護使用者的移動路徑是關鍵的。我們提出了兩個製造一致的路徑的方法，並且提出了三個測量的變數，分別是短期暴露率(short-term disclosure)、長期暴露率(long-term disclosure)和距離差異(distance deviation)。除此之外，我們也考慮到了使用者可能會有多條的習慣移動路徑，並提出方法加以保護。實驗的結果證明我們所提出的方法能比目前的方法更能保護在行動環境中使用者的位置隱私權。

關鍵字：位置隱私權，使用者移動路徑，行動位置感知服務

Abstract

Dummy-based anonymization techniques for protecting location privacy of mobile users have been proposed in the literature. By generating dummies that move in human-like trajectories, this method shows that location privacy of mobile users can be preserved. However, by monitoring long-term movement patterns of users, the trajectories of mobile users can still be exposed. We argue that, once the trajectory of a user is identified, locations of the user is exposed. Thus, it's critical to protect the moving trajectories of mobile users in order to preserve user location privacy. We propose two schemes that generate consistent movement patterns in a long run. Guided by three parameters in user specified privacy profile, namely, *short-term disclosure*, *long-term disclosure* and *distance deviation*, the proposed schemes derive movement trajectories for dummies. Moreover, since a user may has multiple frequent trajectories, we proposed several schemes to deal with this scenario. Experimental results show that our proposed schemes are more effective than existing work in protecting moving trajectories of mobile users and their location privacy.

Keywords —Location privacy, user movement patterns, location-based services.

致　謝

兩年的碩士生涯一眨眼的時間就過了，這是一段很扎實的學習過程。要感謝的人真的太多了，不能單單的只用『謝天』兩個字就帶過。首先要先感謝我的指導老師 – 彭文志老師，在這兩年所給我的指導和照顧，給了我論文不少的想法批評和指教，讓我的論文可以順利上了 IEEE 的 Workshop 以及被 invite 到journal 去。還要感謝賓州大學的李旺謙教授，雖然人遠在國外，但是仍然不辭辛勞幫忙改善我的論文。其次，要感謝我的口試委員陳良弼教授和黃俊龍教授，在口試的時候提供了很多的意見，讓我可以針對本論文作更進一步的改進。除了老師們之外，我也要感謝系辦小姐們的幫忙，適時的提醒我一些該注意的事項，在比賽和獎學金申請的方面也幫了不少的忙，讓我的碩士生涯可以過的非常的順利。

在實驗室中，首先要感謝學長們，無論是博班的學長 – 洪智傑，或是上一屆的碩班學長 – 李志劭、張民憲、楊慧友和蕭向彥學長，都給了我不少在學習和研究上的幫助，從他們那邊我學到了不少作研究的方法。其次還要感謝實驗室的同學，鄉民、boy、佳欣大家相互鼓勵與打氣或是談論八卦的畫面我會永遠記得。還有所有的學弟們，因為有他們的陪伴，讓我的生活充滿了歡樂。除此之外，還要感謝一群死黨，言叡、小咪、大方、建平、大雕、P 嫂、骨感…等，你們在我作研究苦悶的時候，給我帶來了不少的歡樂，還有平日一同去打球一起出去玩，更讓我的研究生生涯過的十分的精彩。還有一同參與比賽的夥伴們，不管是lab 的學長同學學弟或是我的好朋友們，真的很感謝你們，讓我的碩士班生涯過的非常的不一樣，除了論文之外，還多出了非常多比賽的經驗。

最後要感謝我的家人，尤其是我爸媽，您們的養育之恩，以及給予我衣食無憂的環境，讓我可以專心學習而無後顧之憂，而且常常給予我鼓勵，讓我可以更有信心的去面對未來的挑戰。謝謝您們，永遠站在我的背後支持我，給予我莫大的精神鼓勵，謝謝。

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Location-based services (LBSs) have emerged as one of the killer applications for mobile computing and wireless data services. These LBSs are critical to public safety, transportation, emergency response, and disaster management, while providing great market values to companies and industries. Due to the unrestricted mobility of users in the mobile computing environments, users are often interested in acquiring information or services related to their current locations. Thus, very frequently locations information of users are submitted along with queries to the LBS servers. Examples of such queries include finding the nearest restaurants to a user (k nearest neighbor query) and finding ATMs within 500 meters from a user's current location (range query). While LBSs have shown to be valuable to users' daily life, on the other hand, they also expose extraordinary threats to user privacy. If not well protected, the location information of users may be misused by some untrustworthy service providers or stolen by hackers. Once the location information is exposed, adversaries may dig for cues to invades user privacy. Obviously, it is important to protect location privacy.

Recently, the problem of location privacy preserving has received growing interests from the research community [2] [5] [15] [21] [18] [3] [16]. These studies aim at protecting exact location information of users from the potential abuse of LBS providers and hackers. Two primary approaches have been considered, including 1) *trusted anonymizer* based approach; and 2) *client* based approach. In the former, users submit their queries to the LBSs via a trusted server (which is different from the LBS server), such as a base station in the cellular networks. This trusted anonymizer transforms the exact locations of a number of users into a *cloaked spatial area* in accordance with privacy requirements set by users in order to obtain data or services from the LBSs [10] [21] [11]. The second approach assumes no trusted server. Thus, clients are responsible for anonymizing their own location information before transmitting queries to the LBS servers. By issuing several fake locations along with its true location to the LBSs, clients may obtain redundant information or services corresponding to the submitted locations while preserving their location information [18] [19]. Unwanted information is filtered locally to obtain the final query results. In both approaches, the true location of a user is either 1) not distinguishable from other users (the trusted anonymizer based approach), or 2) not distinguishable from the fake locations (the client based approach). Since a trusted server is not always available, in this paper, we tackle some issues faced in the client based approach.

Motivation and Problems. Without relying on a trusted server, generating fake user locations (called *dummies*[1]) for location-dependent queries has been shown to be an effective way to preserve location privacy [18]. In addition to generate dummies based on the user locations, these prior works propose to generate dum-

[1] We follow the terminology used in [18] to name the fake user locations as *dummy locations* and *dummies* in short.

mies based on realistic user movements. However, prior works don't consider a well-recognized observation, i.e., moving behaviors of users usually follow certain patterns [22] [23]. To demonstrate user moving patterns of users, Table 1.1 shows an example of real log data from INFATI [17]. INFATI is the first Intelligent Speed Adaptation development project in Denmark. The project is carried out by Aalborg University. In this project, every car was equipped with a Global Positioning System(GPS) and collects day to day movements of 20 private cars on the road network of Aalborg during two months. The log data contains several attributes, such as car's id, driver's id(one car may be driven by more than one person), data and time, XY coordinate received from GPS receiver, speed and street code. Figure 1.1 shows one car's trajectories, where the XY coordinate are the position coordinates received from GPS receiver. By exploring data mining techniques such as spatial-temporal sequential pattern mining [4] or moving pattern mining [22] [23], adversaries only need to collect enough user's moving logs and can get the frequent moving patterns of user easily. Notice that once trajectories of users are disclosed, adversaries are able to utilize external databases to find even user identity, which incurs more serious disclose of location privacy. The above scenario is referred to the linking attack problem in location privacy, showing the justification of protecting user trajectories. Thus, generating dummies should consider not only realistic user movements but also follow certain patterns.

The problem we study in this paper could be best understood by an example shown in Figure 1.2. In the figure, the solid line denotes the moving trajectory of a true user (denoted as T) and the dotted lines are generated trajectories of dummies (denoted as $d1$ and $d2$). Since true users usually exhibit certain human moving behavior, one is able to identify the solid line as a true user based on the

id	carid	driverid	rdate	rtime	xcoord	ycoord	SPD	strtcord
991	12	0	91200	130310	553570	6315889	44	5490
992	12	0	91200	130311	553562	6315859	42	5490
993	12	0	91200	130312	553554	6315833	43	5490
994	12	0	91200	130313	553547	6315806	38	5490
995	12	0	91200	130314	553541	6315779	42	5490
996	12	0	91200	130315	553535	6315752	44	5490

Table 1.1: The real GPS log data

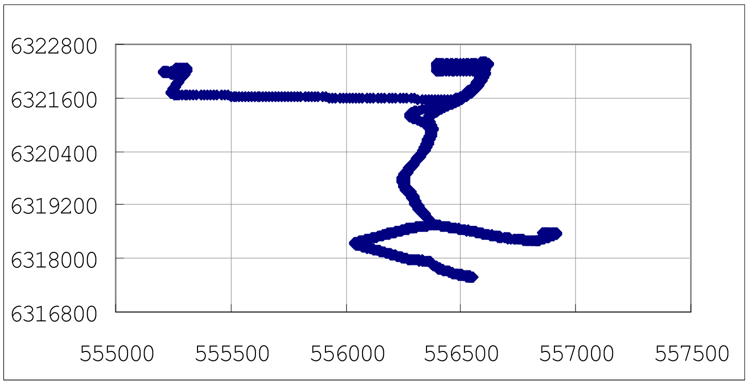
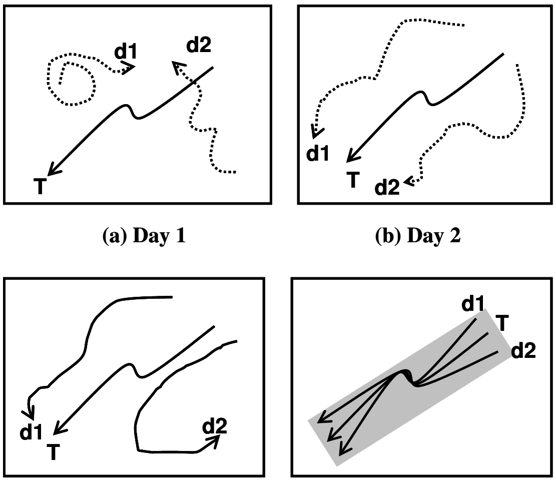


Figure 1.1: Moving trajectory of one example car.

typical moving behavior of humans (as shown in Fig. 1.2(a)). Thus, it's important to generate dummy trajectories based on human moving behavior (as shown in Fig. 1.2(b)). Even though this effort may reduce the chance of the true moving trajectory being identified, a *long-term movement pattern* can be collected to filter inconsistent trajectories. For example, comparing the current trajectories (in Fig. 1.2(c)) and trajectories collected in a different day (e.g., Fig. 1.2(b)), one can tell T is the true trajectory of user. Once the moving trajectory of true user is identified, locations (i.e., not only the current location but also the past locations) of the user is disclosed. Thus, it's important to generate dummies that not only demonstrate moving behavior of users but also follow consistent, long-term movement patterns.

Given that the adversaries obtain a set of trajectories, they will have difficulty determining the true trajectory of a user if users generate dummies following certain movement patterns. However, the user trajectory is still disclosed to a certain degree. Therefore, we use *disclosure* to denote the probability that the user trajectory may be correctly identified by the adversaries. For example, in Fig. 1.2(c), three trajectories are collected and thus the disclosure is $\frac{1}{3}$. To reduce the disclosure, a naive approach is to simply increase the number of dummies, which however incurs overhead in terms of query message length and thus communication and client processing costs. Thus, in this paper, we propose to generate *intersecting dummy trajectories* aiming at increasing the number of possible trajectories from the adversaries' perspective and thus decreasing disclosure of the user trajectory.

Nevertheless, an issue exists with this intersecting trajectories. When the generated trajectories are too close to the true trajectory, the locations of a user may still be exposed, e.g., Fig. 1.2(d) shows an example where the user's moving trajec-



(a) Day 1

(b) Day 2

(c) Case 1

(d) Case 2

Figure 1.2: Moving trajectories of user and dummies.

tory (the shadowed path) can be identified. Thus, our design of dummy generation schemes also take the factor of *distance deviation* among trajectories into consideration. Our approach is to allow users to set up their privacy profile in terms of disclosures (both short-term and long-term) and distance deviation (more details to be discussed in Section 2). We propose two schemes, namely, *random pattern* and *intersection pattern*, to generate dummy trajectories based on the privacy profile. Furthermore, since a user may have more than one moving trajectories, we develop several schemes to protect multiple user moving trajectories with the purpose of using minimal number of dummies. Performance of proposed schemes is comparatively analyzed and sensitivity analysis on several design parameters is conducted. Experimental results show that by generating dummies based on moving patterns, our schemes perform better than the existing techniques.

Organization. The rest of this paper is organized as follows. We first describe some related works in Section 2. Section 3 presents preliminaries, including attacker model and user profile. Our proposal of dummy trajectory generation schemes and the selection of multiple path are presented in Section 4. Section 5 shows our performance study. Section 6 concludes this paper.

Chapter 2

Related works

A significant amount of research efforts have been elaborated on location privacy. Generally speaking, methods of guaranteeing location privacy could protect either user's identification or user's location. Because most Location Information(*LocInfo*) can be a variation of the definition introduced in [8], the triple of the following form: *Position, Time, ID.* Thus, *LocInfo* is defined as a combination of the "Position" that the entity with identifier "ID" maintained at time "Time", within a given coordinate system. Position and ID are considered as the most important part in *LocInfo*. Protecting identifier is to hide user's identifier that attacker cannot take apart who is exactly in this position at this time. Different from ID, Protecting position is to protect user's location from being disclosed that the attacker cannot recognize the user's exactly position at this time.

Specifically, to protect ID in *LocInfo*, the authors in [2] [5] proposed the concept of the mix zone [2] [5]. They assumed the LBS application providers are hostile adversaries, and suggested that application users hide their own identifier from providers [3]. So they proposed the mix zone concept in which a trust third party removes all samples before is passes location samples to the LBS application providers. Instead of static mix zones, the authors in [16] implemented a mix zone concept by exploring a silent period. That protects user from correlation attack which means a method of utilizing the temporal and spatial correlation between the old and new pseudonym of nodes.

To protect user location, a consider amount of research works are conducted [15] [21] [18] [19]. As described before, these works could be further classified according to the architecture of location privacy. With trusted servers, the authors in [13] [12] [11] proposed a cloaking algorithm to blur the resolution of location information along spatial and temporal dimensions. The above algorithm exploits k-anonymity concept. Based on k-anonymity, the authors in [10] [9] devised a personalized and customized k-anonymity model which assume a different k-anonymity requirement for each user. A framework Casper was developed in [21], where a grid-based pyramid structure is implemented to index all user locations. Moreover, privacy-aware query processing is developed when cloaked spatial areas are used as query predicates. Without trusted servers,the authors in [7] proposed 1 P2P structure to protect user's privacy. Explicitly, before issuing any location-based service queries, mobile users will form a group from his/her neighboring peers via multi-hop routing. Then, the spatial cloaked area is computed as the region that covers the entire group of peers. In addition, the authors in [18] [19] proposed an algorithm to generate dummy locations to protect not only location privacy but also true user identifications.

To the best of our knowledge, prior works neither address location privacy issues from long-term observation nor emphasize the necessity of protecting user moving patterns, let alone generating dummies with moving patterns. This paper differentiates from other papers.

Chapter 3

Preliminaries

In this section, some preliminaries are given. In Section 3.1, assumptions and notations used in this paper are presented. The attacker model and user privacy profiles are described in Section 3.2.

3.1 Assumptions and Notations

We assume no trusted server available for location anonymization. Wireless networks are only responsible for communication and will not reveal locations of mobile users. Mobile clients are location aware (via GPS or network based positioning techniques). To facilitate the presentation of our paper, suppose that users are free to move in the space divided into grid cells. Each grid cell has a cell identifier (x, y) indicating that this cell is located at the x column and the y row of the space. The granularity of this representation is determined by the number of grid cells. With larger number of grid cells, finer granularity we have. Note that using cell identification could achieve a certain level of location privacy even if adversaries guess the true location among a set of location data.

Upon a user query message, the mobile client U$_i$ first sends to the LBSs through an authenticated and encrypted connection that adversary cannot hijack the message. A query message issued by a mobile user U_i to a LBS server at time slot t is defined as $M = \{uid, \langle L_i^t, L_{d1}^t, L_{d2}^t...L_{dn}^t\rangle\, Q\}$, where uid is the pseudonym user identification, L_i^t is the true user location, $L_{d1}^t, L_{d2}^t, ..., L_{dn}^t$ are n dummy locations, respectively, and Q is the location-dependent query issued. Therefore, given m consecutive queries, we define the trajectory of a moving client in 2-dimensional (2D) spaces as a sequence $\{L_i^1, L_i^2, ..., L_i^m\}$, while the trajectory of dummy x is { $L_{dx}^1, L_{dx}^2, ..., L_{dx}^m\}$) where $L_i^a \in \mathbb{R}^2$ $(a = 1, ..., m)$ describe locations in t_a, $t_1 < t_2 < ... < t_m \in \mathbb{T}$ are irregularly spaced but temporally ordered time instances, i.e., gaps are allowed. Here, L_i^j (and L_{dx}^m, respectively) denotes the location of user U$_i$ (and dummy d_x), respectively at the jth time slot. Denote a trajectory of mobile user U$_i$ as P$_i$ = $\{PL_i^1, PL_i^2, ..., PL_i^m\}$, where PL_i^j is the location of mobile user U$_i$ at the jth time slot. Suppose that the length of trajectories is m and the maximum user's moving velocity is defined as V_{max}.

3.2 Attacker model

In this section, we describe how adversaries collect and utilize data mining techniques to mine user moving patterns. Explicitly, adversaries can sequential pattern mining to discover user moving patterns, thereby disclosing user trajectories. [1] [14] [4]. Consider an example shown in Figure 3.1, where Table 3.1 is the moving log. As can be seen in Table 3.1, there are five movement sequences, where each location in a movement sequence is the cell identification defined in Section 3.1. Given the minimum support 2, it can be verified that a moving pattern (i.e.,

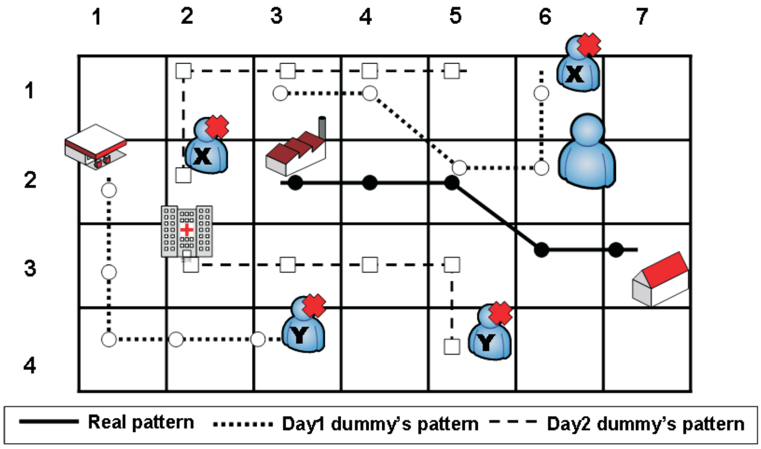


Figure 3.1: Mobile behavior of mobile user A and his dummies (i.e., X and Y)

(7,3)⇒(6,3)⇒(5,2)⇒(4,2)⇒(3,2)) is discovered.

Once moving patterns are discovered, one could easilyidentify true user positions, no matter how dummies are generated. The main theme of this paper is to prevent a privacy threat resulting in long term movement observations. Three formal definitions of privacy preservation are given as follows:

Definition 1. Given an area size $A \in \mathbb{R}^+$, a mobile user's trajectory P_i and n dummies, the probability of successfully identifying the true user's trajectory is smaller than the profile user define.

Definition 2. Given an area size $A \in \mathbb{R}^+$, a mobile user's current location L and n dummies, the probability of successfully identifying the true user's current location is smaller than the profile user define.

Definition 3. Given an area size $A \in \mathbb{R}^+$, a mobile user's trajectory P_i and n dummies, the average distance difference among trajectories of dummies and the user must be larger than the profile user define.

By these definitions, an adversary cannot distinguish user's trajectory or locations. As such, both location and trajectory privacy can be preserved. Users may

	User Id	T_1	T_2	T_3	T_4	T_5
Day 1	User A	(7,3)	(6,3)	(5,2)	(4,2)	(3,2)
	Dummy X	(6,1)	(6,2)	(5,2)	(4,1)	(3,1)
	Dummy Y	(3,4)	(2,4)	(1,4)	(1,3)	(1,2)
Day 2	User A	(7,3)	(6,3)	(5,2)	(4,2)	(3,2)
	Dummy X	(2,2)	(2,1)	(3,1)	(4,1)	(5,1)
	Dummy Y	(5,4)	(5,3)	(4,3)	(3,3)	(2,3)

Table 3.1: An example of query log for mobile user A

set up their privacy profile, which is specified by the following three parameters:

1. **Short-term Disclosure (SD)**: This parameter specifies requirement for protecting the current user location. Thus, given a set of current locations (including true and dummy locations), SD is the probability of successfully identifying the true user location, i.e., $SD = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|D_i|}$, where m is the number of time slots in a trajectory, D_i is the set of true and dummy locations at the ith time slot, and $|D_i|$ is the size of D_i.

2. **Long-term Disclosure (LD)**: This parameter specifies requirement for protecting the user trajectory. Given n trajectories, among which k trajectories have intersected with other trajectories and $(n-k)$ trajectories do not have any intersection. Thus, for those $(n-k)$ trajectories, we have exactly $(n-k)$ possible trajectories. For those k trajectories, we may enumerate all possible trajectories by exhaustively traversing intersections from the start point of each trajectory to the end point. In order not to distract readers from the main theme of this paper, we simply denote the number of possible trajecto-

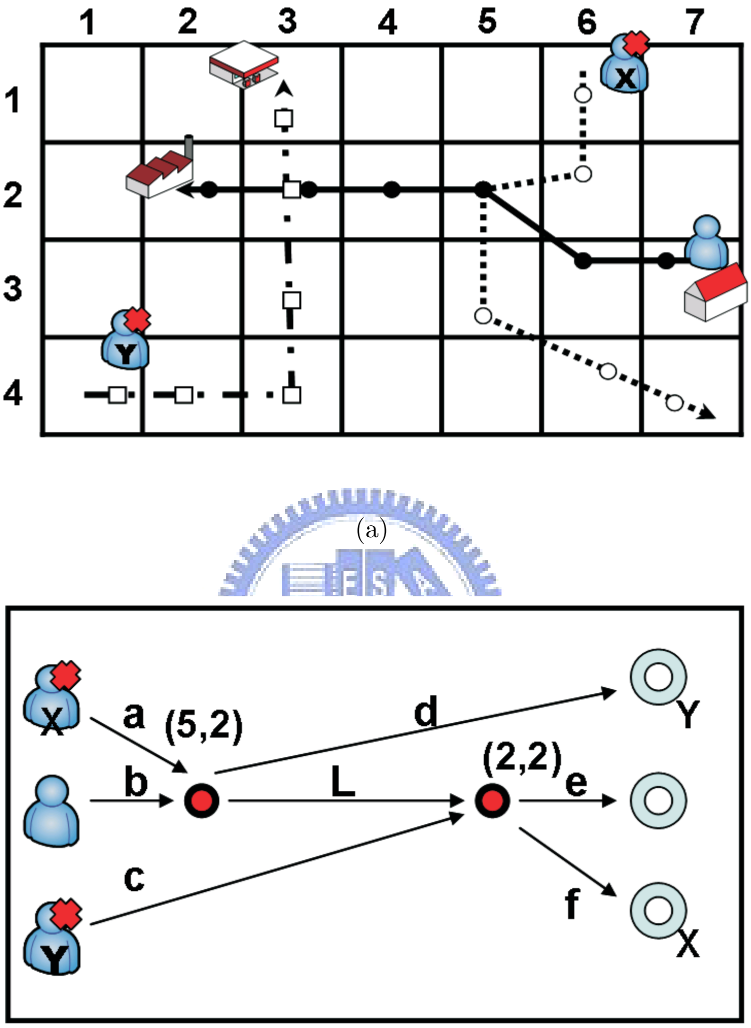
Time slot	1	2	3	4	5	6
True user	(7,3)	(6,3)	(5,2)	(4,2)	(3,2)	(2,2)
Dummy X	(6,1)	(6,2)	(5,2)	(5,3)	(6,4)	(7,4)
Dummy Y	(1,4)	(2,4)	(3,4)	(3,3)	(3,2)	(3,1)
$\lvert D_i \rvert$	3	3	2	3	2	3
Dist	4.2	2.6	1.4	1.4	1.8	3.4

Table 3.2: Privacy measurement of dummy trajectories.

ries by BFS(Breadth-First-Search) among k trajectories as T_k. Consequently, we have LD as $\frac{1}{T_k+(n-k)}$.

3. **Distance deviation:** The distance deviation (dst) is the average of distance difference among trajectories of dummies and the user. As a result, dst of mobile user U$_i$ is formulated as $\frac{1}{m} * \frac{1}{n} * \sum_{k=1}^{n} \sum_{j=1}^{m} dist(PL_{i,}^{j}\ L_{dk}^{j})$, where $dist$ is distance between the true user location and dummy locations in unit of cell size.

Figure 3.2 shows an example of generated dummy trajectories with intersections, while Table 3.2 shows the trajectories as well as the number of current locations (D_i) and distance deviation at different time slots. Thus, we can derive $SD = \frac{1}{6}(\frac{1}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + \frac{1}{3}) = \frac{7}{18}$. Furthermore, for each time slot, we could derive distance differences between dummy and true trajectories to obtain the average distance deviation as 2.47. To facilitate the derivation of total possible trajectories, Figure 3.2(a) is transformed into Figure 3.2(b), where intersection points (i.e., cell (5, 2) and (3, 2)) are marked. Since these three trajectories have two intersection points, it can be verified that we have 8 possible trajectories (i.e.,



(a)

(b)

Figure 3.2: Trajectories with Intersections.

ad, aLe, aLf, bd, bLe, bLf, ce, cf). As such, we could have long-term disclosure $LD = \frac{1}{8+(3-3)} = \frac{1}{8}$.

Chapter 4

Generating Dummies with Patterns

Given a privacy profile, our goal is to generate dummy trajectories that satisfy the user-define parameters in privacy profile *(SD, LD, dst)*. In this section, we propose several schemes, namely, *random pattern scheme* and *intersection pattern-based schemes* to generate dummies that exhibit long-term user movement patterns.

4.1 Random Pattern Scheme

In this scheme, the starting point and the destination of a dummy are first selected. Then, the grid cells between the starting point and the destination are determined based on the speed of a dummy and four movement types, including horizontal movement, vertical movement, both and stay in the current position. Because the humans moving speed is limited, the velocity of dummies should also be limited (i.e. smaller than V_{max}). Figure 4.1(b) is an easy example with random dummy generation.

In this scheme, a dummy will move randomly from the starting point towards the destination. This naive scheme demonstrates that even after a long term observation, it's difficult for adversaries to identify true user since dummies also exhibits long term, consistent movement patterns. Given the original user moving trajectory in Figure 4.1(a). Figure 4.1 shows a dummy trajectory generated by *random pattern scheme*. However, without taking into account factors such as distance deviation, this scheme simply include more dummies when the privacy requirements are not satisfied.

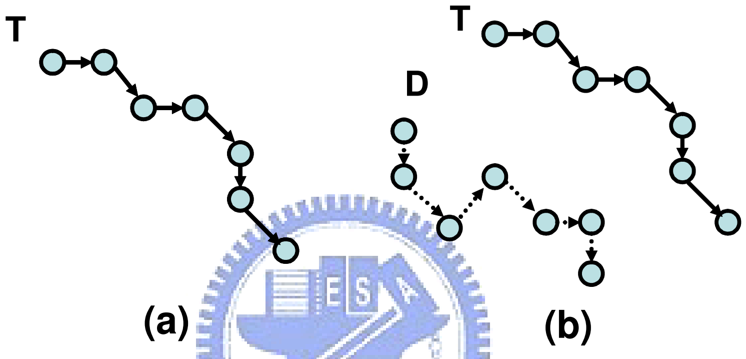


Figure 4.1: (a)original pattern (b)random pattern

4.2 Intersection Pattern-based Scheme

The main idea in this scheme is to have some intersections between trajectories of dummies and the real user that can generate more possible trajectories. Adversaries are harder to identify a true user trajectory from a set of possible trajectories. The benefits of using intersection pattern scheme are described below. First, if the number of intersections among a user trajectory dummy and trajectories are increased, one can use smaller number of dummies to satisfy the *LD* in user profile. Second, it is hard for adversaries to tell which trajectories are made by dummies. Third, if dummies have some intersections with true user by using caching technique, data requested by dummies are used by a true user in his future movement.

In this intersection pattern scheme, generated dummy trajectories should fulfil the privacy profile of the user. Since there are three requirements in privacy profiles, our approach is to first select intersections from the candidate set. The candidate set depends on some constraints(i.e., important place, query cost) which we will discuss later. Afterward, our approach derive the solution space for the requirement of distance derivation. Then, within this solution space, we obtain the short-term and long-term disclosures (i.e., SD and LD). The trajectories with disclosures smaller than what specified are selected as dummy trajectories. With proper selection of dummy trajectories, we can minimize the number of dummies so as to satisfy the user privacy requirements. In view of the concept we mentioned above, we proposed two kinds of intersection dummy generation: *rotation dummy generation* and *k-intersect dummy generation*.

4.2.1 Rotation Dummy Generation

Given a user trajectory, we generate a new trajectory for a dummy by rotating the known user trajectory in the rotation pattern scheme. To perform a rotation on a user's sequential pattern, the rotating point and the rotating angle must be decided. Clearly, the rotation point of user trajectory is an intersection point. Consider Figure 4.2 as an example, where the dotted point is the rotate point and θ is the rotate angle.

In order to derive the solution space for the distance derivation (i.e., dst), both the *rotation angle* and the *rotation point* within a true user trajectory have a great impact on the distance deviation. To simplify the derivation of distance deviation,

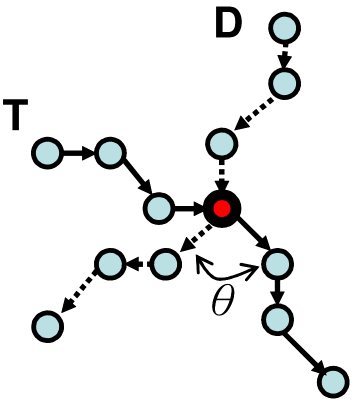


Figure 4.2: An example of executing rotate pattern scheme

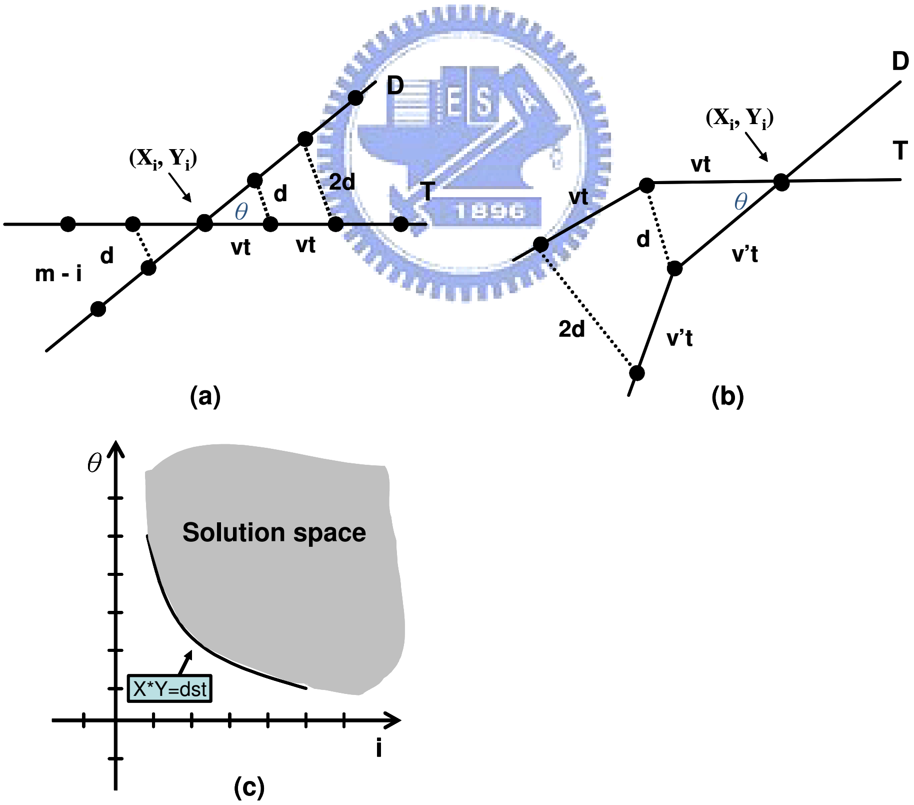


Figure 4.3: Solution space for distance derivation

assume that we have a true user trajectory in Figure 4.3(a), we assume the real user's moving speed is v and the distance between two consecutive movements can be represented as vt. The rotation point is the location at the ith time slot in a true user trajectory, denoted as (X_i, Y_i), and the rotation angle is θ. d is the distance difference between the location of a true user and that of a dummy at the $(i+1)$th time slot. According to the cosine theorem, we have $d = \sqrt{2}|vt|\sqrt{1-\cos\theta}$. Hence, we could derive that the distance deviation of these two trajectories as follows:

$$\begin{aligned} dst^r &= \frac{1}{m} * ((d + ... + id) + (d + ... + (m-i)d)) \\ &= \sqrt{2}|vt|\sqrt{1-\cos\theta} * (\sum_{j=0}^{i} j + \sum_{j=0}^{m-i} j) \end{aligned}$$

If user trajectories are not straight lines, the above derivation is still held. Consider two realistic trajectories in Figure 4.3(b). In order to make the distance of two corresponding points at $(i+1)$th time slot be d, we could dynamically set up the dummy's speed to $v'(0 \leqq v' \leqq V_{max})$. The distance of the following points along these two trajectories is the multiple of d. Similarly, we can get the following formulas:

$$d = \sqrt{(vt)^2 + (v't)^2 - 2vv't^2\cos\theta}$$

Because v and v' are also constant, we can derive to the following formula.

$$dst^r = \sqrt{C_1 - C_2\cos\theta} * (\sum_{j=0}^{i} j + \sum_{j=0}^{m-i} j), whereC_1andC_2areconstraints$$

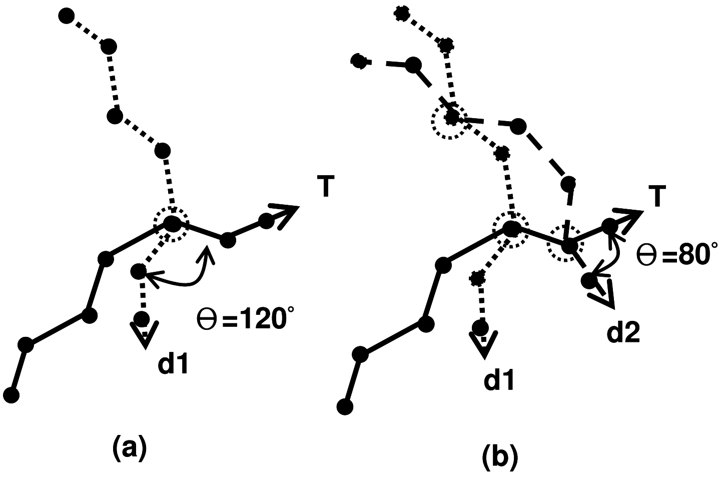


Figure 4.4: An example of the rotation pattern scheme.

From the above derivation, we could conclude that both the rotation angle (i.e., θ) and the rotation point (i.e., i) have an impact on the distance deviation. Assume that we have n dummies trajectories and the distance deviation of n dummy trajectories is dst_n. If one dummy is added into the set of n dummies, the $(n+1)$ dummies should be larger or equal to the requirement of distance deviation (i.e., dst). Thus, we have the following formula:

$$\frac{n}{n+1}dst_n + \frac{1}{n+1}dst^r \geq dst$$

Consequently, when one dummy is added into the current set of dummies, this dummy should have a constraint on $dst^r \geq (n+1)dst - n(dst_n)$. Therefore, we could have a solution space shown in Figure 4.3(c). For each point (expressed by (θ, i)) in the solution space, we should calculate the corresponding disclosure. Hence, a solution point with the minimal hit probabilities is selected. If the hit probabilities are still larger than the required hit probabilities, one should repeat the above procedure to add one additional dummy until the all privacy criteria are satisfied.

For example, consider a true trajectory (the line marked with T) in Figure 4.4(a) and a user privacy profile (i.e., $SD = 40\%$, $LD = 10\%$, $dst = 2.1$). Initially, there is no dummy (i.e., $n = 0$) and $dst_0 = 0$. As such, we could have $dst^r \geq (0+1)*2.1$. Table 4.1(a) show some selected possible solution space when the number of dummy is 0. In Table 4.1(a), the solution (i.e., $(120^{\mathbf{o}}, 5)$) is selected and then n is increased to 1. The value of dst_1 is updated accordingly. However, since disclosures are still larger than the required values (i.e., $56.25\% \geq 40\%$ and $25\% \geq 10\%$), we should add one more dummy to reduce the disclosures. Following the same procedure, we have $dst^r \geq (1+1)*2.1 - 1*2.8$ and Table 4.1(b) is the solution space when the number of dummy is one. From Table 4.1(b), one could select $(80^{\mathbf{o}}, 6)$ since the corresponding disclosures is smaller than the required values and it needn't to add one more dummy. Hence, Figure 4.4(b) shows the final dummy trajectories.

If we can generate more intersections between each trajectory[1], we can use less dummies to generate more possible trajectories for lower disclosures. Assume we add one dummy $d2$ in Figure 4.5, dummy $d2$ has intersecttions with real user T and dummy $d1$. The intersection of trajectory i and j denotes as $I_{i,j}$. We can indicate the distance between $I_{T,d1}$ and $I_{d1,d2}$ is L, and the distance between $I_{T,d2}$ and $I_{d1,d2}$ is D in Figure 4.5. Assume the rotation angle between T and $d1$ is α and rotation angle between T and $d2$ is β, we want to find the suitable rotation point and rotation angle that can make $d2$ intersect with $d1$ and T. We use sine theorem in Figure 4.5.

$$\frac{D}{\sin(\beta - \alpha)} = \frac{L}{\sin(180^\circ - \beta)}$$

Then we can derive the following formula.

[1] Note that: not only between user and dummy

θ	i	SD	LD
120	5	56.25%	25%*
50	3	56.25%	25%
180	1	56.25%	25%

(a). Solution space when n=0

θ	i	SD	LD
170	8	37.5%	16.67%
120	7	37.5%	12.5%
80	6	39.6%	8.33%*

(b). Solution space when n=1

Table 4.1: A solution space.

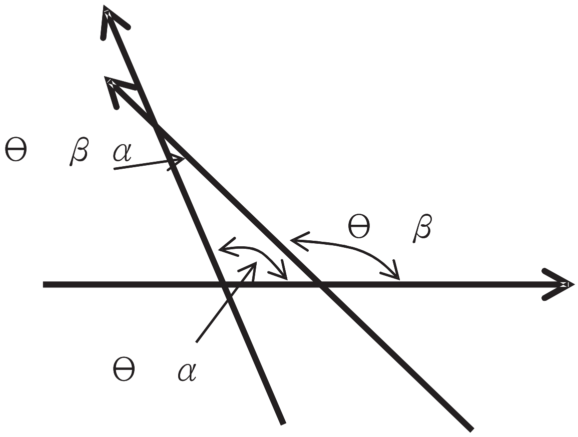


Figure 4.5: Generate more intersections between dummy trajectories.

If $\beta \geq \alpha$

$$\frac{m-i}{D} \geq \frac{L}{D} = \frac{\sin(180^\circ - \beta)}{\sin(\beta - \alpha)} = \frac{\sin\beta}{\sin(\beta - \alpha)}$$

If $\beta < \alpha$

$$\frac{m-i}{D} \geq \frac{L}{D} = \frac{\sin(180^\circ - \alpha)}{\sin(\alpha - \beta)} = \frac{\sin\alpha}{\sin(\alpha - \beta)}$$

From the above derivation, we could reduce the candidate sets of rotation point i and rotation angle β because i and β must satisfy the formula to make another intersection. Consequentially, the solution space is further reduces. Based on achieving user's all requirements, when adding one dummy each time, we intend to generate more intersections to reduce the total number of dummies.

4.2.2 K-intersect Dummy Generation

Rotation dummy generation only has one intersection between user's trajectory and dummies. If the number of intersections between a user and the dummies are increased, it is more difficult for adversaries to figure out the user's true trajectory and *LD* is thus decreased. But the *dst* is also influence a lot. Besides, in the situation that attackers has some background knowledge of users, the intersections between a user and the dummies can decrease the exposure of users' trajectory and thus protect users' location privacy. In this scheme, we increase the number of intersections between user trajectory and the dummy trajectories.

This scheme selects k points from the intersection candidate set to be the intersection points. Then the paths between the intersection points by the randomized dummy generation. The intersection points can be represented as $C = C_1, C_2, ..., C_k$ where C_i is the ith intersection point. The trajectory not included by the intersec-

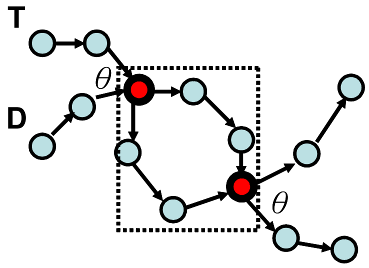


Figure 4.6: An example of 2-intersect pattern scheme

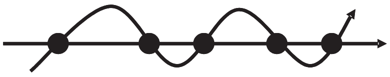


Figure 4.7: The problem of multiple intersections

tion points, the rotation dummy generation is used. Explicitly, C_1 and C_k can be regards as the rotation points and the following formula is used:

$$\frac{(m-k)*dst_r + k*dst^k}{m} \geqq dst$$

$$\Longrightarrow dst_r \geqq \frac{m*dst - dst^k*k}{m-k}$$

dst^k means the distance deviation and a means the length of trajectory between the cutting points C_1 and C_k. The combinations of the paths describe above will be outputted as the dummy's moving pattern. An example of 2-intersect dummy generation example is shown in Figure 4.6, where the dotted circles are the intersection points. The trajectory in dotted square is generated by randomized dummy approach and beyond the square is generated by rotation dummy approach.

Note that increasing the number of intersection points is not always good. Because the more intersections between user's and dummy's trajectories the less distance derivation the user has. For example, in Figure 4.7, the dummy's trajectory is too closed to user trajectory. That will cause the injury of *dst* and attacker may break user's privacy level. In this paper, we set the value of k is two to make more intersection than the rotation dummy generation and not hurt the quality of privacy in value *dst*.

As mentioned before, the selection of intersection candidate sets depends on several factors. We explore two factors to select candidate sets. First, the candidate sets should not be an *important place* to the user. For example, if we choose a users home as the intersection point, dummies and the user will stay in the same cell for a long period of time. Therefore the dummy cannot effectively protect the users location privacy. Second, in order to increase the cache utilization, we develop a cache scheme to determine intersection points in which users are likely to visit.

To determine which places are important, we should consider the staying time and sensitive places. Obviously, choosing a place that the user stays for a long time as the intersection point will decrease the location anonymity. The other type of important place is *sensitive area* [6], sensitive area means the places(e.g., hospital, nightclub) user don't want people know that he is inside. For example, when shopping in a mall, most people may not be very concerned even if their locations are known. However, users may worry about their locations exposed (e,g, hospital). Our method will exclude those important places, which including the place that user stayed for more than threshold slots T_{max} and user-specific sensitive area S_i, to form the remainder for candidate position set.

In dummy techniques, communication cost increases are generated as a side effect. Since, the service provider must create a reply message not only for the true position data but also for the dummy. In dummy methods, LBSs will return both users' and dummies' data, user will filter out the dummies requests. That

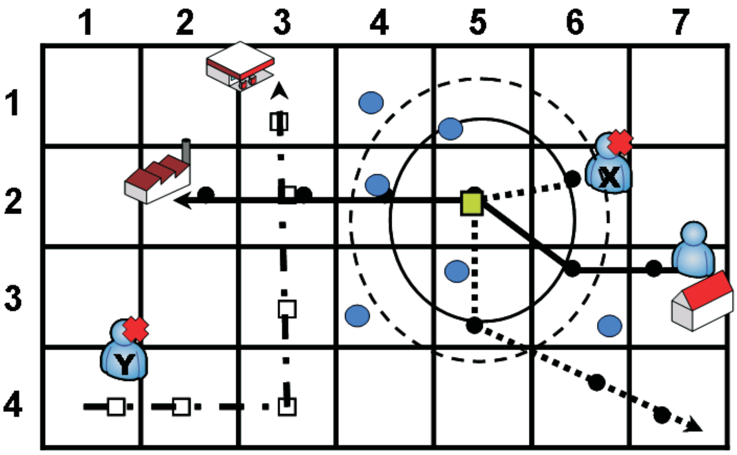


Figure 4.8: An example of knn query caching.

is undoubtedly a waste of resource. Caching dummy data for future use is possible, if select intersection points before that a user is likely to stay. Not only the intersection but also the area near the intersection could be cached. Adding a dummy trajectory intersects user's trajectory at different time slots in the generating dummy scheme. Consider Figure 4.8 as an example, where KNN queries are issued. Initially, we retrieve more than k data sources and let dummy X arrive the intersection (2,5) before the arrival of this user. Thus, the data returned for dummy X when X is used to arrive at intersection (2,5).

Based on the above concept, we employ *query cost* to evaluate the performance. The query cost QC_d is defined as:

$$QC_d = (n+1) * \sum_{i=1}^{m} SZ_i - ((n+1)I_{bt} + I_{ft} + I_{at}) * \sum_{j=1}^{k} SZ_j$$

The first term represents the query cost of dummy method. SZ_i and SZ_j mean the size of answer messages, n and m mean the total number of dummies and the total time slot respectively. The second term represents the saved query cost of cache. I_{bt} are the intersection points before the user arrive, I_{at} are the intersection points which dummy and user arrive at the same time and I_{ft} are the intersection points after the user arrive. Because I_{bt} are beneficial for the cache scheme, user and dummy needn't query when user arrive the I_{bt}. If we want to lower the *query cost*, one should increase the number of I_{bt}.

Dummies should arrive to the PL_i early, we can derive that $PL_d^{t1} = PL^{t2}$ and $t1 < t2$, where PL_d^{t1} is the dummy's location at time slot $t1$ and PL^{t2} is the user's location at time slot $t2$. Figure 4.8 shows an example of *query cost*, dummy X arrives the position (2,5) before user and there are two dummies and five time slots. Assume that the size of answer messages(SZ) are 10, it can be verified that QC_d=$(2 + 1) * 50 - (3 * 1 + 0 + 1) * 20 = 70$ in this example.

4.3 Generating Dummies with Multiple Trajectories

According to the research in [23], a user in general has multiple moving trajectories. Consider Figure 4.9(a) as an example, where a userhas four frequent trajectories and each has different probabilities. This user goes to his office with $A\%$, $B\%$ to the hospital, to the gas station with $C\%$ and the park with $D\%$. Every person has different static frequent trajectories and each trajectories has distinct probability. Our method should also guarantee the user profile if user is on his frequent trajectories. Each user has k moving trajectories with corresponding with probabilities $p_1, p_2, ...p_k$ and these trajectories are denoted as $\{(PL_i^1(p_1), PL_i^1(p_2), \ldots, PL_i^1(p_k)), (PL_i^2(p_1), PL_i^2(p_2), \ldots, PL_i^2(p_k)), \ldots, (PL_i^m(p_1), PL_i^m(p_2), \ldots, PL_i^m(p_k))\}$, where $(PL_i^j(p_k)$ is the location of mobile user U_i at the

jth time slot with the probability p_k. Notice that $\sum_{i=1}^{k} p_i$ may not be 100% because we only consider about frequent trajectories. Trajectories with smaller probabilities are not considered. We propose two schemes, namely, *Multiple Path Selection*(MPS) and *Multiple Path without Selection*($MPwS$). In the MPS scheme, we also present two types of dummy generation, named *Multiple Random Dummy Generation*($MRANDG$) and *Multiple Rotation Dummy Generation*($MROTDG$).

4.3.1 Multiple Path without Selection ($MPwS$)

Given user trajectories with probability $p_1, p_2, ...p_k$ and user profile, scheme $MPwS$ generates dummy trajectories for each trajectory. Suppose that each trajectories will generate n dummies, we will have $k * n$ dummy trajectories, where n is the number of dummies.

Although this method certainly guarantees for each trajectories for the privacy profile, the number of dummies is huge. As a result, the *query cost* will increase as well. Therefore, we proposed *Multiple Path with Selection*(MPS) that consider the probabilities of trajectories for dummy generation of each trajectories.

4.3.2 Multiple Random Dummy Generation ($MRANDG$)

In multiple random dummy method($MRANDG$), each dummy creates k dummy patterns with probability $p_1, p_2, ...p_k$, which is the same as user's trajectories. So it should generate $k * n$ dummy trajectories, where n is the number of dummies selects dummy trajectories with each probability. Take Figure 4.9 for an illustration, Figure 4.9(a) shows the original user's four trajectories with different probability(A%, B%, C%, D%) respectively. Using the random dummy algorithm to add one dummy in Figure 4.9(b), it is obviously to see that dummy X generates four

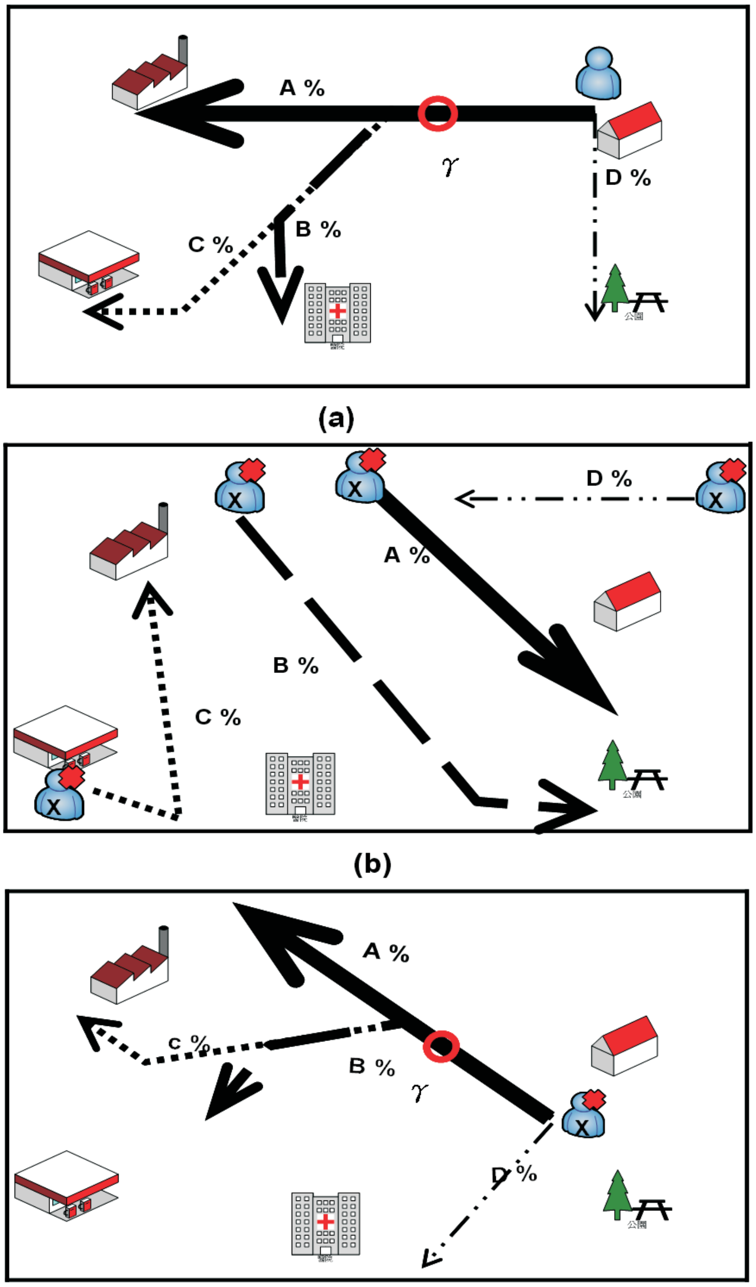


Figure 4.9: An example of multiple patterns.

dummy trajectories with the probability A%, B%, C%, D% respectively. Note that the privacy profile is not guaranteed, since the selection of dummy patterns is accordance with the probabilities. The other problem is user people's trajectories usually have partial similarity in that user's moving trajectories are also have some common sub-trajectories [20] [4]. In Figure 4.9(a), user's moving trajectories are also have some common sub-trajectories, but the dummy X's trajectories have no similar portions of the trajectories.

4.3.3 Multiple Rotation Dummy Generation ($MROTDG$)

Considering about that patterns usually has partial similarity, it is defined δ to measure the similarity of dummy trajectories and user trajectory. If $\delta = 0\%$, these two trajectories don't have any overlap, if $\delta = 100\%$, these two patterns are totally the same. Let B denote a user moving behavior set that we can represent $B(U_i) = \delta_1, \delta_2, ..., \delta_k$ where δ_i means the similarity of trajectory i with probability p_i. All the moving behavior δ_i are compare to the most frequent trajectory. Considering in rotation dummy generation, we try to make all the trajectories satisfy user's profile. Our key idea is that we rotate the whole trajectories with the same angle θ at the rotation point i to solve the selection problem. If a user is on the branch point, dummy will be able to select the suitable dummy trajectory. Figure 4.9(c) shows an example of selection rotation point as γ. The selection of the rotation point i and rotation angle θ are proposed as follows:

Possibility Based Method(PBM): Each dummy will generate k different dummy trajectories with different probability. These k dummy trajectories have its own solution space. We can divide it into two situation - rotation angle selection and rotation point selection. The key idea of this method is to find out the

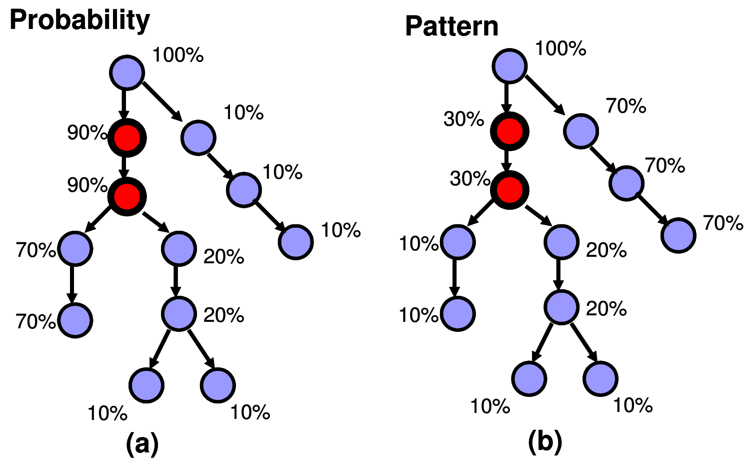


Figure 4.10: Two methods of select intersection point in multiple patterns.

most probability in the sub-trajectories. In other words, we want to find out the intersection i which can calculate the maximum of $\sum_{j=0}^{a} PL^i(p_j)$. a is the sum of the location in each trajectory probability. Every trajectory has its own solution space about θ and i, so we can find out the overlap of rotation angles θ easily. First at all, we find out the rotation point i, which makes $\sum_{j=0}^{a} PL^i(p_j)$ maximum. Depend on the rotation point, we find filter out some rotation angles in solution space. The remaining rotation angles are θ.

Figure 4.9 shows a multiple trajectories of a real user. The candidate of rotation point is in the whole trajectories. We sort the candidate positions in order of time and position are shown in Figure 4.10(a), which probabilities with $A = 70\%, B = 10\%, C = 10\%, D = 30\%$. First at all, we exclude the important place(i.e., home) and find out the highest probability to be the rotation point, in this example is the position with 90% probability. Then we use this constraint to find out the suitable rotation angle θ. This method can guarantee most quality of user's trajectory privacy. In the method, selection will not be a problem. User and dummy will have same sub-trajectories, so dummy also has the graph like 4.10(a) and know which trajectory should be chosen.

Trajectory Based Method (TBM): The approach is different from the possibility based method in rotation point selection. Possibility based method is based on selection highest probability location for the intersection. Trajectory based method is based on the number of trajectories. The intersection i selects the largest number of trajectories pass through.

Assume that $A = 10\%, B = 10\%, C = 10\%, D = 70\%$, and we exclude the important place and find out the location which has most trajectories pass through. In Figure 4.10(b), the position with 30% probability will be chosen, not the position with 70% probability. This method will guarantee most trajectories but not highest probability.

Chapter 5

Performance Study

In this section, we evaluate the performance of our proposed schemes and conduct experiments to evaluate our dummy generation scheme. We first describe the simulation model in Section 5.1 and we will show the experimental results in Section 5.2.

5.1 Simulation Model

In our simulation, we use two kinds of trajectory sets, one is the real data - INFATI [17] data set and the other one is our trajectory generator. The INFATI data derive from the INFATI Project, which was day to day movements of several private cars on the road network of Aalborg. We select one car's moving log for the real data set. In order to discuss conveniently. We use integer as unit time instance and set the whole time period from 0 to 50. The entire area within which the car has been moving is divided into grid of size 100*100. Because the data set is sampling every second when car drove, we assume the moving object send his request every k seconds. k is default to be 30.

Our trajectory generater is divided the space into 50*50 grid cells which size is 10m*10m. Assume that the number of time slots is 20. Moreover, We simply assume that there exists k moving trajectories for each user with probability $p_1, p_2, ...p_k$ and partial similarity $\delta_1, \delta_2, ..., \delta_k$, (i.e., the pattern has a starting point and a destination point). Then, those grid cells between the starting point and the destination are selected based on the nature of movements, (i.e., the next move is a neighboring cell of the current location). Three movement types are implemented, the horizontal movement, the vertical movement, and both. To emphasize the privacy threat of long-term observation, we implemented the prior work in [18] [19] as scheme *dummy*. Suppose that adversaries are able to collect the query log in which the movements of dummies and true users are recorded. Adversaries may explore data mining techniques [23] to discover movement patterns of users.

To evaluate the simulation result, performance metrics are *Number of dummies*, *Pattern Exposure Rate*, *Correct Rate* and *Query Cost* for the evaluation.

Number of dummies:*Number of dummies* is the number of dummies needed to satisfy the privacy profile

Pattern Exposure Rate:*Pattern Exposure Rate* can be represented as the formula *Pattern Exposure Rate*= $\frac{\text{The trajectory exposed by the attacker}}{\text{The total time slots of trajectory}}$

Correct Rate:*Correct Rate* is that the proportion of dummy selected the right trajectory.

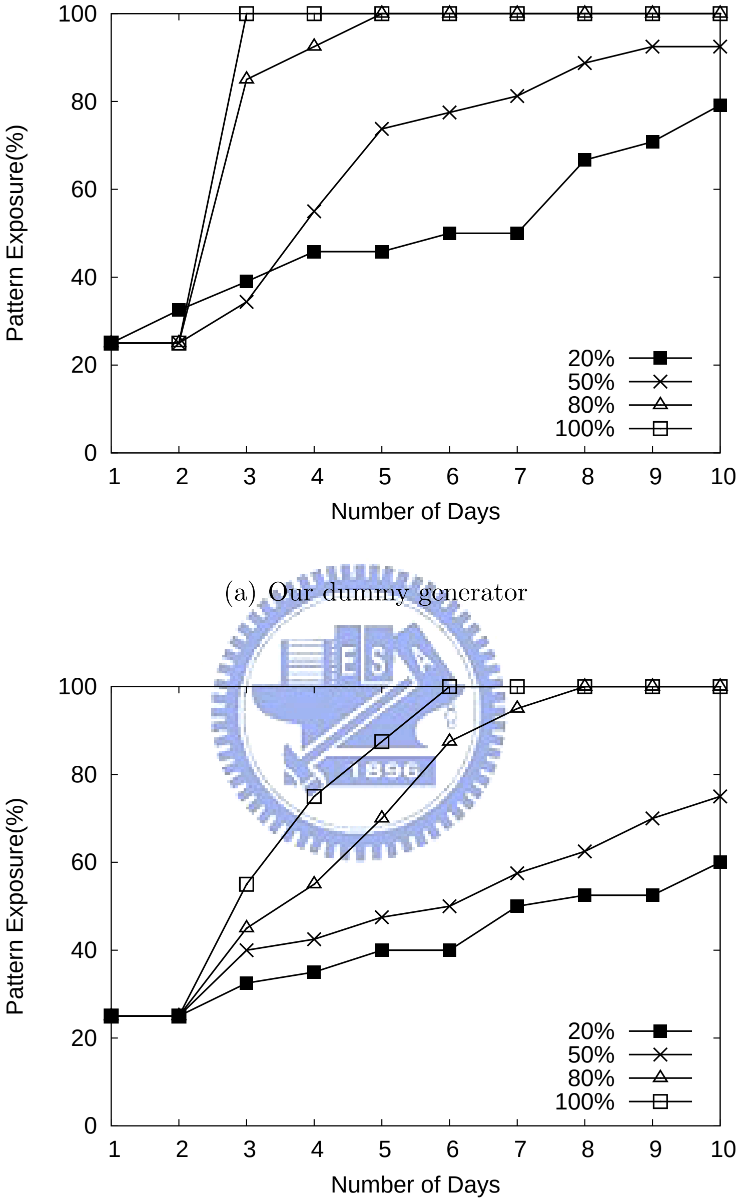
Query Cost:*Query Cost* is the communication cost defined in Section 4.2.2.

5.2 Experimental Results

We will discuss the settings and prove that long term observation will expose user's trajectory and compare with the the Kido's dummy [18] under different settings in Section 5.2.1. In Section 5.2.2, we will compare our method in several aspects the number of dummy and the saving of resource. Finally, we present the correct rate of different approaches in multiple trajectories.

5.2.1 Comparison to Dummy

We use the INFATI data set for the real data. In first experiment, we will show that how many days of query logs if the attacker collects will exposure the whole trajectory. We pick up one car which data was collected during December 2000 and January 2001 and our trajectory generator for the experiment. We control the coefficient of variation of query's sampling. Assume that user's query is discrete uniform distribution, then compare with different query probability is 20%, 50%, 80%, 100%. We assume the adversaries user the sequential pattern mining and the support is set to two and the confidence is set to 50%. Figure 5.1 shows the experimental results. The needed days is shown on the X-axis and the *pattern exposure rate* is on the Y-axis. *pattern exposure rate* represents the total pattern guessed by the adversaries. If the *pattern exposure rate* is closed to 100% that means the attacker has high probability to hit the real trajectory. We can see the result in Figure 5.1 that user's moving trajectory can be exposed as long as the attacker get enough moving logs. If we assume the attacker set the minimum support 2 and the minimum confidence 50%, we can see the result in Figure 5.1(a)and(b) that the lower the query sample the more the attacker should collect. If the query sampling



(a) Our dummy generator

(b) INFATI data set

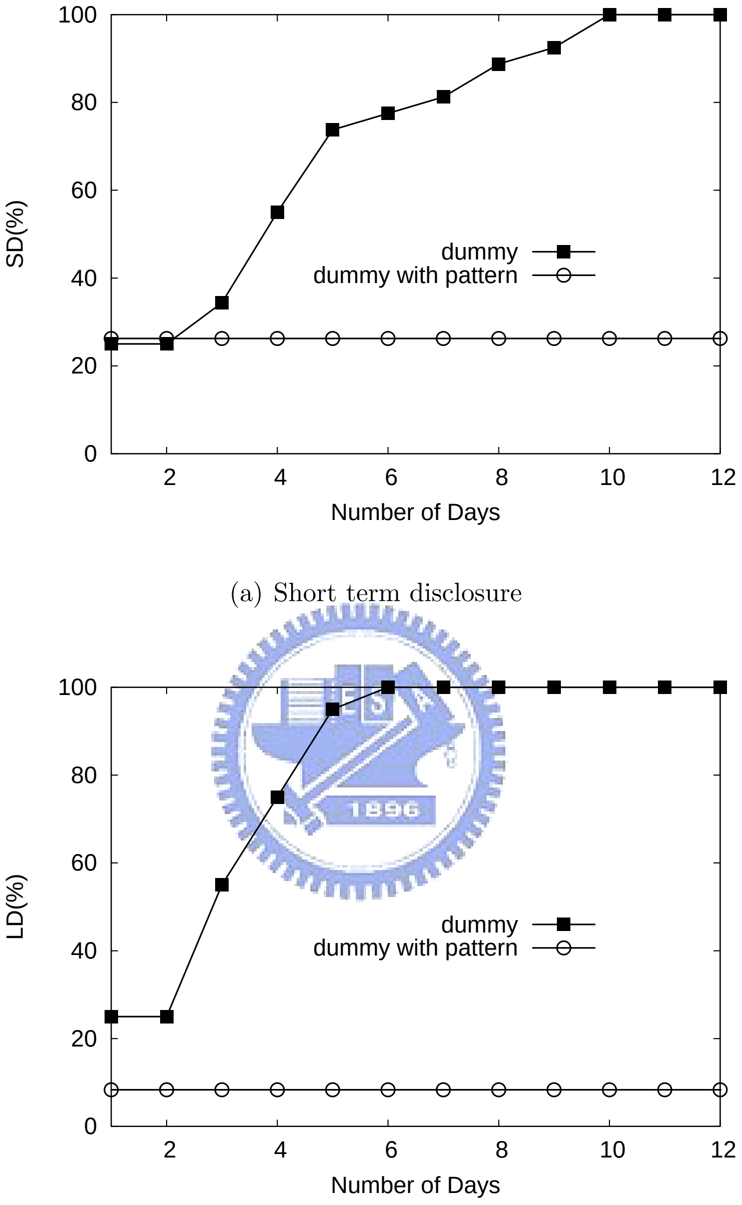
Figure 5.1: The needed day of exposure a user's whole trajectory.

arrives to 100%, the attacker only need 3 days and 6 days moving logs to exposure the user's whole trajectory in our pattern generator and INFATI data, respectively.

We now investigate the impact of movement patterns. Suppose a privacy profile is set to $SD = 20\%, LD = 10\%$, and $dst = 2.8$. We compare our rotation pattern scheme with the dummy scheme. Figure 5.2 shows the experimental result. In Figure 5.2(a), it can be seen that when the amount of data collected increases with the time, both SD and LD of the dummy scheme increase. This agrees with our claim that long-term privacy threat exists if dummies do not follow long-term, consistent movement patterns. Once collected a sufficient amount of data, the true user trajectory is completely exposed, that results in 100% disclosure in term of SD and LD. On the other hand, our scheme is able to satisfy the specified disclosures (i.e., $SD = 20\%, LD = 10\%$), showing that generating dummies with patterns could prevent both short-term and long-term location privacy.

5.2.2 Sensitive Analysis of Dummy Generation Scheme

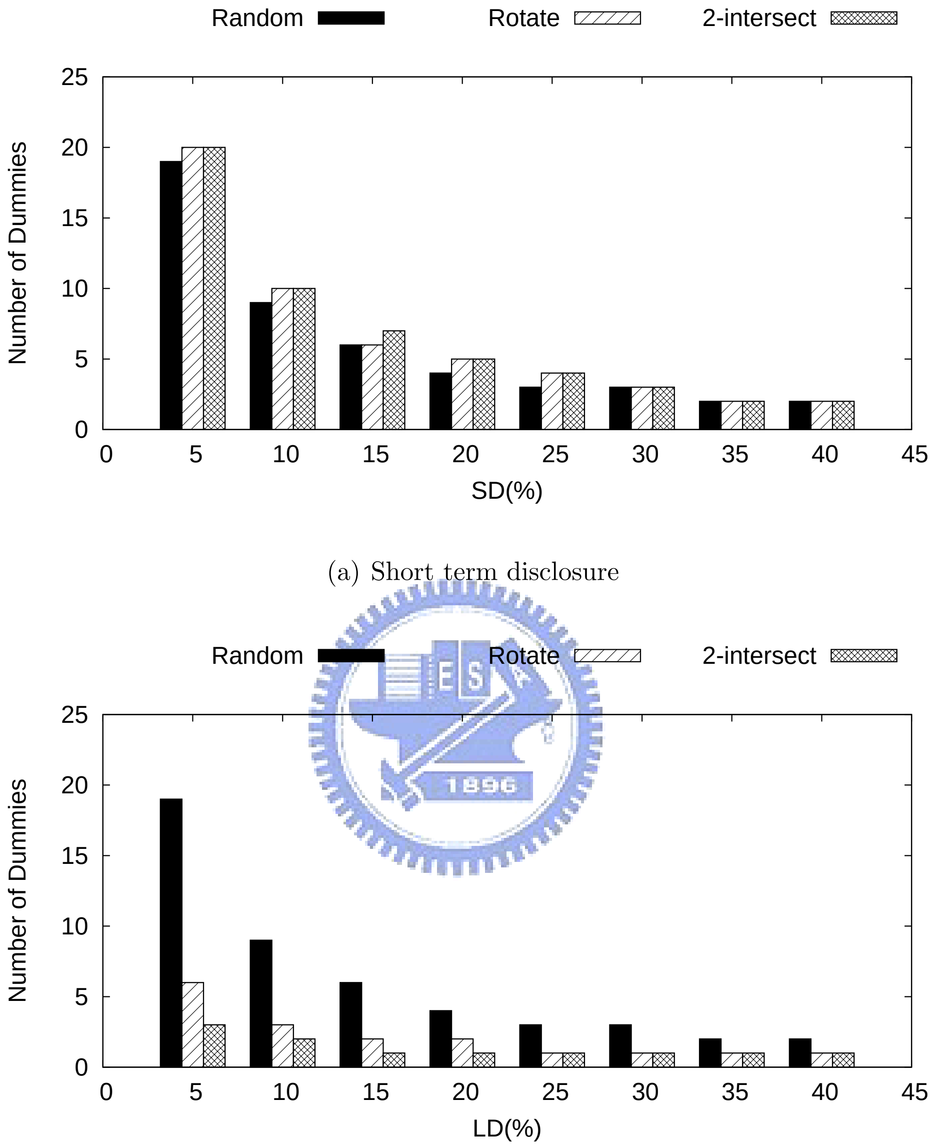
Next, the performance of our two proposed schemes is compared. We also use the real data for this experiment. The proposed random pattern scheme, rotation pattern scheme and k-intersect scheme are denoted as Random, Rotate and k-intersect, respectively. As mentioned earlier, when the privacy requirements are not satisfied, additional dummies are included. However, a larger number of dummies increases query message lengths, leading to a considerable cost in communication and client processing. Thus, one should use as minimum number of dummies as possible to satisfy user privacy profiles or use the cache scheme. The performance of schemes Random, Rotate and 2-intersect with the value of SD varied is shown in Figure 5.3(a), where $LD = 50\%$ and $dst = 2.8$. Since SD is related to short-



(a) Short term disclosure

(b) Long term disclosure

Figure 5.2: The performance comparison of dummy without moving patterns and dummy with moving patterns.



(a) Short term disclosure

(b) Long term disclosure

Figure 5.3: The performance of scheme Random, Rotate and k-intersect with LD varied.

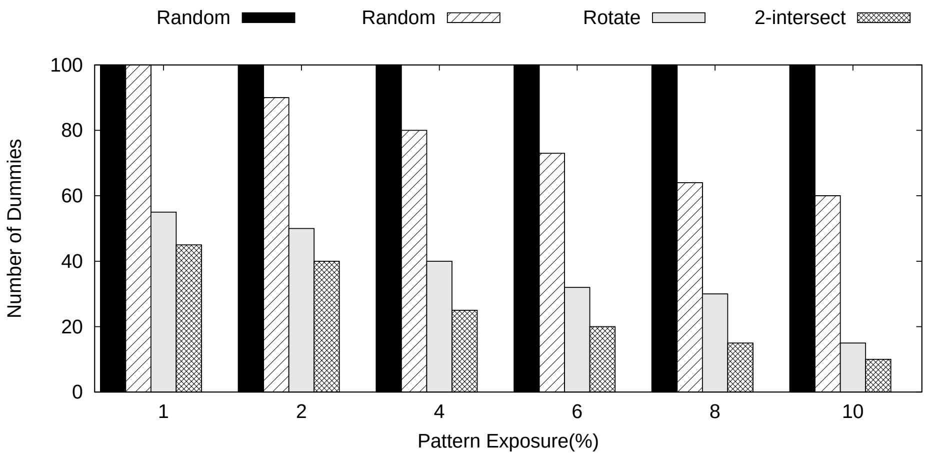
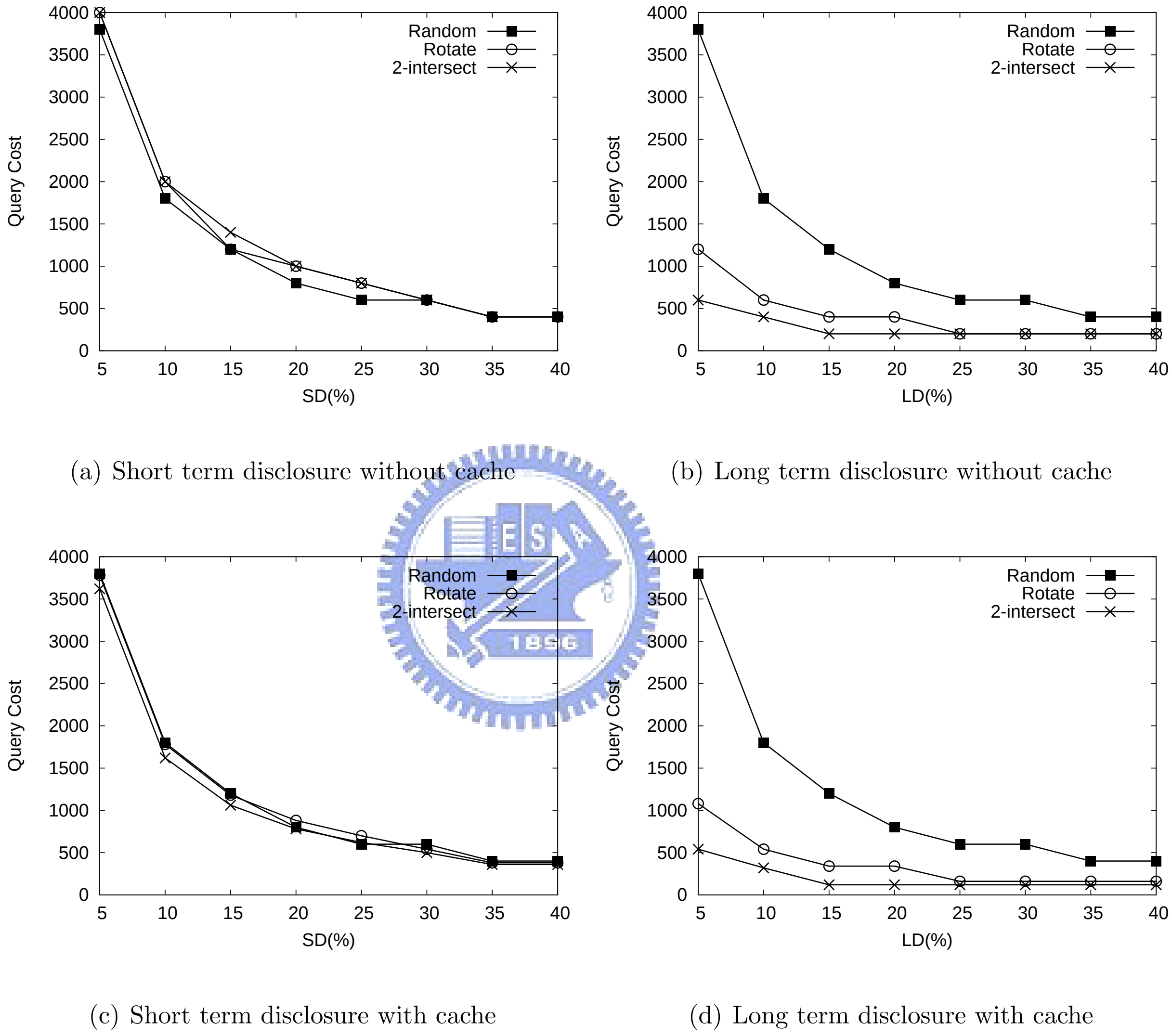


Figure 5.4: The exposure rate of scheme Dummy, Random, Rotate and k-intersect with the number of dummies varied.

term disclosure, these scheme Random, Rotate and 2-intersect use almost the same number of dummies to meet the requirement of SD. Furthermore, an experiment of varying LD is conducted with $SD = 50\%$ and $dst = 2.8$. Figure 5.3(b) shows the experimental result. It can be seen that if we make an intersection between two trajectories that may uses a smaller number of dummies than scheme Random. By intersecting trajectories, scheme Rotate and k-intersect are able to increase the number of possible trajectories. Hence, these two scheme could generate smaller number of dummies to meet the privacy requirement.

One of the benefits in generating intersection dummy pattern is that even if some position exposed by the attacker, the whole trajectory is still not exposed. When some place is disclosed, it also means the positions nearby these place are exposure. We can use the intersection dummy to solve this problem. See the result in Figure 5.4, we can clearly see that if some position exposed, Dummy will expose the whole pattern. Provided that dummy has intersections, we can clearly find that the *Pattern Exposure Rate* disclose the least percentage in the user's whole



(a) Short term disclosure without cache

(b) Long term disclosure without cache

(c) Short term disclosure with cache

(d) Long term disclosure with cache

Figure 5.5: The query cost of scheme Random, Rotate and k-intersect.

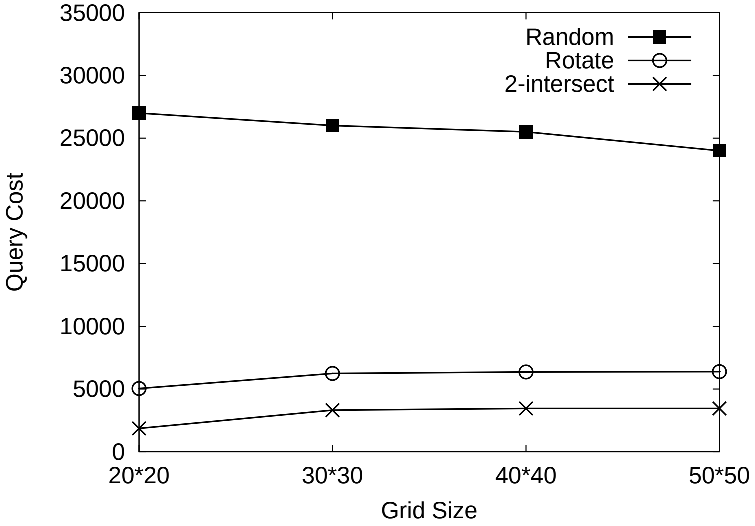


Figure 5.6: The affect of query cost with the grid size

pattern. The more intersection dummy has, the least pattern exposed.

Another issue is the waste of resource. Considering in the communication cost of our method, we proposed a evaluation function - *Query Cost* and compare our methods in Figure 5.5. We set the answer message size - SZ to be 10. We compare into two situation - with cache and without cache to see the result. Figure 5.5(a) and (b) represent dummy without caching and Figure 5.5(c) and (d) represent dummy with caching. In Figure 5.5, we observe that if user want to keep highly privacy, the query cost will be higher. All of our methods only have slightly difference in Figure 5.5(a) and (c). However, in Figure 5.5(b) and (d), both intersection dummy schemes have a better performance than random scheme in terms of LD, especially in 2-intersect dummy. That is because the intersections can reduce the total number of dummies. If we cache the data for re-use, we can compare 5.5(a)(c) and Figure 5.5(b)(d) and get the calculation that cache can bring a little benefit depending on the number of intersections. If dummy has more intersections, the cache scheme can re-use more.

5.2.3 Experimental Results of Dummy Generation Schemes for Multiple Trajectories

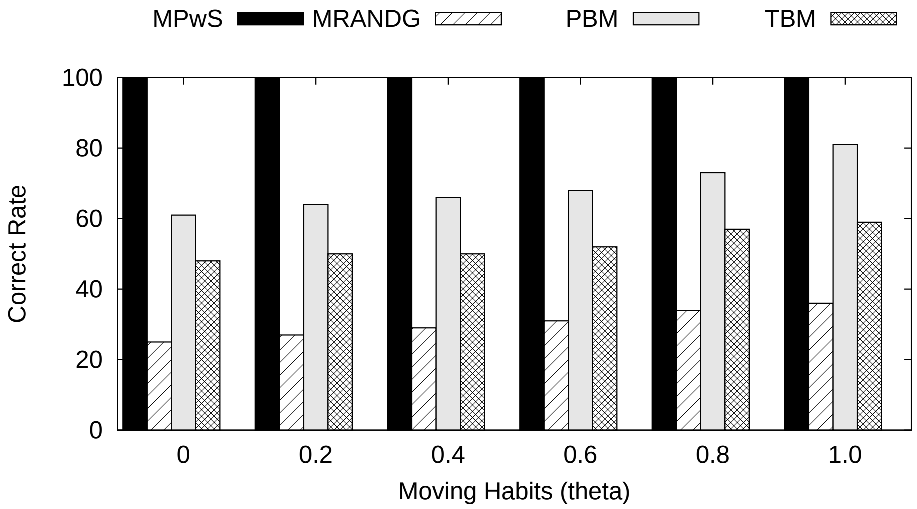
In this experiment, we discuss about the situation that user has more than one frequent patterns. We investigate the influence of the different frequent pattern probability $p_1, p_2..., p_k$ and the partial similarity $\delta_1, \delta_2, ...\delta_k$. We propose *correct rate* that means the probability of selecting accurate dummy trajectory.

First of all, we discuss the different frequent pattern probability. Assume the user has several frequent trajectories with probability which obey zipf-like [24] distribution. Use our trajectory generator to generate several trajectories with different probability, every trajectory's probability describe as follows:

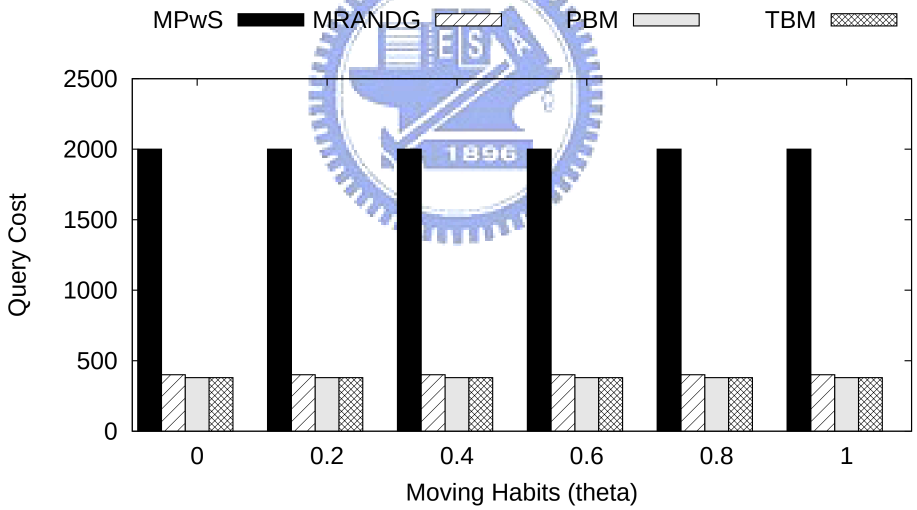
$$P_{x=k} = \begin{cases} \dfrac{(\frac{1}{k})^{\theta}}{\sum\limits_{i=1}^{a}(\frac{1}{i})^{\theta}} & k = 1...a \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

which means the different user's moving habits. We can modify the value of θ and a to control the bias of user's moving behavior. The value of θ means the exponent. If $theta = 1$, user will has a higher probability to move on frequent trajectory. In the other hand, if $theta = 0$, zipf-like distribution will obey an uniform-distribution that user will has the same probability on each trajectory. The value of a represents the number of user's frequent trajectories. We assume these frequent trajectories has partial similarity $\delta = 50\%$ to the highest possible trajectory. Compared with the four methods($MPwS$, $MRANDG$, PBM, TBM) we proposed before and discuss the two evaluation of *Correct Rate* and *Query Cost*.

First of all, we discuss the influence of δ. The amount of user's frequent trajectories is set to 5 and the size of package SZ is set to 10. We can observe the

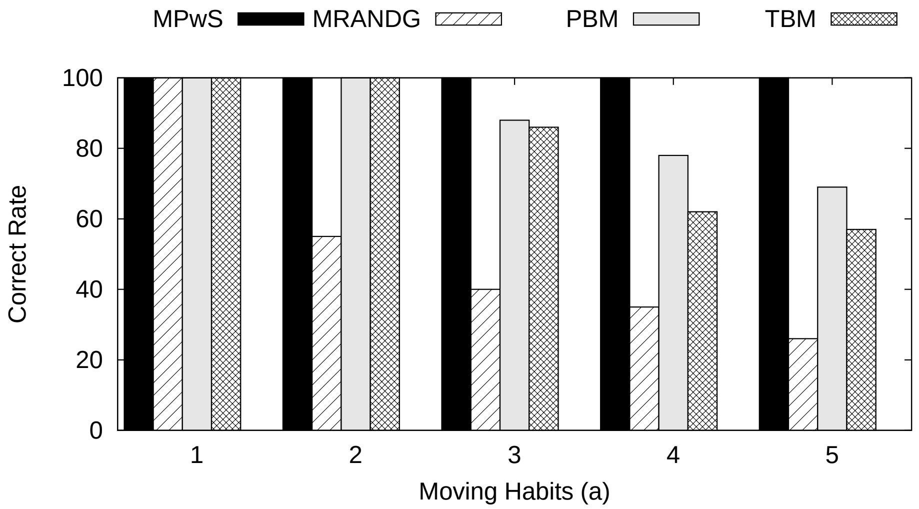


(a) Correct Rate

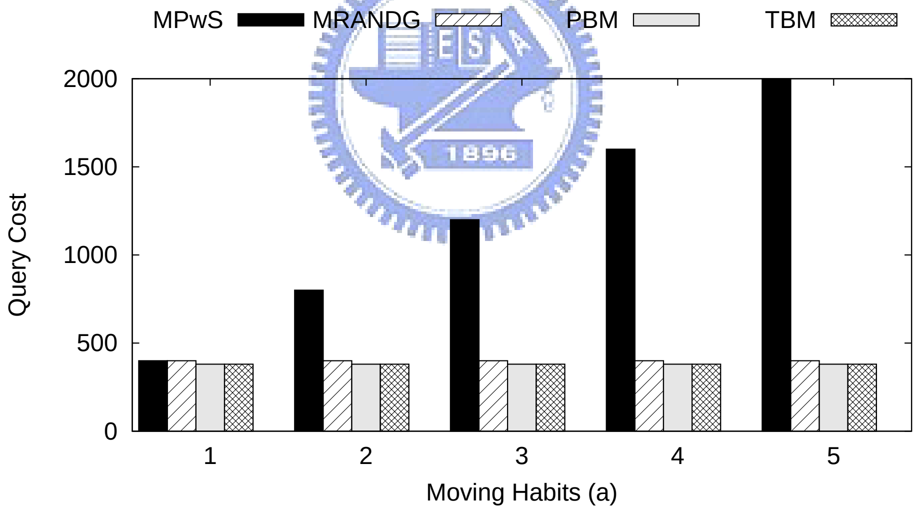


(b) Query Cost

Figure 5.7: The query cost and correct rate in different probability of multiple path(theta).

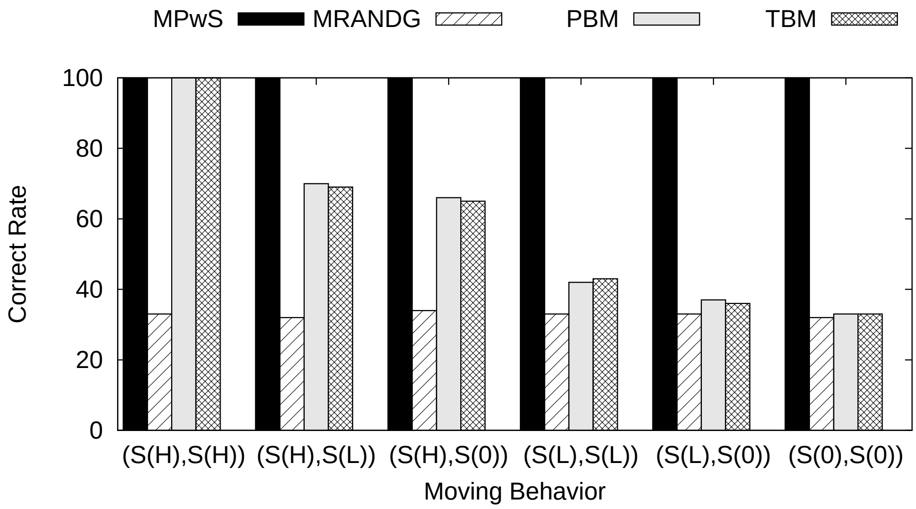


(a) Correct Rate

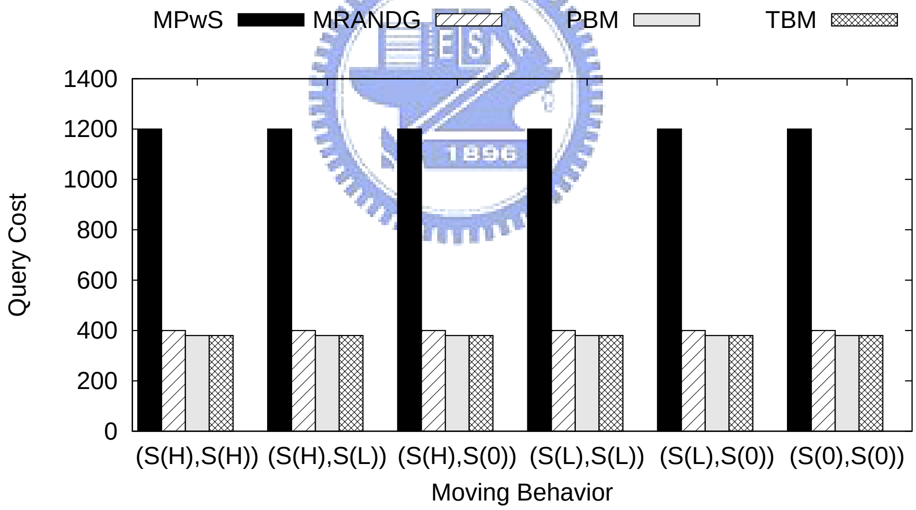


(b) Query Cost

Figure 5.8: The query cost and correct rate in different probability of multiple path(data object).



(a) Correct Rate



(b) Query Cost

Figure 5.9: The Query Cost and Correct Rate in different similarity of multiple path(theta).

result in Figure 5.7, $MPwS$ can keep the quality of privacy to the 100% but the *Query Cost* is also too high to be apply. The selection method MPS proposed for lower the *Query Cost* and try to keep the quality of privacy as high as possible. PBM has a better performance in all the selection methods, especially in people whose highest possible trajectory are more frequent trajectory than other trajectories. Although it cannot guarantee the 100% user privacy, but the *Query Cost* is less than the method $MPwS$. If user has many uniform frequent patterns, the performance of all the method are about the same.

Then we debate the different number of frequent trajectories(a). Considering about the real situation - user has a frequent trajectory, we set the value of δ to be 1 and the size of package SZ is also set to 10. We can see the result clearly in Figure 5.8, the more patterns will cause the higher *Query Cost* in method $MPwS$. The selection methods can guarantee the higher quality of privacy, if user's frequent trajectories are less and the *Query Cost* are still lower than the method $MPwS$. Especially, the method TBM can ensure high quality of privacy. Method PBM are similar to the method TBM when the amount of patterns are less because it consider about the most patterns not the most possibly. When patterns become less, these two method are always the same meanings.

Finally, we consider the partial similarity. We divide these trajectories into the following situations: highly partial similarity $S(H)$, low partial similarity $S(L)$, no partial similarity $S(0)$. $S(H)$ means the other trajectories have about 90% to the most frequent trajectory, $S(L)$ have 30% and $S(0)$ have no similarity. We generate three trajectories with probability obey by uniform distribution(33%). The simulation result are shown in Figure 5.9. If each trajectory has highly similarity, the performance of TBM and PBM will better than others and it can guarantee almost 100% quality of privacy. If the similarity become lower or no similarity, these methods' performance will tend to change into the same. Using the method $MROTDG$, *Query Cost* will become lower. If each trajectory has highly partial similarity, it is more suitable to using $MROTDG$.

Chapter 6

Conclusions

We observed that existing works using dummies to protect location privacy are still exposed to privacy threat in a long run. Explicitly, by exploring data mining techniques, adversaries may be able to determine user movement patterns, thereby invading user location privacy. To deal with this problem, we proposed two schemes to derive dummy trajectories, they are random scheme and intersection scheme. Specifically, random pattern scheme randomly generates dummies with consistent movement patterns, while the rotation pattern and k-intersect explore the idea of creating intersections among moving trajectories. We also consider about lower the communication cost and the problem of multiple path. Our preliminary performance study shows that by generating dummies with movement patterns, our proposal outperforms the existing dummy-based scheme for protecting trajectory and locations of mobile users.

Bibliography

[1] R. Agrawal and R. Srikant. Mining sequential patterns. *Proc. of the 11th IEEE International Conference on Data Engineering (ICDE)*, 00:3–14, 1995.

[2] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[3] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *Proc. of the 2nd IEEE Workshop on Pervasive Computing and Communication Security (PerSec)*, 2004.

[4] H. Cao, N. Mamoulis, and D. W. Cheung. Mining Frequent Spatio-Temporal Sequential Patterns. In *Proc. of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 82–89, 2005.

[5] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[6] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proc. of the 6th internation workshop on Privacy Enhancing Technologies (PET)*, pages 393–412, 2006.

[7] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In *Proc. of the 14th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS)*, 2006.

[8] A. Friday. A Lightweight Approach to Managing Privacy in Location-Based Services. In *Proc. of the Equator Annual Conference (EAC)*, 2002.

[9] B. Gedik and L. Liu. A Customizable k-Anonymity Model for Protecting Location Privacy. In *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2005.

[10] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 620–629, 2005.

[11] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 31–42, 2003.

[12] M. Gruteser and D. Grunwald. A methodological assessment of location privacy risks in wireless hotspot networks. In *Proc. of the First International Conference on Security in Pervasive Computing (SPC)*, volume 2802, pages 10–24, 2003.

[13] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitative analysis. *ACM Mobile Networks and Applications (MONET)*, 10(3):315–325, 2005.

[14] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Academic Press, 2001.

[15] J. I. Hong and J. A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *Proc. of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 177–189, 2004.

[16] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki. Enhancing wireless location privacy using silent period. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1187–1192, 2005.

[17] C. Jensen, H. Lahrmann, S. Pakalnis, and J. Runge. The infati data, 2005.

[18] H. Kido, Y. Yanagisawa, and T. Satoh. An Anonymous Communication Technique using Dummies for Location-based Services. In *Proc. of the Second International Conference on Pervasive Services (ICPS)*, pages 88–97, 2005.

[19] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of Location Privacy using Dummies for Location-based Services. In *Proc. of the 21th IEEE International Conference on Data Engineering Workshop (ICDEW)*, page 1248, 2005.

[20] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *Proc. of the 26th ACM Conference on Management of Data (SIGMOD)*, pages 593–604, 2007.

[21] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 763 – 774, 2006.

[22] W.-C. Peng and M.-S. Chen. Developing data allocation schemes by incremental mining of user moving patterns in a mobile computing system. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15(1):70–85, 2003.

[23] W.-C. Peng, Y.-Z. Ko, and W.-C. Lee. On mining moving patterns for object tracking sensor networks. In *Proc. of the 7th International Conference on Mobile Data Management (MDM)*, pages 41–44, 2006.

[24] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley (Reading MA), 1949.