# Query m-Invariance: Preventing Query Disclosures in Continuous Location-Based Services

Rinku Dewri, Indrakshi Ray, Indrajit Ray and Darrell Whitley
*Department of Computer Science*
*Colorado State University, Fort Collins, CO, USA*
{*rinku,iray,indrajit,whitley*}*@cs.colostate.edu*

***Abstract*—Location obfuscation using cloaking regions preserves location anonymity by hiding the true user among a set of other equally likely users. Furthermore, a cloaking region should also guarantee that the type of queries issued by users within the region are mutually diverse enough. The first requirement is fulfilled by satisfying location $k$-anonymity while the second one is ensured by satisfying query $\ell$-diversity. However, these two models are not sufficient to prevent the association of queries to users when the service depends on continuous location updates. Successive cloaking regions for a user may be $k$-anonymous and query $\ell$-diverse but still be prone to correlation attacks. In this paper, we provide a formal analysis of the privacy risks involved in a continuous location-based service, and show how continuous queries can invalidate the privacy guarantees provided by $k$-anonymity and $\ell$-diversity. Drawing upon the principle of $m$-invariance in database privacy, we show how query $m$-invariance can provide location and query privacy in continuous services.**

***Keywords*-query privacy, continuous location-based services**

## I. Introduction

Technological advances in location tracking and its growing embedment in mobile devices have opened up a new spectrum of on-demand services. These services deliver customized information based on the location of a mobile object. A location-based service (LBS) may simply provide information on the nearest gas station, perform targeted marketing by location-based advertising, or enhance emergency response services, among others. Application domains are potentially endless with location-tracking technology. However, a serious concern surrounding their acceptance is the potential usage of the location data to infer sensitive personal information on the mobile users.

Privacy in location-based services has been studied from two different perspectives – *location anonymity* and *query privacy*. Location anonymity is related to the disclosure of exact locations that a user has visited. This knowledge can in turn reveal personal lifestyles, places of frequent visits, or even the medical problems of the involved user. With access to exact location data, sender anonymity can be violated without the capability to track a mobile user. Location obfuscation is therefore one of the widely researched approaches to safeguard location anonymity. This technique guarantees that the location data received at the LBS provider can be associated back to more than one object – to at least $k$ objects under the *location $k$-anonymity* model [1]. For this, a *cloaking region* is communicated to the service provider instead of the actual location. A $k$-anonymous cloaking region contains at least $k-1$ other mobile users besides the service user.

Query privacy is related to the disclosure of sensitive information in the query itself and its association to a user. Consider a marketing agency such as CellFire® that delivers mobile coupons to users based on their location and category of interest. The retail partners sponsoring such an agency are spread out across multiple categories, ranging from apparels, groceries, automotives, entertainment, electronics to insurance, telecommunication, marketing and fitness. Each category in itself can be sub-divided; apparels, for example, can be divided into men's, women's or children's. Users therefore use *service attribute* identifiers that specify their interest category. More often than not, a user's service attribute value is considered sensitive since it directly reveals personal preferences (or requirements) of the user. Addressing the sensitivity is more important in a service like Google™ Adwords that can perform location-based marketing on virtually any area of interest. Query privacy is therefore an essential requirement. It has a more direct impact on user privacy than location anonymity.

### A. Motivation

Preservation of query privacy in a LBS is similar to protection against attribute disclosures in data privacy. A typical principle used in this context is *query $\ell$-diversity* [2]. A cloaking region conforming to query $\ell$-diversity contains users with at least $\ell$ "well-represented" service attribute values. One way of enforcing the principle is to ascertain that there are users with at least $\ell$ distinct interest categories. Henceforth, any reference to query $\ell$-diversity implies this particular enforcement. Query $\ell$-diversity ensures that a user cannot be linked to less than $\ell$ distinct service attribute values, thereby preventing homogeneity attacks [3]. However, this approach is not sufficient to prevent query disclosures in a *continuous location-based service*.

A continuous LBS is one to which users issue recurrent queries over a period of time. Each query is accompanied

978-0-7695-4048-1/10 $26.00 © 2010 IEEE
DOI 10.1109/MDM.2010.52

| time | user set | service attribute values |
|---|---|---|
| $t_1$ | $\{U_1, U_2, U_3\}$ | $\{a, b, c\}$ |
| $t_2$ | $\{U_1, U_2, U_4\}$ | $\{a, b, d\}$ |
| $t_3$ | $\{U_1, U_3, U_4\}$ | $\{a, c, d\}$ |

Table I: Successive query 3-diverse cloaking regions.

by current location information in order to obtain updated results. An example is a mobile user cruising through an urban locality and repeatedly using a marketing service to find the real estates on sale in the neighborhood. In an attempt to maintain privacy, the continuous LBS in this case receives a sequence of cloaking regions corresponding to the recurrent queries. Let us assume that the set of users inside the cloaking regions and their service attribute values are as shown in Table I. All cloaking regions generated for the involved user are query 3-diverse and location 3-anonymous. However, given the information that the three cloaking regions are generated for the same user repeatedly inquiring about a particular category of interest, it is evident that the attribute value of interest is '$a$'. Further, $U_1$ being the only user common in all the cloaking regions, an adversary infers that $U_1$ has an interest in the category '$a$'. This form of disclosure occurs because existing models providing query privacy do not consider the possibility of correlating consecutive sets of service attribute values generated during the recurrent use of a LBS. New techniques are therefore required to ensure that users of continuous location-based services are well protected from threats originating from query disclosures.

### B. Related Work

While significant research has gone into algorithms that enforce location anonymity [1], [4], [5], [6], very few of them address the problem in the context of a continuous LBS. Gruteser and Liu specifically investigate privacy issues in continuous LBS [7]. They argue that privacy in continuous LBS applications can be situation dependent, hence pressing the requirement for sensitive and insensitive areas. Hoh and Gruteser propose a perturbation algorithm to cross paths of users (by exchanging their pseudonyms) when they are close to each other [8]. However, these approaches rely on the exchange of exact location information with the LBS. Bettini et al. first introduced *historical $k$-anonymity* as an extension of $k$-anonymity to a continuous LBS [9]. They propose a spatio-temporal generalization algorithm to compute cloaking regions that always contain at least $k$ fixed users. Xu and Cai propose an information theoretic measure of anonymity in continuous LBS [10]. They define a *$k$-anonymity area* as the cloaking region whose entropy is at least $k$. However, the algorithm is prone to *inversion attacks* where an adversary uses knowledge of the anonymizing algorithm to breach privacy. The most recent of algorithms to enforce historical $k$-anonymity is *ProvidentHider* [11]. However, none of these algorithms consider query privacy.

Chow and Mokbel argue that spatial cloaking algorithms should satisfy the *$k$-sharing* and *memorization* properties to be robust against service attribute associations [12]. Query privacy is preserved by ensuring that more than one user in the cloaked region is interested in the same service attribute value as the issuer. We later refer to this as *many-to-one* queries. Given the restriction that the cloaking region must memorize and maintain a fixed set of users, the size of the induced cloaking region becomes an issue with this technique. Riboni et al. argue that an adversary may derive an association between a user and a service attribute value based on the distribution of service attribute values in the cloaking regions generated for the user [13]. Therefore, they propose generalizing service attribute values so that the distance between the distribution of service attribute values in cloaking regions for the user and that in regions generated for other users is below a threshold. Besides the fact that generalizing service attributes adversely affects service quality, it is also not clear if performing such generalizations can prevent disclosures emerging from correlations in consecutive cloaking regions. The *$t$-closeness* model [14] on which their algorithm is based upon is itself known to be sensitive to the distance metric.

### C. Contributions

This paper presents the first formal analysis of privacy attacks leading to query disclosures in a continuous LBS. We explicitly characterize the privacy threats under consideration and state the background knowledge required to execute the underlying attacks. We model the attacks that can lead to query disclosures and formally show how a technique such as query $\ell$-diversity fails to provide query privacy in a continuous LBS. While the threats analyzed here are new in the context of location-based services, similar problems have been explored for privacy protection during the re-publication of dynamic microdata. The principal of *$m$-invariance* [15] is of particular interest here because of the similarity in privacy issues it helps resolve and those present in a continuous LBS. Drawing upon the privacy guarantees of $m$-invariance, we formulate the principle of *query $m$-invariance* and show how it can be used to control the amount of risk present in the use of a continuous LBS. We further propose a cloaking algorithm to efficiently enforce the principle. The algorithm uses a partitioning scheme of the query $m$-invariant user set so that service quality is not severely affected due to the privacy requirements. We supplement all analysis with extensive experimental validation.

The remainder of the paper is organized as follows. Section II presents the system architecture and highlights the requirement for query $m$-invariance. The cloaking algorithm is presented in Section III. Section IV details the

experimental setup and results from the comparative study. Finally, Section V concludes the paper.

## II. Preventing Query Disclosures

Formal evaluation of privacy attacks and preservation techniques is difficult without explicitly stating the extent of an adversary's knowledge. Earlier studies have identified two attack categories – *identity inferencing* (association of a user with the location(s) it has visited) and *query association* (inference of the sensitive attribute(s) involved in a user's request).

The extent of success while executing attacks in these categories is decided by the adversary's background knowledge. In this study, we assume that the adversary's background knowledge is in terms of location information of one or more users. Based on the availability of location information, we categorize the adversaries into two types.

*(i) Location-unaware adversaries:* This type of adversary does not posses knowledge of exact user locations. However, identity inferencing by such adversaries is possible when the revealed location data corresponds to a private address (restricted space identification) or can be associated to a user based on observed evidence (observation identification). If the location data is from a continuous LBS, then trajectories can also be linked to a user. Location $k$-anonymity prevents such inferencing by cloaking the exact location data inside a bounding box containing at least $k$ users. However, a $k$-anonymous cloaking region can implicitly reveal service attributes if all users in it specify the same (or similar) values. Query $\ell$-diversity prevents such attacks by ensuring that a cloaking region contains users with at least $\ell$ distinct attribute values.

*(ii) Location-aware adversaries:* This type of adversary has exact location information on one or more users, and possibly at multiple time instances. User identities are therefore assumed to be known to the adversary. Hence, exact location data may be communicated to the LBS. However, location-aware adversaries cannot infer service attributes from the location knowledge as long as every location communicated to the LBS is query $\ell$-diverse. In other words, a service request should involve a set of $\ell$ distinct attribute values (one of which is the real one) for every location update. Note that one cannot dismiss the absence of location-unaware adversaries in a given setting. Hence, cloaking regions are still used instead of exact locations.

A continuous LBS introduces other threats in the presence of location-aware adversaries. Given that both types of adversaries may be present, a LBS may adopt one of the following two methods to prevent query association.

*(a) Many-to-one queries:* In this method, a $k$-anonymous cloaking region communicated to the LBS is associated with a single service attribute (the one belonging to the actual user). Therefore, there are at least $k$ potential users who may be the owner of the service attribute. However, if only one user is common across all the cloaking regions, then the attribute value must be associated with that user. Therefore, a stronger requirement, often called historical $k$-anonymity [9], is enforced where every cloaking region must invariably contain a set of $k$ fixed users. However, historical $k$-anonymity can lead to large cloaking regions if the invariant users move away from each other over time.

*(b) Many-to-many queries:* In this method, a cloaking region is communicated to the LBS with a set of service attribute values (ones belonging to the users inside the region). Query association is prevented here by enforcing query $\ell$-diversity in the set of attribute values. However, as highlighted in Section I-A, query $\ell$-diversity is not sufficient in a continuous LBS.

Our focus in this paper is in the second strategy of prevention. We shall discuss the system architecture in accordance with this strategy and then show how query $m$-invariance eliminates the issues with query $\ell$-diversity in a continuous LBS.

### A. System architecture

Fig. 1 depicts our system consisting of interactions between three layers – (i) *mobile users*, (ii) a *trusted anonymity server*, and (iii) a *continuous LBS provider*. The trusted anonymity server acts as a channel for any communication between mobile users and continuous LBS providers. All privacy guarantees are therefore enforced at the trusted anonymity server. A mobile user $\mathcal{U}$ initiates a service session by registering itself with the anonymity server. The registration process includes the exchange of current location information and service parameters. The service parameters collectively signify a service attribute value ($\mathcal{U}.\mathcal{S}$) for use with the LBS, as well as the anonymity level to enforce while generating the requests. The service attribute is considered sensitive information whose disclosure results in a privacy breach. The anonymity server generates a set of cloaking regions $A_1, \ldots, A_n$ and a *set* $S$ of service attribute values for the requesting user. The user is present in one of these regions. Multiple range queries are then issued to the LBS provider for each of these regions, denoted as $(A_i, S)$ in the figure. The LBS generates the results for each query such that a user anywhere in the cloaking region $A_i$ with an interest in any of the values in $S$ is served. A candidate result set is formed by merging all results from the multiple range queries. The anonymity server then filters the result set and communicates the accurate result to the mobile user. A request is *suppressed* (dropped) when the anonymity requirements cannot be met. The mobile user periodically updates its location with the anonymity server and receives updated results. The user unregisters and terminates the session when the service is no longer required. We assume that a user does not change its service attribute value during a session. A separate session is started if a request with different service parameters is to be made. Therefore, a user

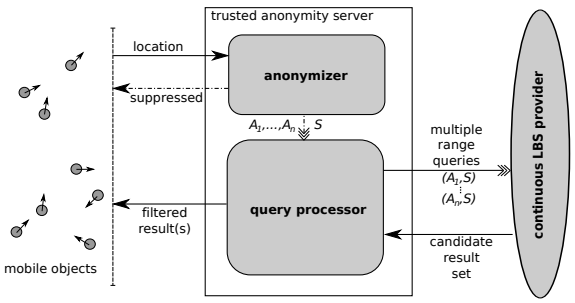


Figure 1: Schematic of the system architecture.

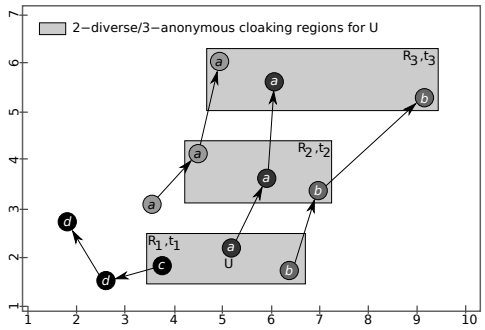


Figure 2: Example showing movement of users during a particular session registered to $U$. Values within the circles signify service attribute values of the users. Each cloaking region generated for $U$ is 2-diverse and 3-anonymous.

can have multiple sessions running at the same time. Without any loss of generality, we assume that every user has a single running session at most.

### B. Query associations in a continuous LBS

The purpose of a cloaking region is to make a given mobile user $\mathcal{U}$ indistinguishable from a set of other users. This set of users, including $\mathcal{U}$, forms the *anonymity set* of $\mathcal{U}$. For the purpose of range queries, a cloaking region for $\mathcal{U}$ is usually characterized by the minimum bounding rectangle (MBR) of the users in its anonymity set. The area of a cloaking region depends on the size of the anonymity set, as well as the time instance. Given a cloaking region $R$ at time $t$, we shall use the notation $Users(R,t)$ to signify the set of users inside $R$ at time instance $t$. Our system architecture uses multiple cloaking regions $A_1,\ldots,A_n$ while serving a single request. The requirement for this is discussed later. In the following discussion, the cloaking region of a user is the MBR of the set of users that appear in at least one $A_i$.

*Definition 1:* ***(Session Profile)*** *Let $R_1,\ldots,R_n$ be the cloaking regions of a user $\mathcal{U}$ at time instances $t_1,\ldots t_n$ respectively during a particular session, where $t_i > t_j$ for $i > j$. Let $S_1,\ldots,S_n$ be the set of service attribute values of users in the successive anonymity sets of $\mathcal{U}$ at different time instances, i.e. $S_i = \{u.\mathcal{S} | u \in Users(R_i,t_i)\}$. The session profile of $\mathcal{U}$ is then the set $SP(\mathcal{U}) = \cup_{i=1}^{n}(\{t_i\}\times\{R_i\}\times S_i)$.*

An entry in a session profile is therefore of the form $\langle t,\mathcal{R},\mathcal{S}\rangle$. For an $e \in SP(\mathcal{U})$, we shall use $e.t$, $e.\mathcal{R}$ and $e.\mathcal{S}$ to denote the corresponding terms. We shall also refer to $\mathcal{U}$ as the owner of the session profile. Consider the movement of the users shown in Fig. 2. Let us assume that a session for $U$ lasted for three time stamps $t_1, t_2$ and $t_3$, during which the 2-diverse cloaking regions $R_1, R_2$ and $R_3$ are generated. Note that users other than $U$ may terminate their session while $U$'s session is in progress. As a result, their service attribute value may change during $U$'s session. Table IIa lists the session profile of $U$ w.r.t. this session.

Ideally, no knowledge on the owner of the session is required to form a session profile. A continuous LBS can improve service quality if successive requests from the same user can be distinguished from others [16]. Hence, the

| | $t$ | $\mathcal{R}$ | $\mathcal{S}$ |
|---|---|---|---|
| 1 | $t_1$ | $R_1$ | $a$ |
| 2 | $t_1$ | $R_1$ | $b$ |
| 3 | $t_1$ | $R_1$ | $c$ |
| 4 | $t_2$ | $R_2$ | $a$ |
| 5 | $t_2$ | $R_2$ | $b$ |
| 6 | $t_3$ | $R_3$ | $a$ |
| 7 | $t_3$ | $R_3$ | $b$ |

(a) $SP(U)$

| | $t$ | $x$ | $y$ | $u$ |
|---|---|---|---|---|
| 1 | $t_1$ | 5.1 | 2.3 | Alice |
| 2 | $t_1$ | 6.4 | 1.8 | Bob |
| 3 | $t_2$ | 5.8 | 3.6 | Alice |
| 4 | $t_2$ | 6.9 | 3.5 | Bob |
| 5 | $t_3$ | 5.9 | 5.8 | Alice |
| 6 | $t_3$ | 9.2 | 5.5 | Bob |

(b) $BK(U)$

Table II: Session profile $SP(U)$ and background knowledge $BK(U)$ used during a query association attack on $U$.

anonymity server typically maintains some session identifier with the continuous LBS. All cloaking regions with the same identifier belong to the same user. This information, along with the request logs (time stamp and attribute values) accumulated at the LBS, is sufficient to build the session profile. The objective is to accurately associate a service attribute value to the owner of the profile. Note that, under a location-aware adversary model, identification of the owner of the profile implies a successful query association only when the anonymity server uses the many-to-one system of querying the LBS. In a many-to-many system, the adversary will still have to associate one of the many attribute values to the owner. Next, we formally state the background knowledge of the location-aware adversary that can be used to link the owner to its service attribute value.

*Definition 2:* ***(Background Knowledge)*** *The background knowledge of an adversary is a set $BK$ of tuples of the form $\langle t,x,y,u\rangle$ which implies that the user $u$ is known to have been at the location $(x,y)$ at time instance $t$.*

We shall only consider a subset of the background knowledge possessed by an adversary. This subset corresponds to the information that is relevant to perform a query association along with the data in a session profile. Given a session profile $SP(\mathcal{U})$, the background knowledge corresponding to the session is given as $BK(\mathcal{U}) = \{b \in BK | i \in$

| $BK(U)$ | $SP(U)$ |
|---|---|
| 1: Alice | 2: $b$ |
| 2: Bob | 2: $b$ |
| 3: Alice | 5: $b$ |
| 4: Bob | 5: $b$ |
| 5: Alice | 6: $a$ |
| 6: Bob | 7: $b$ |

(a)

| $BK(U)$ | $SP(U)$ |
|---|---|
| 1: Alice | 1: $a$ |
| 2: Bob | 2: $b$ |
| 3: Alice | 4: $a$ |
| 4: Bob | 5: $b$ |
| 5: Alice | 6: $a$ |
| 6: Bob | 7: $b$ |

| $BK(U)$ | $SP(U)$ |
|---|---|
| 1: Alice | 1: $a$ |
| 2: Bob | 1: $a$ |
| 3: Alice | 4: $a$ |
| 4: Bob | 4: $a$ |
| 5: Alice | 6: $a$ |
| 6: Bob | 6: $a$ |

| $BK(U)$ | $SP(U)$ |
|---|---|
| 1: Alice | 2: $b$ |
| 2: Bob | 2: $b$ |
| 3: Alice | 5: $b$ |
| 4: Bob | 5: $b$ |
| 5: Alice | 7: $b$ |
| 6: Bob | 7: $b$ |

| $BK(U)$ | $SP(U)$ |
|---|---|
| 1: Alice | 2: $b$ |
| 2: Bob | 1: $a$ |
| 3: Alice | 5: $b$ |
| 4: Bob | 4: $a$ |
| 5: Alice | 7: $b$ |
| 6: Bob | 6: $a$ |

(b)

Table III: Associating service attribute values using query association attacks. (a) Condition 2c in Def. 3 not met for Alice. (b) All possible query association attacks w.r.t. Table II.

$\{1,\ldots,n\}, b.u \in \bigcap_i Users(R_i,t_i), b.t = t_i\}$. We make the worst case assumption that the adversary is aware of the location of every user present in the cloaking regions of $\mathcal{U}$ at all time instances when the queries are issued. In other words, for every $t_i$ when a query is issued, there exists $|\bigcap_i Users(R_i,t_i)|$ entries in $BK(\mathcal{U})$. These entries correspond to the users that are present in all the cloaking regions generated during a session, and can potentially be the owner of the session. Table IIb lists the background knowledge used for $U$. The cloaking regions contain two potential owners, complete location information on whom is listed in $BK(U)$. Background knowledge associates users to locations while a session profile associates locations to service attribute values. The adversary relates the location data in $BK(\mathcal{U})$ and $SP(\mathcal{U})$ to link users to attribute values. We call this a *query association attack.*

*Definition 3:* ***(Query Association Attack)*** *Given a session profile $SP(\mathcal{U})$ and the background knowledge $BK(\mathcal{U})$, a query association attack on user $\mathcal{U}$ is a mapping $f : BK(\mathcal{U}) \rightarrow SP(\mathcal{U})$ such that*

1) *every $b \in BK(\mathcal{U})$ is mapped to exactly one $e \in SP(\mathcal{U})$,*
2) *every $b \in BK(\mathcal{U})$ with $f(b) = e$ satisfies*
   a) *$(b.x, b.y)$ is inside $e.\mathcal{R}$*
   b) *$b.t = e.t$*
   c) *for all $b' \in \{b^o \in BK(\mathcal{U}) | b^o.u = b.u\}$, $f(b').\mathcal{S} = e.\mathcal{S}$.*

The first condition states that a user can be associated with only one attribute value in a given time instance. The second condition prohibits the adversary from arbitrarily mapping tuples between $BK(\mathcal{U})$ and $SP(\mathcal{U})$. Conditions 2a and 2b state that a user must be inside the cloaking region (and at the specific time instance) corresponding to the entry to which it is mapped to. Condition 2c requires that a user be associated with a single attribute value across all time instances. This condition forms the basis for a successful attack since it is known that the owner of the session profile will always have the same service attribute value within the session. Consider the mapping between $BK(U)$ and $SP(U)$ shown in Table IIIa. This mapping associates Alice with the value '$b$' at time $t_1$ ($1 \rightarrow 2$) and $t_2$ ($3 \rightarrow 5$), but with '$a$' at time $t_3$ ($5 \rightarrow 6$). If Alice is the owner of the profile, then she must be associated with the same value at all time instances. In other words, the mapping fails to satisfy condition 2c and is not considered a possible query association attack. The mappings shown in Table IIIb are the only possible query association attacks in this case. Privacy is then measured as the probability that a query association attack accurately associates a user with its service attribute value.

*Definition 4:* ***(Disclosure Risk)*** *Given a session profile $SP(\mathcal{U})$, let $QAA(\mathcal{U})$ be the set of all possible query association attacks on user $\mathcal{U}$. Consider the subset $QAA_b(\mathcal{U})$ of query association attacks that accurately identifies the service attribute value of $\mathcal{U}$, i.e. given $b \in BK(\mathcal{U})$ with $b.u = \mathcal{U}$, $QAA_b(\mathcal{U}) = \{f \in QAA(\mathcal{U}) | \forall b, f(b).\mathcal{S} = \mathcal{U}.\mathcal{S}\}$. The disclosure risk for $\mathcal{U}$ is the fraction of query association attacks on $\mathcal{U}$ that accurately maps it with its service attribute value, given as $DR(\mathcal{U}) = \frac{|QAA_b(\mathcal{U})|}{|QAA(\mathcal{U}|}$.*

With reference to Table IIIb, we have $|QAA(U)| = 4$, out of which two mappings accurately associate $U$ (i.e. Alice) with the service attribute value used by her during the session (i.e. '$a$'). Therefore, $|QAA_b(U)| = 2$ and disclosure risk of Alice is 0.5.

*Theorem 1: Let $R_1,\ldots,R_n$ be the cloaking regions of a user $\mathcal{U}$ at time instances $t_1,\ldots t_n$ respectively during a particular session, where $t_i > t_j$ for $i > j$. Let $S_1,\ldots,S_n$ be the set of service attribute values of users in the successive anonymity sets of $\mathcal{U}$ at different time instances, i.e. $S_i = \{u.\mathcal{S} | u \in Users(R_i,t_i)\}$. The disclosure risk of $\mathcal{U}$ is 1.0 if $|\bigcap_i S_i| = 1$.*

*Proof:* Let $f : BK(\mathcal{U}) \rightarrow SP(\mathcal{U})$ be any query association attack. Consider a tuple $b \in BK(\mathcal{U})$ such that $b.u = \mathcal{U}$ and let $f(b) = e$. Hence, for any tuple $b' \in BK(\mathcal{U})$ with $b'.u = \mathcal{U}$, we have $f(b').\mathcal{S} = e.\mathcal{S}$ (from Def. 3, condition 2c). Note that $e.\mathcal{S}$ is the service attribute value that the adversary has associated with $\mathcal{U}$ under the attack $f$. We show that $e.\mathcal{S}$ is in fact $\mathcal{U}.\mathcal{S}$ for any $f$, and hence all possible query association attacks accurately associate $\mathcal{U}$ with its service attribute value, i.e. $QAA(\mathcal{U}) = QAA_b(\mathcal{U}) \implies DR(\mathcal{U}) = 1.0$.

By definition of $BK(\mathcal{U})$, every $b'$ has a different time stamp ($b'.t$) and is therefore mapped to a different $f(b')$. Further, only one cloaking region is associated with a time

stamp in a session profile. Hence, every $f(b')$ has a different cloaking region depending on the time stamp. Since there is a $b'$ for $t_1, \ldots, t_n$, there is a $f(b')$ for every $t_1, \ldots, t_n$. $S_i$ is the set of service attribute values associated to users in the cloaking region at time $t_i$. Therefore, any $S_i$ includes the value $f(b').\mathcal{S} = e.\mathcal{S}$, implying $e.\mathcal{S} \in \cap_i S_i$. Also, $e.\mathcal{S}$ must be the only element in $\cap_i S_i$ as the size of this set is given to be one. Given that $\mathcal{U}$ belongs to all cloaking regions in the session profile and $\mathcal{U}.\mathcal{S}$ is the parameter with which it issues its query, $\mathcal{U}.\mathcal{S}$ must also be in $\cap_i S_i$. This gives us $e.\mathcal{S} = \mathcal{U}.S$. ■

### C. Query $m$-invariance

Theorem 1 underlines why location $k$-anonymity and query $\ell$-diversity are not sufficient to prevent query association attacks. Location $k$-anonymity only guarantees that the number of users in every cloaking region is at least $k$. However, the same users may not be present across all the cloaking regions, thereby requiring a much smaller $BK(\mathcal{U})$. Historical $k$-anonymity guarantees that background knowledge must be available on at least $k$ users. Nonetheless, query association attacks can still reveal the service attribute value if only one such value is consistently present across all queries. Query $\ell$-diversity guarantees that there are at least $\ell$ distinct values in every query, but does not try to invariably maintain the same set of values across queries. The requirement for such an invariant property motivates us to consider the principle of query $m$-invariance.

*Definition 5:* ***(Query m-Invariance)*** *Let $R_1, \ldots, R_n$ be the cloaking regions of a user $\mathcal{U}$ at time instances $t_1, \ldots t_n$ respectively during a particular session, where $t_i > t_j$ for $i > j$. A cloaking region $R_j$ is query $m$-invariant if $|\cap_{i=1}^{j} S_i| \geq m$ where $S_i = \{u.\mathcal{S} | u \in Users(R_i, t_i)\}$.*

Query $m$-invariance implicitly implies location $m$-anonymity and query $m$-diversity. The principle draws upon the observation that the number of possible query association attacks will increase if a user can be associated with more number of service attribute values. However, this would require multiple values to be present at all time stamps in the session profile. Query $m$-invariance guarantees that the number of such values is not less than $m$. With reference to Fig. 2, there are two values ('$a$' and '$b$') that are invariably present across all cloaking regions. The disclosure risk in this case is $\frac{1}{2}$. In general, the following theorem provides the upper bound on the disclosure risk for query $m$-invariant cloaking regions.

*Theorem 2: Let $R_1, \ldots, R_n$ be query $m$-invariant cloaking regions of a user $\mathcal{U}$ at time instances $t_1, \ldots t_n$ respectively during a particular session, where $t_i > t_j$ for $i > j$. The disclosure risk of $\mathcal{U}$ is then at most $\frac{1}{m}$.*

*Proof:* Since $\mathcal{U}$ is present inside every $R_i; 1 \leq i \leq n$, $BK(\mathcal{U})$ contains a tuple for $\mathcal{U}$ at each time instance $t_1, \ldots, t_n$. Let $b_1, \ldots, b_n$ denote these tuples, i.e. $b_i.u = \mathcal{U}$ and $b_i.t = t_i$, for $1 \leq i \leq n$. Given a query association attack $f : BK(\mathcal{U}) \rightarrow SP(\mathcal{U})$, we have $f(b_1).\mathcal{S} = \ldots = f(b_n).\mathcal{S}$ by Def. 3, condition 2c. Consider an arbitrary $b_k$. Note that if $f$ maps $b_k$ such that $f(b_k).\mathcal{S} = \mathcal{U}.\mathcal{S}$, then every $b_i; i \neq k$ will also be mapped such that $f(b_i).\mathcal{S} = \mathcal{U}.\mathcal{S}$. By definition, such a query association attack then belongs to $QAA_b(\mathcal{U})$. Hence, we need a count of the number of attacks that map an arbitrarily chosen $b_k$ in $\{b_1, \ldots, b_n\}$ to a session entry such that $f(b_k).\mathcal{S} = \mathcal{U}.\mathcal{S}$.

Since $b_k.u$ must be associated with the same service attribute value at all time stamps, it must be one that is present in all $S_i$, for $1 \leq i \leq n$. Let $p = |\cap_i S_i|$. Further, let $q$ be the number of users in $\cap_i Users(R_i, t_i)$. The same users are present in $BK(\mathcal{U})$ at time stamp $t_k$. The function $f$ associates the $q$ users with one of the $p$ service attribute values. This can be done in $p^q$ ways, out of which $p^{q-1}$ is the number of ways where a particular user is fixed to a specific value. Hence $\frac{p^{q-1}}{p^q} = \frac{1}{p}$ is the fraction of attacks that associate $b_k.u$ (or $\mathcal{U}$) with a particular value in $\cap_i S_i$ (which can be $\mathcal{U}.\mathcal{S}$ since it belongs to all $S_i$). Since all cloaking regions are query $m$-invariant, we have $p \geq m$, implying that the fraction is at most $\frac{1}{m}$. ■

Note that, by symmetry, any user in $\cap_i Users(R_i, t_i)$ has a disclosure risk of at most $\frac{1}{m}$. Further, the number of users in $\cap_i Users(R_i, t_i)$ does not affect the disclosure risk as far as query association attacks are concerned. There is no restriction on the size of common users set (as in historical $k$-anonymity) since, under the location-aware adversary model, user identities are already assumed to be known. As far as location aware-adversaries are concerned, it is sufficient to have $k$-anonymous cloaking regions.

## III. A Cloaking Algorithm

A trivial implementation of query $m$-invariance is to randomly decide $m$ distinct service attribute values (one of them must be the user's attribute value) and use it as the *invariant set* of values (we call it $S$ in Fig. 1) across all cloaking regions in the session. However, this implementation is vulnerable to other inference attacks. Consider the user $U$ who consistently uses the service attribute value '$a$'. Hence, the set $S$ in all sessions belonging to $U$ will have '$a$'. Given that other values in $S$ will be generated randomly, an adversary can observe that the value '$a$' is present with a high frequency in the set $S$ across all sessions whenever user $U$ is in the common users set. This allows the adversary make a highly confident association between $U$ and '$a$'. The method to prevent such inference attacks is to preserve *reciprocity* in the set $S$, i.e. the set $S$ should be the same no matter which user in $\cap_i Users(R_i, t_i)$ is the owner of the session. Ideally, such a set is $\{u.\mathcal{S} | u \in \cap_i Users(R_i, t_i)\}$. However, forming this set is not possible without clairvoyant knowledge about the users that will be present in every

**Procedure 1** m-InvariantCloak(User $\mathcal{U}$)

**Require:** Mobile user $\mathcal{U}$.
**Ensure:** A set of peer groups (one of them includes $\mathcal{U}$).

1: $\mathcal{L}$ = set of available mobile users sorted by their Hilbert index
2: $\mathcal{D}_{prev} = \phi; \mathcal{D} = \phi$
3: **if** ($\mathcal{U}.invSet = \phi$) **then**
4: $\quad \mathcal{D} = m\text{-}DiverseCloak(\mathcal{L}, \mathcal{U})$
5: $\quad params = \mathcal{U}.invSet = \{u.\mathcal{S} | u \in \mathcal{D}\}$
6: **else**
7: $\quad$ **repeat**
8: $\quad\quad \mathcal{D}_{prev} = \mathcal{D}; \mathcal{D} = \phi; params = \phi$
9: $\quad\quad$ **for all** ($l \in \mathcal{L}$ in order) **do**
10: $\quad\quad\quad \mathcal{D} = \mathcal{D} \cup \{l\}$
11: $\quad\quad\quad params = params \cup \{l.\mathcal{S}\}$
12: $\quad\quad\quad$ **if** ($|params \cap \mathcal{U}.invSet| = \mathcal{U}.m$) **then**
13: $\quad\quad\quad\quad$ **break**
14: $\quad\quad\quad$ **end if**
15: $\quad\quad$ **end for**
16: $\quad\quad \mathcal{L} = \mathcal{L} - \mathcal{D}$
17: $\quad$ **until** ($\mathcal{U} \in \mathcal{D}$)
18: **end if**
19: **if** ($|params \cap \mathcal{U}.invSet| < \mathcal{U}.m$) **then**
20: $\quad$ **if** ($\mathcal{D}_{prev} = \phi$) **then**
21: $\quad\quad$ **return null**
22: $\quad$ **else**
23: $\quad\quad \mathcal{D} = \mathcal{D}_{prev} \cup \mathcal{D}$
24: $\quad$ **end if**
25: **end if**
26: $\mathcal{U}.invSet = \mathcal{U}.invSet \cap \{u.\mathcal{S} | u \in \mathcal{D}\}$
27: **return** $PartitionSet(\mathcal{D})$

cloaking region generated in the session.

### A. m-InvariantCloak

Our approach considers a $m$-diverse set of users in the first time stamp $t_1$ and uses their service attribute values as the set $S$. In successive instances, the cloaking region is adjusted so that each value in $S$ is the service attribute value of at least one user inside the region. All cloaking regions then have users with service attribute values in $S$, thereby making $\bigcap_i S_i$ equal to $S$. Since the first cloaking region is $m$-diverse, $S$ has at least $m$ elements and every cloaking region is query $m$-invariant. Further, reciprocity in the user set is preserved by using *Hilbert Cloak* [6] to determine the anonymity sets. Procedure 1 outlines this approach.

*m-InvariantCloak* starts with a list $\mathcal{L}$ of all registered users sorted by their Hilbert index. For every registered user $\mathcal{U}$, it maintains – (i) a set of service attribute values ($\mathcal{U}.invSet$) that has invariably been present in every cloaking region generated for $\mathcal{U}$ in the current session, (ii) the anonymity requirement ($\mathcal{U}.m$) for $\mathcal{U}$ and (iii) the service attribute value ($\mathcal{U}.\mathcal{S}$) used by $\mathcal{U}$ in the current session. The set of users in the first cloaking region is generated by *m-DiverseCloak* (Lines 3-5). This function returns a $m$-diverse set of users using the *Hilbert Cloak* algorithm. *Hilbert Cloak* partitions the set of users into buckets such that each bucket is $m$-diverse. Starting from the first user in $\mathcal{L}$, users are successively put into the same bucket until $m$-diversity is satisfied, upon which a new bucket is created. The algorithm returns the set of users in the bucket that contains $\mathcal{U}$. The invariant set for $\mathcal{U}$ is the set of service attribute values of the returned users (Line 5). For subsequent cloaking requests in the same session, the buckets are formed such that each contains $\mathcal{U}.m$ service attribute values from $\mathcal{U}.invSet$ (Lines 7-17). New buckets are formed until the one with $\mathcal{U}$ is found. Note that if $\mathcal{U}$ is in the last bucket, then there may be less than $\mathcal{U}.m$ distinct attribute values in it. In such a case, $\mathcal{U}$'s bucket is merged with the previous one (Line 23) if it exists; otherwise the request must be suppressed (Line 21). Request suppression is not likely as long as $\mathcal{U}.m$ is not higher than the number of possible service attribute values. Once the bucket of $\mathcal{U}$ is decided, its invariant set is updated. The update is required because the first invariant set may contain more than $\mathcal{U}.m$ diverse values out of which only $\mathcal{U}.m$ fixed values are to be retained from the second instance onwards. The merging of buckets is the only reason why an invariant set may have more than $\mathcal{U}.m$ elements.

Users in the set $\mathcal{D}$ at the end of Line 25 can be used to issue point queries along with the service attribute set $\mathcal{U}.invSet$. This will be a case of one-to-many queries. As mentioned earlier, we discourage such queries because of the possible presence of location-unaware adversaries. *m-InvariantCloak* therefore partitions $\mathcal{D}$ into *peer groups.* A peer group is a subset of $\mathcal{D}$. Each user must appear in exactly one peer group. *PartitionSet* performs this partitioning.

### B. Balancing service quality and identity disclosure risk

The objective of *PartitionSet* is to partition a $m$-invariant user set into peer groups. Every group then defines its own minimum bounding rectangle (called a *sub-MBR*) over which a range query is issued. Results to such a query are formed such that any user anywhere inside the rectangle is served. The two extremes of forming the partitions are as follows.

*(i) All users in one group:* This option provides the best protection against location-unaware adversaries as a single cloaking region containing all users in $\mathcal{D}$ is formed. However, a large cloaking region may be generated resulting in a large candidate result set.

*(ii) Every user in its own group:* This option provides the best service quality as exact location information is available to the LBS to compute the result set. However, the method provides no protection from location-unaware adversaries.

The partitioning method should therefore find the right balance between service quality and identity disclosure risks. We use a method based on maximum spatial resolution to form the peer groups. This resolution, denoted by $\alpha$, specifies the maximum area of a cloaking region that is considered acceptable for quality purposes. Procedure 2 outlines the *PartitionSet* algorithm. We assume the existence

**Procedure 2** PartitionSet(Set $\mathcal{L}$)

**Require:** A set $\mathcal{L}$ of users and system global $\alpha$.
**Ensure:** A set of peer groups.
1: Sort objects in $\mathcal{L}$ by their Hilbert index
2: *peerGroups* $= \phi$
3: *bucket* $= \phi$
4: **for all** ($l \in \mathcal{L}$ in order) **do**
5: **if** (*AreaMBR*(*bucket*$\cup\{l\}$) $\leq \alpha$ **or** $|$*bucket*$| < 2$) **then**
6: *bucket* $=$ *bucket*$\cup\{l\}$
7: **else**
8: *peerGroups* $=$ *peerGroups* $\cup$ {*bucket*}
9: *bucket* $= \{l\}$
10: **end if**
11: **end for**
12: **if** ($|$*bucket*$| < 2$) **then**
13: remove last bucket entered into *peerGroups* and merge it with *bucket*
14: **end if**
15: *peerGroups* $=$ *peerGroups* $\cup$ {*bucket*}
16: **return** *peerGroups*

of a function *AreaMBR* that returns the area of the minimum bounding rectangle of a set of users.

*PartitionSet* begins with a Hilbert-sorted list of users to partition. The partitioning is performed in a manner similar to *Hilbert Cloak*, with the difference that each bucket must include at least 2 users, and induces an area of at most $\alpha$ if more than 2 users are to be included. If the last group has less than 2 users then it is merged with the group formed prior to it (Lines 12-14). The partitioning can also be performed so that every group has a fixed number of users. We avoid this approach since user densities vary across time and space, as a result of which, the area of cloaking regions may be beyond acceptable levels. Note that the invariant set of service attribute values is not changed by the partitioning scheme. In fact, the same set is used for the range queries corresponding to each cloaking region. Hence the following theorem holds.

*Theorem 3: Let $G_1, \ldots, G_n$ be the peer groups returned by m-InvariantCloak for a mobile user $\mathcal{U}$. With reference to the system architecture in Section II-A, we define $S = \{g.\mathcal{S} | g \in \cup_i G_i\}$ and $A_i =$ the minimum bounding rectangle of users in $G_i$. The anonymity server then preserves query $m$-invariance for $\mathcal{U}$.*

## IV. Empirical Study

The empirical study compares the effectiveness of location $k$-anonymity, query $\ell$-diversity and query $m$-invariance in limiting the privacy risks in a continuous LBS. *Hilbert Cloak* is used to create the location $k$-anonymous and query $\ell$-diverse cloaking regions, while *m-InvariantCloak* is used for query $m$-invariance. The cloaking region returned by *Hilbert Cloak* for $k$-anonymity and $\ell$-diversity is partitioned similar to as in Procedure 2. The following statistics are used to evaluate the performance.

- *safeguard against query disclosures:* number of vulnerable sessions extracted using Theorem 1.
- *service quality:* area of a sub-MBR.
- *safeguard against location-unaware adversaries:* number of users inside a sub-MBR.
- *anonymization time:* time required to compute a privacy preserving cloaking region.

### A. Experimental setup

We have generated trace data using a simulator [5] that operates multiple mobile objects based on real-world road network information available from the National Mapping Division of the US Geological Survey. We have used an area of approximately $168\ km^2$ in the Chamblee region of Georgia, USA for this study. Three road types are identified based on the available data – expressway, arterial and collector. Real traffic volume data is used to determine the number of mobile users in the different road types [1]. Refer to [5] for details on the simulator.

| road type | traffic volume | mean speed | standard deviation |
|---|---|---|---|
| expressway | 2916.6 cars/hr | 90 km/hr | 20 km/hr |
| arterial | 916.6 cars/hr | 60 km/hr | 15 km/hr |
| collector | 250 cars/hr | 50 km/hr | 10 km/hr |

Table IV: Mean speed, standard deviation and traffic volume on the three road types used.

The used traffic volume information (Table IV) results in 8,558 users with 34% on expressways, 8% on arterial roads and 58% on collector roads. The trace data consists of multiple records spanning one hour of simulated time. A record is made up of a time stamp, user identifier, and $x$ and $y$ co-ordinates of the user's location. Duration of a session for a user is determined from a normal distribution with mean 10 minutes and standard deviation 5 minutes. A new duration is assigned at the end of a session. The granularity of the data is maintained such that the Euclidean distance between successive locations of the same user is approximately 100 meters. Each user has an associated $k/\ell/m$ value drawn from the range $[2, 50]$ by using a Zipf distribution favoring higher values. Service attribute values are assigned from a set of 100 values using another Zipf distribution. Both distributions have an exponent of 0.6. The trace data is sorted by the time stamp of records.

During evaluation, the first minute of records is used for initialization. Subsequently, every request is considered for anonymization. The session duration time is used to determine if a request is a new one or a continuing one. The anonymizer is then called to determine the cloaking region(s), if possible. The process continues until the session ends; a new session is started when the user issues the next request. A new service attribute value is assigned to a user at the beginning of every session. Over 4,000,000 anonymization requests are generated during a pass of the entire trace data.

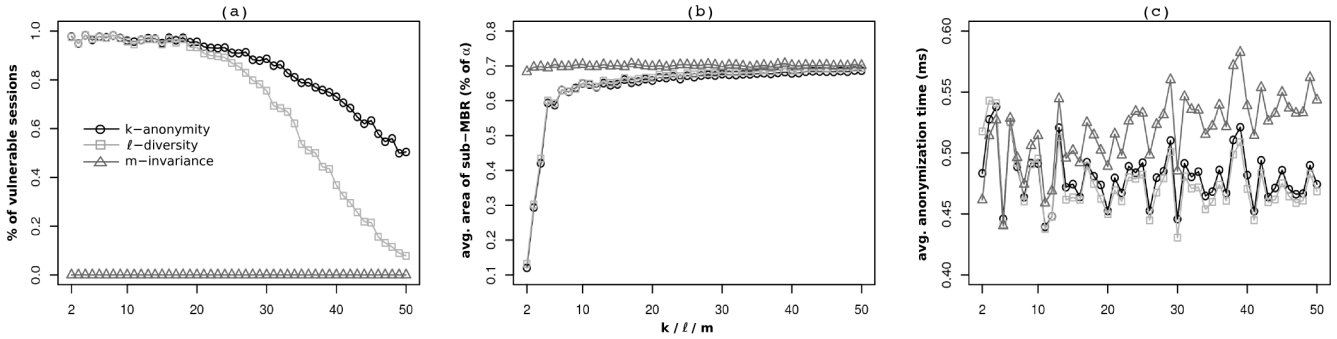


Figure 3: Comparative performance of location $k$-anonymity, query $\ell$-diversity and query $m$-invariance.

The default spatial resolution is set to $\alpha = 0.0625\ km^2$. The precision is around $250\ m$ (assuming a square area) with this setting. A service such as location-based marketing typically does not require such high precision. The entire map is assumed to be on a grid of $2^{14} \times 2^{14}$ cells (a cell at every meter) while calculating the Hilbert indices [17]. Objects in the same cell have the same Hilbert index. All simulation results are obtained on an Intel Core2Duo 2x1.86Ghz machine with 2GB memory and running Fedora 10.

### *B. Simulation results*

Fig. 3 compares the effectiveness of location $k$-anonymity, query $\ell$-diversity and query $m$-invariance. Query $m$-invariance is most effective in preventing query association attacks (Fig. 3a). Query $\ell$-diversity can prevent query disclosures in more number of sessions compared to location $k$-anonymity. This is anticipated since $\ell$-diverse cloaking regions are required to have at least $\ell$ distinct service attribute values. The invariance property in query $m$-invariance further prevents the possibility of only one attribute value being common across the cloaking regions. Both $k$-anonymity and $\ell$-diversity have almost 100% vulnerability for weaker anonymity requirements. In general, the fewer the number of service attribute values in the first cloaking region, the higher are the chances of not having any of those values in subsequently generated regions. However, by definition, such chances are reduced to zero in the query $m$-invariance model.

We also observe that query $\ell$-diversity manages to reduce the number of vulnerable sessions to impressive lows for cases with higher anonymity requirements. This is not unlikely since the probability of a particular value being in a subset of all possible attribute values is higher when larger subsets are to be formed. However, the performance is not indicative of similar behavior. Query $m$-invariance guarantees that the invariant set has a size of at least $m$. On the other hand, the low percentage of vulnerable sessions with query $\ell$-diversity only means that the invariant set is not of size one. The actual size of the set can very well be less than $\ell$. Hence, the disclosure risk is not guaranteed to be less than $\frac{1}{\ell}$.

Sub-MBR areas are typically smaller with $k$-anonymity and $\ell$-diversity (Fig. 3b). Query $m$-invariance generates comparatively larger areas for weaker anonymity requirements. This is because the users that satisfy the invariant set requirement may often be far away from each other. This is specifically true if the invariant set has values that are not frequently requested. Nevertheless, the sub-MBR area is within the spatial resolution, implying that peer groups could be formed without violating the spatial constraint (the 'or' condition in Line 5 of Procedure 2). Further, the service quality is consistent across all anonymity requirements. Fig. 3c illustrates the average time required to anonymize a request. The query $m$-invariance requirement does not impose any significant overhead in terms of computation time.

Fig. 4a depicts the impact of $\alpha$ on service quality. A very small spatial resolution (such as $\alpha = 0.0025\ km^2$) is difficult to satisfy irrespective of the anonymity level required. Given that each peer group must contain at least 2 users, the spatial constraint is easily violated and the sub-MBR area is consequently larger than specified. The area is large enough to accommodate 2 users. Fig. 4b corroborates this observation since the average number of users inside a sub-MBR is 2 for such an $\alpha$. A cloaking region of $0.0025 km^2$ resolves to a precision of around $50m$, often not required in a LBS. The resolution has a direct impact on the number of users in a peer group. Larger resolutions allow more users to be included in a group, thereby providing stronger protection from location-unaware adversaries. Note that a 2-invariant cloaking region can potentially contain a large number of users. This is specifically true when most users are inclined towards specific service attribute values. As a result, a cloaking region that contains 2 distinct attribute values essentially contains multiple users having the same value. A peer group with $45$ users on the average (in the case of $\alpha = 1.0\ km^2$) is therefore not surprising for a weak requirement such as 2-invariance. A reasonable $\alpha$ such as the default value sufficiently maintains a good balance between

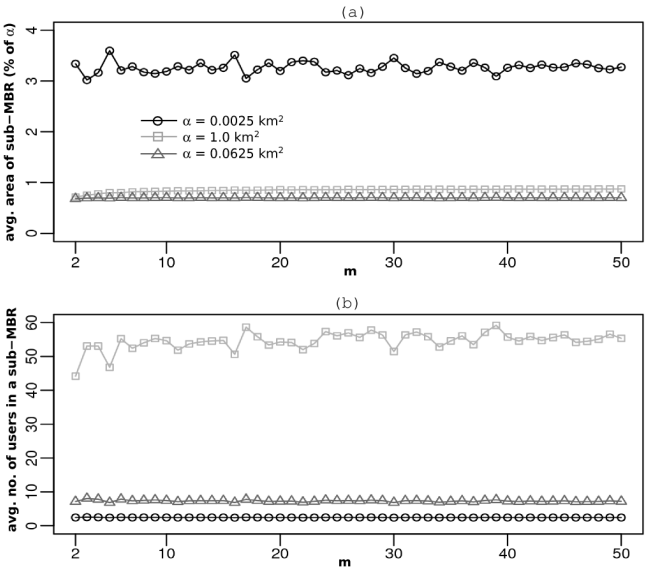


Figure 4: Impact of spatial resolution $\alpha$.

service quality and identity disclosure risks.

## V. Conclusions

Identity and query privacy must be adequately guaranteed before location-based services can be deployed on a large scale. Assuming a location-aware adversary model, we have provided a formal analysis to show that service attributes risk disclosure if the privacy model does not guarantee that an invariant set of attribute values is present in all cloaking regions generated for a continuous LBS. We therefore propose using the principle of query $m$-invariance where all cloaking regions are required to contain users with a fixed set of service attribute values. We have shown that this requirement limits the involved risk, which in itself can be controlled by the parameter $m$. We further propose a cloaking algorithm to enforce the principle and have shown its effectiveness compared to location $k$-anonymity and query $\ell$-diversity. Results on trace data generated on a real-world road network show that query $m$-invariance can be enforced without significantly affecting service quality or imposing computational overhead. Future work can be directed towards understanding the risks from query association attacks under alternative forms of background knowledge. Specifically, we are interested in exploring the privacy guarantees required to tackle adversaries with limited knowledge on user locations.

## Acknowledgment

This work was partially supported by the U.S. AFOSR under contract FA9550-07-1-0042. The authors also thank Bugra Gedik for providing the trace generator.

## References

[1] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking," in Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services, 2003, pp. 31–42.

[2] F. Liu, K. A. Hua, and Y. Cai, "Query l-Diversity in Location-Based Services," in Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009, pp. 436–442.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "$\ell$–Diversity: Privacy Beyond $k$–Anonymity," in Proceedings of the 22nd International Conference on Data Engineering, 2006, p. 24.

[4] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting Anonymous Location Queries in Mobile Environments with Privacy Grid," in Proceedings of the 17th International World Wide Web Conference, 2008, pp. 237–246.

[5] B. Gedik and L. Liu, "Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms," IEEE Transactions on Mobile Computing, vol. 7, no. 1, pp. 1–18, 2008.

[6] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location-Based Identity Inference in Anonymous Spatial Queries," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 12, pp. 1719–1733, 2007.

[7] M. Gruteser and X. Liu, "Protecting Privacy in Continuous Location-Tracking Applications," IEEE Security and Privacy, vol. 2, no. 2, pp. 28–34, 2004.

[8] B. Hoh and M. Gruteser, "Protecting Location Privacy Through Path Confusion," in Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, 2005, pp. 194–205.

[9] C. Bettini, X. S. Wang, and S. Jajodia, "Protecting Privacy Against Location-Based Personal Identification," in Proceedings of the 2nd VLDB Workshop on Secure Data Management, 2005, pp. 185–199.

[10] T. Xu and Y. Cai, "Location Anonymity in Continuous Location-Based Services," in Proceedings of the 15th International Symposium on Advances in Geographic Information Systems, 2007, p. 39.

[11] S. Mascetti, C. Bettini, X. S. Wang, D. Freni, and S. Jajodia, "ProvidentHider: An Algorithm to Preserve Historical k-Anonymity in LBS," in Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009, pp. 172–181.

[12] C.-Y. Chow and M. Mokbel, "Enabling Private Continuous Queries for Revealed User Locations," in Proceedings of the 10th International Symposium on Spatial and Temporal Databases, 2007, pp. 258–275.

[13] C. B. D. Riboni, L. Pareschi and S. Jajodia, "Preserving Anonymity of Recurrent Location-Based Queries," in Proceedings of the 16th International Symposium on Temporal Representation and Reasoning, 2009.

[14] N. Li, T. Li, and S. Venkatasubramaniam, "$t$–Closeness: Privacy Beyond $k$–Anonymity and $\ell$–Diversity," in Proceedings of the 23rd International Conference on Data Engineering, 2007, pp. 106–115.

[15] X. Xiao and Y. Tao, "m-Invariance: Towards Privacy Preserving Republication of Dynamic Datasets," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 2007, pp. 689–700.

[16] A. R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," IEEE Security and Privacy, vol. 2, pp. 46–55, 2003.

[17] X. Liu and G. Schrack, "Encoding and Decoding the Hilbert Order," Software-Practice and Experience, vol. 26, no. 12, pp. 1335–1346, 1996.