TRANSACTIONS ON DATA PRIVACY 5 (2012) 333 –376

Mobile Systems Privacy: "MobiPriv" a Robust System for Snapshot or Continuous Querying Location Based Mobile Systems.

Leon Stenneth and Philip S. Yu
Computer Science, University of Illinois, Chicago, 60607, USA
Email: lstenn2@uic.edu, psyu@cs.uic.edu

Abstract. Many mobile phones have a GPS sensor that reports accurate location. Thus, if these location data are not protected adequately, they may cause privacy breeches. Several reports are available where people have been stalked through GPS. The contributions of this paper are in two folds. First, we examine privacy issues in snapshot queries and propose a method that guarantees that all queries are protected. Previously proposed algorithms achieve a low success rate in some situations. Next, we study continuous queries and illustrate that current snapshot solutions cannot be applied to continuous queries. In this paper, a robust suite of algorithms called MobiPriv that is useful for privacy preservation in both snapshot and continuous querying mobile location based system is presented. MobiPriv addresses the shortcomings of previous work in location and query privacy. The efficiency and effectiveness of the proposed MobiPriv scheme is evaluated against previously proposed approaches. Results indicate that MobiPriv has high success rate, good run time performance, and resilience.

1 Introduction

A GPS sensor is available on many smart phones that can provide within 20m accuracy. Many third party location based services are been created randomly to provide services such as location based social networks, transportation itinerary planning, and location based coupons/deals. In these systems, the mobile users reveal their locations. If these locations are not adequately protected, it may reveal a person's political views, religious affiliations, or state of health. Knowledge of a mobile user's location may lead to stalking or unwanted adver

tisements sent to the mobile device with the marketing of products or services [23, 24, 30]. There has also been an increase in the number of GPS based harassments [11].

Location privacy preservation aims to prevent adversaries from learning a mobile user's past or current locations, or the times of the visits. To preserve location privacy, Gruteser and Grunwald [1] introduced the K-anonymity model in location context. In this model, location privacy ensures that a mobile user's location is indistinguishable from K-1 other user locations. The K value denotes the desired minimum anonymity level. A value of K = 1 implies that anonymity is not required for the message. A value of K > 1 means that the message will be assigned a spatiotemporal cloaking box that is indistinguishable from at least K-1 other messages, each from a different mobile client. Therefore, larger K values imply higher degrees of privacy. One way to determine the appropriate K value is to assess the certainty with which an adversary can associate the message. This certainty is given by 1/K.

Another level of protection demanded in location based systems is safety against request/query linking, by preventing an adversary from knowing the mobile user that has submitted a sensitive query. Sensitive queries such as, "*Where is closest XXX Rehab Center to my current location?*" should not be linked to the mobile user. In query privacy, the location component inside the query is the quasi-identifier [31]. Fortunately, the concept of K-anonymity [1] can also be used for query privacy protection. Mobile users in these frameworks are considered K-anonymous if a query cannot be distinguished from at least K-1 other queries. The technique is to expand the query location point until we locate K-1 other queries. This way, the exact query locations are hidden.

Traditional approaches such as encryption, hashing or removing the mobile user identification (e.g. user name) from the submitted request cannot overcome the privacy threats in location based systems because the quasi identifier is location and not user-id. Furthermore, encryption provides an "all (100%) or nothing (0%)" service. In these privacy aware systems, mobile users may want a personalized percentage of privacy (e.g. 80%), this cannot be done using encryption. Moreover, location based log files may be released to the public for research or data mining purposes. Due to the uncertainty that would have been introduced by encryption, this released data would be difficult to mine.

Personalized control of location information disclosure is needed in location based systems because the privacy requirement varies from one mobile user to another. For example, in a location based social network application such as Foursquare, some people may feel comfortable revealing their exact location at all times, other mobile users may feel comfortable revealing the zip code only, or the city, or country. Clearly, mobile users may need a model that allow them to define the personalize privacy level that they desire.

Several privacy focus architectures are considered for privacy aware location based services. These architectures are *client-server, trusted third party*, and *peer based*. In the strict client server architecture, clients communicate directly with the LBS by submitting a request to the LBS, the LBS then returns a response directly to the client [11, 12, 13, 14]. In the peer based model, clients communicate directly with each other to achieve location privacy [21]. The intention of the clients is to cloak with each other in order to satisfy location K-anonymity. The trusted third party model utilizes the concept of a middle-ware between the mobile user and the LBS. We sometimes refer to the middle-ware as an *anonymization server* or *AS*. Mobile requests are first sent to the middle-ware, the incoming request is then cloaked with other requests by the anonymization server before submission to the LBS. The proposed work is focused on the trusted third party architecture since these systems are now being deployed to the public [25]. The trusted third party framework is also common in privacy preserving location based systems [8, 15, 19, 29, 32].

The proposed privacy solution is effective for both snapshot and continuous query systems. A snapshot query is a request submitted once by the mobile user. For example, "*Where is the closest sushi restaurant?*" On the other hand, a continuous query is submitted at discrete time points by the same mobile user. For example, "*Continuously send me gas price coupons as I travel the inter-state highway?*" Most of the current work have focused on snapshot queries. However, since the cloaking set for the same mobile user may be different at discrete time stamps, a snapshot solution may not be sufficient in a continuous querying environment as indicated by [29].

Importantly, we observe that current anonymization techniques [7, 8, 15, 19, 28, 29] underperform if K-1 other users or requests cannot be found. Consequently, the request is discarded since it cannot be anonymized. One contribution of the proposed work is ensuring that all incoming location based requests can be safely anonymized to the desired level.

The remainder of the paper is organized as follows. Section 2 highlights the related work. Section 3 addresses the preliminaries and Section 4 discusses the dummy generation scheme. In Section 5 an overview of MobiPriv system is presented, and in Section 6 the proposed algorithms are discussed. Section 7 and Section 8 describes the evaluation matrices and the experimental results for snapshot systems. Continuous query study and evaluation is presented in Section 9. Finally, Section 10 compares the two proposed algorithms, and Section 11 concludes the paper.

2 Related work

This work is an extension of our previous work [32]. The concept of K-anonymity was originally used within the realm of relational databases [9]. The idea of location k-anonymity was first proposed by Gruteser and Grunwald in [10]. The discussion on the related work is presented in two sections. First, location based privacy in snapshot systems is discussed. Then, location based privacy in continuous querying system is discussed. In location based snapshot systems, a location based query is submitted once to the location based server. For example, "*Where is my nearest bus stop?"* In location-based continuous queries, the same query is submitted at discrete time points. For example, "*Continuously send me Groupon coupons that are redeemable at least 2 miles from my current location?"*

2.1 Location based snapshot queries

The main shortcoming of privacy aware snapshot systems is that if K-1 other requests cannot be found, the query is discarded because the desired privacy cannot be provided. Current anonymization techniques [3, 5, 6] are ineffective if K-1 other users or requests cannot be found. In these works, if K-1 other queries cannot be located within the QoS constraints, the query is dropped. This is where our research on snapshot system privacy is focused: on improving the success rate. Thus, in this work, all requests sent to the AS can be anonymized safely without being dropped. We evaluate the success rate of the proposed approach compared with previously proposed work.

In [10], the value of K in K-anonymity is static and uniform for all mobile users in the system. The framework of a personalized value of K was first introduced in [8] by Gedik and Lui. In this model, each user defines their personalized value of K as opposed to the static approach in [10] where all the users share the same K value.

The authors in [11] neglect to use anonymization servers, instead relied on a client-server technique called "*SpaceTwist*". In [11], the authors defined the concept of demand space and supply space. The demand space is the space yet to be explored and the supply space is the space already explored. The true location of the client is only known by the client. The client then sends a request using a false location to the server. On receiving the request, the server then replies with a response. The client incrementally sends more requests to the server increasing its supply space and reducing its demand space. When the supply space totally covers the demand space, the algorithm completes and the client is guaranteed to have received a correct response. We identify two obvious pitfalls with such an algorithm. The first is excess communication cost. Sec-

ondly, if an adversary has background knowledge that a person is at a particular spatial point re-identification is possible.

In [11, 12, 13, 14], strict client-server approaches are considered instead of using anonymization servers. Chow's work in [21] is based on the peer to peer approach for privacy preservation.

The proposed work is different, we concentrated on the trusted third party architecture with the anonymization server. Below, the four most relevant works to the proposed algorithms are discussed. The four most relevant works are *Interval Cloaking* [10], *Clique-Cloaking* [8], *Casper* (Pyramid Based) [19], and *PrivacyGrid* [15].

— *Interval Cloaking* [10] - This cloaking technique iteratively divides the region into quadrants. The region before division is called q_{prev} . If after division the quadrant that the user resides contains more than K users, then iterative division continues. If after division the quadrant that user resides contains less than K users, then q_{prev} is returned as the cloaking box.

— *Clique-Cloaking* [8] - A clique in a constraint graph is identified. The nodes in the graph are messages and the edges are formed between two messages if the cloaking region of the messages overlaps to include the messages. Messages can be cloaked together if they are neighbors in the graph and the K requirement can be satisfied. If messages cannot be cloaked they are expired and dropped. Also, users tend to be close to edges of the minimum bounding rectangle.

— *Casper* [19] – Casper maintains an anonymizer and a privacy aware query processor. For anonymization and cloaking, a pyramid structure is maintained. The cells in the region contain the number of mobile users active in the cell. If the current region/cell of user cannot satisfy the value of K or A_{min} then a neighboring cells are considered. Casper has a superb run time performance. However, Casper does not return the smallest cloaking region and also it is expensive (*updates and cloaking*) to maintain the pyramid structure. The same authors of [19] also introduced *TinyCasper* [18]. TinyCasper is solely for wireless sensor networks. We evaluated the proposed work against Casper [19].

— *PrivacyGrid* [15] - *PrivacyGrid* guarantees a small CR by using a *Bottom-up*, *Top-down,* or *Hybrid* approach. In the *Bottom-up* approach, the mobile user's cell is expanded to meet the K-anonymity and L-diversity requirement. For a given cell it may expand to its immediate neighbor east, west, north or south. The next cell to be chosen is the cell with the highest mobile user count. This cell will be included in cloaking box. The *Top-down* approach selects the largest possible cloaking box that satisfies the users QoS requirement. If K users cannot be found in this

cloaking box, the query cannot be satisfied. If K users are found, then the algorithm attempts to prune the cloaking box to a smaller cloaking box and verifies if K mobile users are still present. The *Hybrid* approach makes a decision to use bottom up or top down, depending on the value of K and the QoS (i.e. spatial tolerance). The *Hybrid* scheme combines the strengths of both the top down and the bottom up approaches.

We evaluated the effectiveness of the proposed algorithms against, Casper, PrivacyGrid's Bottom up, PrivacyGrid's Top down, and PrivacyGrid's Hybrid schemes.

2.2 Location based continuous queries

Most of the current work focused on snapshot queries. Since the cloaking set for the same mobile user may be different at distinct time stamps, a snapshot solution may not be sufficient in a continuous querying environment. An aggregation or intersection of multiple snapshots in a continuous query can lead to privacy breaches [29].

Work on location based continuous queries includes [28, 29]. In [29], the same set of mobile users are aggregated and submitted to the location based system by the anonymization server for each request from a mobile client. The main shortfall of this model is the possible reduction of quality of service for subsequent mobile requests from the same mobile client. The quality of service problem of [29] was addressed in [28], and [28] assumes that the transportation mode can be determined from a set of GPS points [33]. Thus, in [28], they anonymize each local and global candidate set by considering similar transportation modes using a dynamic cloaking strategy.

Since current snapshot solutions are not applicable for privacy reservation in continuous querying systems, a robust and effective model for location based privacy preservation for continuous queries is proposed in this work.

Dummies were first introduced by Kido [12]. The proposed work is different, realistic and diverse dummy request are generated on the anonymization server with respect to K-anonymity. The dummy related work in [12] uses the strict client server architecture without the anonymization server. Additionally, [12] did not consider K-anonymity or diversity constraints.

3 Preliminaries

To define the problem of privacy preservation in mobile location based services, first, we need to define the privacy information that we are interested in preserving. Second, the threat model or background knowledge that an adversary

may use to attack the privacy of users in a location based system. Variations in formulation of these issues may lead to different versions of privacy preservation in mobile location based systems. In this paper, we discuss a version which we believe is useful in many location based systems.
In this section, we will present formal definitions, requirements, architecture, threat model, and adversaries' knowledge.

3.1 Formal definitions

Definition 1.1 (LBS Quasi-Identifier) A set of attributes {q1, . ., qn} of a mobile query is called a LBS quasi-identifier if these request parameters can be linked with external data to uniquely identify at least one mobile client in the system.
One example of quasi identifier (QI) in location based systems is the location {*latitude, longitude*} of the mobile user submitting the request.
The above definition recalls the definition of QI introduced in relational databases [31].

Definition 1.2 (Location-based snapshot query) A location-based snapshot query $Q_t^{snapshot}$ is a location based query submitted at time t. Such a query will not be submitted at time *t+n,* $\forall n > 1, t \geq 0$.

Definition 1.3 (Location-based continuous query) A location-based continuous query $Q_t^{continuous}$ is a sequence of snapshot queries submitted at discrete time points. $Q_t^{continuous}$ = { $Q_t^{snapshot}$, $Q_{t+1}^{snapshot}$, $Q_{t+2}^{snapshot}$, $Q_{n-1}^{snapshot}$, $Q_n^{snapshot}$ }. For example, "*Continuously send me information on public passenger buses that are within five minutes from my current location?"*

Definition 1.4 (Region request) A region request R_t is a group of queries that are anonymized together to satisfy the K-anonymity requirement, an area encompassing all the queries is submitted to the LBS service provider. Each region request $\mathfrak{R}$ is formalized as:

$$\mathrm{Rt} = (\mathfrak{R}_{\mathrm{id}}, \mathbb{Q}_{\mathrm{set}}, \mathcal{L}_{\mathrm{t}})$$

where $\mathfrak{R}_{\mathrm{id}}$ is the identifier of the region request, $\mathbb{Q}_{\mathrm{set}}$ is the set of queries contained in Rt and $\mathcal{L}_{\mathrm{t}} = \left(\ell^{\mathrm{t}}_{\mathrm{x}-}, \ell^{\mathrm{t}}_{\mathrm{y}-}, \ell^{\mathrm{t}}_{\mathrm{x}+}, \ell^{\mathrm{t}}_{\mathrm{y}+}\right)$ is the location of the bounding rectangle for $\{\mathbb{Q}_1, \mathbb{Q}_2, .. \mathbb{Q}_{n-1}, \mathbb{Q}_n\}$ at time t.

Definition 1.5 (Local K-anonymity) A region request (R_t) satisfies local K-anonymity (K_{local}) if it contains at least K_{local} different location based requests.

Definition 1.6 (Global K-anonymity) A continuous query $Q_t^{continuous}$ satisfies global K-anonymity if the number of requests in intersection of the region requests or snapshots in the continuous query is at least K_{global}. Therefore, for *n*

region requests R_1, R_2 ... Rn in the continuous query $|R_1 \cap R_2 \cap \ldots Rn| >= K_{global}$ and $K_{global} <= K_{local}$.

3.2 Requirements

— When a mobile user submits a location based request, she should be given the option to hide her location. The location is an identifier in location based services. Let $q_m(x, y)$ be a query q submitted by mobile user m from a point (x, y) where latitude = x and longitude = y. The request q_m is converted to some $q_{m'}$ $((x1, x2), (y1,y2))$, where $q_{m'}$ is the new query and $(x, y) \in ((x1, x2), (y1,y2))$ such that $((x1, x2), (y1,y2))$ is a region encompassing the point (x, y).

— When a mobile user submits a sensitive query, the query should not be linkable to the mobile user. For example, a query such as "*Where is the closest parking lot to the HIV rehabilitation center?"* should not be linkable to the mobile user that submitted the query. This kind of query is sensitive query and may be embarrassing to the sender if the query is revealed.

— Queries should not be discarded if K-1 other mobile clients or mobile requests cannot be found. In previous work [8, 15], if K-1 other mobile queries cannot be found, the query is discarded. Any privacy model that suffers from this pitfall is referred to as "best effort". On the other hand, MobiPriv is a guaranteed service, and all queries can be anonymized successfully without a delay. Let N be the total number of mobile requests sent by mobile clients to the anonymization server. The number of mobile request that can be anonymized successfully by the anonymization server is M_t. The anonymization server ensures that:

$$\frac{|N|}{|M_t|} * 100 = 100\%$$

This implies that all the queries sent to the system will be anonymized successfully and none will be dropped. In our evaluations we refer to this property as the *success rate* of the privacy algorithms.

— Using historical data such as past queries should not influence query linking. This is frequent in continuous querying systems where the adversary can aggregate all the regions that he suspects a mobile client visits and use this information to link queries to the mobile client. Assume that mobile user u_k submits multiple requests with a privacy requirement of K from points in the regions R_1, R_2, R_3…R_n. Then,

$$R_1 \cap R_2 \cap R_3 \cap R_4 \cap \ldots R_n \neq u_k$$

— When a mobile user submits a request, if K-1 other requests are not available at the time of request, the system should not wait for these requests to become available. Some location based services provide emergency and lifesaving functionalities, therefore delays cannot be tolerated.

3.3 System architecture

The system consists of mobile devices with positioning capabilities, location based services (LBS), wireless networks, and the proposed algorithms running on a privacy aware middle-ware called an anonymization server (AS). Below, we discuss each component. See Figure 1 for a layout of the architecture.

Mobile device (clients): Mobile device includes mobile phone, PDA, and other devices such as laptops with positioning capabilities. First, each mobile device computes its physical location from the GPS or Wi-Fi component on the device. Mobile users specify the privacy requirement that they desire from the user interface of their mobile device in the proposed system. Both the personalized privacy requirement and the query containing the location data is forwarded to the anonymization server. Observe, for a typical query the location component of the query may not be the current mobile user's location. For example, Alice is at Point A and Alice submits a query requesting the closest buses to Point B. In this case, there is no need for Alice to reveal that she is at Point A.

Anonymization server (AS): The anonymization server knows the location of all the mobile users. The physical location computed by the mobile device is sent to the anonymization server with the query. The role of the anonymization server is to privatize location and the request before submitting it to the location based system. We assume that anonymization server is the trusted third party, the adversaries loiter between the anonymization server and the location based system.

Secure communication service: The communication link between the mobile clients and the anonymization server is assumed to be wireless connections that are secure.

Service provider (LBS): The service provider provides location based services to its subscribed mobile users. On receipt of a request from the anonymization server, the location based server processes the request and returns a response to the anonymization server. The service provider has the ability to process a given cloaked region and also process an exact point. The service provider is not responsible for privacy policies of the mobile clients.

Operation flow: Mobile users submit requests incorporating positioning information such as current location (latitude and longitude) as a parameter of the

request to the AS. The AS then cloaks the client's query location point into a region containing *K-1* other mobile user request.

The AS then forwards the aggregate region request to the LBS. The LBS processes the query and sends a response. This response sent by the LBS is generic and should be filtered to get precise results. Filtering can be done on the AS or by the mobile client. Filtering on the mobile device may be costly due to limited battery power and processing capabilities. A diagram depicting the architecture is shown in Figure 1.

One novelty about our approach is, if *K-1* other mobile user request cannot be found we generate realistic diverse dummies instead of dropping the query as in previous anonymization approaches that uses the trusted middle-ware.

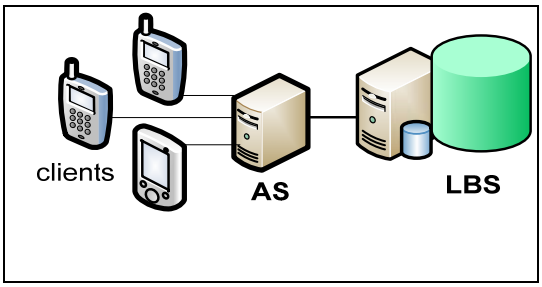

Figure 1- LBS with Anonymization Server (AS)

3.4 Threat model and adversary's knowledge

To evaluate the privacy protection of the proposed system, we consider the following threat model. Mobile users communicate precise personal location information, while adversaries' intention is to decipher the physical location of the mobile clients.

Additionally, as the mobile clients may submit sensitive queries, adversaries intend to infer which mobile user has submitted current or past queries. For example, if a person submits a query such as "Where is the less expensive bar in the red light district closest to Downtown Chicago?" Then, this query should not be linked to the sender.

In summary, the adversaries have two main goals: (1) Discovering the location that the submitted requests are related to, (2) Correlating a sensitive query to a mobile user. Adversaries loiter between anonymization server and location based server and is aware of the time that the mobile users may submit the queries.

Adversaries may "look up" the correlated street address corresponding to a mobile user location (latitude, longitude) or query location (latitude, longitude) using reverse geocoding techniques and they are aware of the location of some mobile users. Moreover, adversaries may observe the region sent by the anonymization server to the location based server and determine the number of que-

ries in that region request. Furthermore, the adversary may have background knowledge about the victim. Background knowledge includes home address, office address, etc. Additionally, adversaries may perform query linking by observing the location component inside a query. Advance adversaries may also listen over time and correlate multiple queries sent by the same mobile users. These assumptions are realistic. Below me model some of the adversaries' capabilities and we will show that the proposed approach is resilient to these capabilities

— **Count** (*R*) – Given a region R, with multiple mobile users or mobile requests $\{r_1, r_2, r_3 \ldots r_{k-1}, r_k\}$. The adversary can determine the number of queries in *R*. Thus, from R, the adversary can determine an integer value corresponding to $|\{r_1, r_2, r_3 \ldots r_{k-1}, r_k\}|$.

— **Lookup** (*lat, lon*) – Given a location point (latitude, longitude), the adversary can perform reverse geocoding and determine the high level street address.

— **Intersect** ($R_1, R_2, R_3, R_4 \ldots R_{n-1}, R_n$) – Correlates multiple requests from the same sender. Given ($R_1, R_2, R_3, R_4 \ldots R_{n-1}, R_n$), the adversary can perform $\{R_1 \cap R_2 \cap R3 \cap R4 \ldots R_{n-1} \cap R_n\}$ and determine common requests across regions.

4 Dummy requests

We define the term dummy request, or dummies, to be a *fake* mobile request U_a automatically and realistically generated by the system. Dummies should be generated in a way that an adversary cannot differentiate a dummy from a real user request U_r. In LBS systems such as road navigation systems, mobile users send continuous position queries. If dummies are generated randomly, then adversaries can easily find the difference between the real user and the dummies. For example, real requests tend to be related to road networks and dummies may wonder off into the Euclidian space if they are randomly generated. In our algorithm, the dummies are generated relative to the temporal and spatial property of the real user request. The identification numbers of the realistic dummies are taken from the dummy profile. The proposed dummies requests are also *diverse* and are different from the mobile user request.

This paper is not the first to use dummies as a concept to increase privacy in location based systems. However, it is the first to consider dummy request generation on the anonymization server (AS). *Kido et al* [12] introduced the concept of dummy location generation in location based systems but did not consider K-

anonymity. In [12], the client sends the true position along with the dummy positions to the LBS. The LBS then responds with answers to both the true position and the dummy position. The proposed work is different as [12] did not consider the anonymization server instead only considered the client server architecture. We also considered dummy query generation and not dummy location generation as described in [12]. One disadvantage of the approach in [12] is the high processing cost on the mobile device to filter the results that were returned by the LBS. The excess filtering cost on the mobile client negatively affects the short battery life and limited processing power of hand-held mobile devices. Consequently, MobiPriv filters on the anonymization server.

Further, if an adversary has knowledge of history then a user may be re-identified in [12] by taking the intersection of the regions that the mobile user sent the requests from. Thus, [12] is not useful in a continuous query environment.

4.1 Dummy profile

In MobiPriv, the dummy profile is a file containing a list of all mobile users in the system along with corresponding dummy user identification numbers. We define *profileCount* to be the number of dummies associated with a real mobile user on the dummy profile. We initialize *profileCount* to be the maximum K value allowed in the system and construct the dummy profile as follows. For each possible real user U_r in the system, we associate a set of dummies with the real user identification. To generate dummy requests for a particular user, we first consult the dummy profile and take the dummies from the dummy profile in topological order. We generate dummy requests to satisfy the local anonymization groups which then become candidates to prevent query linking in continuous queries. The proposed work (i.e. MobiPriv) is the first to consider realistic diverse dummy user generation on the anonymization server.

We will now explain how to generate realistic dummy requests. Figure 3 shows a diagram of a cloaked request. The parameters dx and dy are the user defined spatial tolerances, and coordinates x, y represents the location (latitude and longitude) of the request. From dx, dy, x, and y we build the bounding box such that parameters $x1$ and $x2$ are the regions x-coordinates. Likewise, y and $y2$ are the y-coordinates.

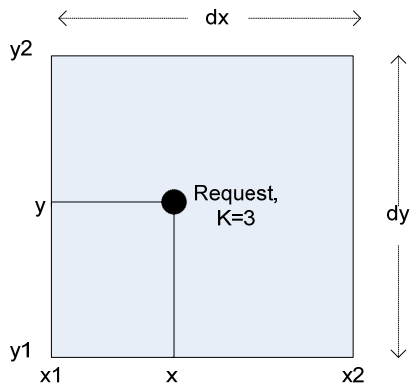

Figure 2-Dummy request generation

First, the following offsets are defined; (1) x_{offset}, and (2) y_{offset}. The values for these offset are computed based on the spatial properties (i.e. x and y) of the incoming request as shown below.

$$x_{offset} = \min\,(x2 - x, x - x1)/2$$

$$y_{offset} = \min\,(y2 - y, y - y1)/2$$

Let *<0,1>* means a random number between 0 and 1 inclusive. Given these offsets, we then use x + <0,1> * x_{offset} and y + <0,1> * y_{offset} as spatial properties for the dummy requests. Therefore, for each dummy request, the following is computed.

$$dummy^{x}_{offset} = x + < 0,1 > * \; x_{offset}$$

$$dummy^{y}_{offset} = y + < 0,1 > * \; y_{offset}$$

Where $dummy^{x}_{offset}$ is the dummy's x offset and $dummy^{y}_{offset}$ is the y offset. The temporal property of the dummy requests can be ascertained similarly. Using this strategy, the dummy requests are related to the spatial and temporal properties of the real request. Hence, the dummy requests may be perceived as real requests by adversaries.

The dummy generation algorithm is presented (see Algorithm 1). In line 4, *profileCount* is declared to be 50, likewise in the experimental evaluations. This implies that the system generates at most 50 realistic dummy requests to satisfy the privacy requirements. Line 6, the proposed work reads the dummy profile for the mobile user that submits the request.

From the dummy profile, the identification of any dummy request that will be generated is known. In lines 7- 18, it is guaranteed that the total number of required dummies is less than *profileCount* and the proposed work generates all the dummy identifications from the dummy profile (line 12). Likewise, the temporal and spatial properties of the dummies are related to the mobile user (line 9, 10, 11).

```
/* Algorithm 1: Realistic-Dummy-Generation*/
1. precondition: mobileUserId!=null, totalDummies>0
2. input: mobileUserId, totalDummies,C /* C is query*/
3. method:
4. profileCount ←50, count ← 0 ,xoffset , yoffset , toffset, dummies []
5./ * xoffset , yoffset , toffset are computed as shown in Section 4.1 */
6. profile [] = read_dummy_profile(mobileUserId,totalDummies)
7. if(totalDummies <= profileCount)
8.   while(count<totalDummies)
9.       t = C.t  + <0,1>* toffset
10.      x = C.x + <0,1> * xoffset
11.      y = C.y + <0,1> * yoffset
12.      id = profile [count]
13       C' = diversify(C) /* diversifying the dummy query */
14.      newDummy = createDummy(id,x,y,t,C')
15.      dummies [count] = newDummy
16.      count++
17.  end while
18. end if
19. else
20. return dummies
21. end else
end
```

In line 13, to prevent the *homogeneity attack,* the query is *diversified*. For example, if the mobile request is related to 1036 N. Michigan Ave in Chicago, the dummy request may be related to 1038 N. Michigan Ave Chicago. The proposed work ensures that the dummies' query point is not the same as the real mobile users' query point. Instead, a different building or symbolic address in the region is utilized. The algorithm then creates the new dummy request based on the time, location, and query of the mobile user (line 14). In line 15, the new dummy is added to the list of dummies. Finally, the candidate list of dummies is returned in line 20.

In this algorithm, the dummy request becomes realistic and diverse by relating its temporal, spatial, and query point properties to the real mobile user's request.

4.2 Dummy diversity

If all the *K* users in the region submit request to the same symbolic address such as the same movie theater, K-anonymity fails, because the adversary can infer that some people are interested in that symbolic address. This kind of attack is referred to as the *homogeneity attack* [26]. For this reason the concept of diversity is considered [26]. Diversity adds another dimension to the privacy level and ensures that our query locations and query contents are diversified. For example, request should span across different postal addresses or different buildings. The dummy requests that are generated are diverse and are submitted to different symbolic addresses using a reverse geocoding scheme.

Using reverse geocoding, the location point of interests of the user is converted to a readable street address. Reverse geocoding is the process of converting a location point to an address or place name. For example, reverse geocoding latitude: 41.976216 and longitude: -87.90331, produces the address 99 Access Road Chicago, Illinois, 60666, USA. Based on the address or place that is returned, the proposed work constructs the dummies' point of interest. For example, a dummy point of interest for the aforementioned latitude and longitude would be at 100 Access Road, Chicago, Illinois, 606666, USA.

4.3 Query privacy in MobiPriv implies location privacy

We now show that anonymization with *K*-1 other mobile requests as done in *MobiPriv* is a sufficient condition for location privacy. First, let's define a query $q_m(x,y)(X,Y)$, where q is the query submitted by mobile user *m*, located at location (x, y), and the location component of the POI in the query is (X,Y). Both x and X corresponds to the latitude, y and Y corresponds to the longitude. We have two types of queries (1) **Type 1** - The mobile user location is a component in the query. Therefore, (x, y) = (X, Y). (2) **Type 2**- The mobile user location is not a component in the query. Therefore, (x, y) != (X, Y). Location privacy ensures that (1) The location (x, y) is not revealed or (2) The location (x, y) is revealed and is indistinguishable from other mobile user location.

— Consider **type 1** queries, the (x, y) location of the mobile user is revealed in the query. In this case, the current mobile user location is a component of the query. For example, the query "*return the closest busses to my current location?*" In this case, the mobile user's current location constitutes the query. Therefore, (x, y) ≡ (X, Y). In MobiPriv, we anonymize with regards to the query location point, in this case location (X,Y) is expanded to find *K-1* other mobile requests. Since (x, y) ≡ (X, Y) and a mobile user cannot submit multiple requests simultaneously, the location (x, y) also become anonymous since the adversary only knows the location of some of the mobile clients in MobiPriv. For example in Figure

2 (a), the entire grid represents the LBS world as seen by the anonymization server. The mobile users are M1, M2 and M3. They submit queries Q1, Q2 and Q3 respectively. The (row, column) pair represents the location of a mobile user or a location component of the query. Therefore, the locations of M1, M2 and M3 are (4, 0), (4, 1) and (0, 0) respectively. Likewise, the location component of the queries Q1, Q2 and Q3 are (4, 0), (2, 2) and (2, 3) respectively. Observe, only Q1 is a type 1 query, the remainder of this discussion is based on Q1. Mobile user M1 submits the query Q1 to the AS, with K=2, we perform spatial expansion of the query point to find other queries. The closest query is Q2; Q1 is therefore cloaked with Q2. We therefore submit a region consisting of Q1 and Q2 to the LBS. The query privacy achieved is 1/K (1/2) as required. The location of mobile client M1 cannot be deciphered by the adversary since we submit a region and the adversary only knows the location of some mobile users in the system.

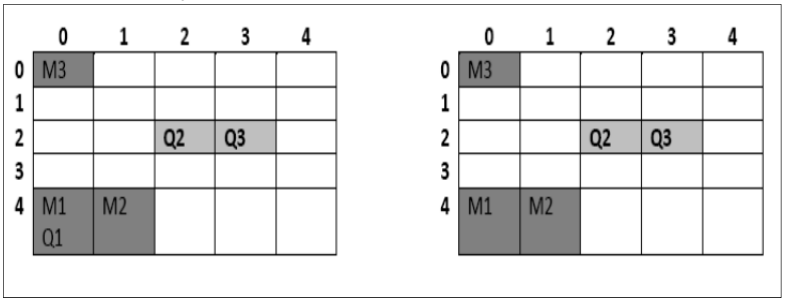

Figure 3 (a, b) - MobiPriv Query Privacy implies Location Privacy

Proof that query privacy implies location privacy for type 1 queries
For type 1 queries, the client's location M is a part of the query N. The proof is by contradiction, we assume to the contrary that protecting the query N would not protect the client's location M. Now, to protect N, the propose approach expand all the location components in N until K-1 other requests are found. Thus, the adversaries will observe that there are K interests in some location and has a 1/K chance of correlating the interests. Now, since we expand the query location points and M (i.e. the client's location) is a component of the query, then M too would have been expanded and protected up to 1/K. This contradicts our initial assumption that protecting N would not protect M.

— Consider **type 2** queries, where the location of the mobile user is not a component of the query. For example, a mobile user submits a request for the closest parking lot to Point B, and the mobile user is located at Point A. In this case, there is no need for the mobile user to send his current location (Point A) to our anonymization server; hence location pri-

vacy is not breached. For example, in Figure 2 (b), Mobile user M1 submits a query Q2. Q2's location point (2, 2) is not related to M1's location (4, 0). For query privacy, we expand the region around Q2 to find Q3 and submit a region containing Q2 and Q3. Since M1's location is not relevant to the query, M1's location can be kept hidden.

This implies that regardless if the mobile user reveals their location in the query (i.e. type 1 queries) or not (i.e. type 2 queries), query privacy in MobiPriv always imply location privacy.

5 MobiPriv: System overview

MobiPriv is a three tier model similar to the trusted third party mechanism discussed earlier. The proposed suite of algorithms runs on the anonymization server. The first step is a *request submission* phase where the user submits a request. The request contains the percentage privacy level required by the user. Next, is the *transformation* phase where this personalized user percentage of privacy is converted to some value of *K* by a *mapping function*. Other phases include the *perturbation* phase whereby we form a region based on the spatial resolution from the request. The real user is then placed in the region. Next, we have two anonymization options, *CloakLessK* and *CloakedK*.

In *CloakLessK,* once the AS receives a mobile user's request it (i.e. the AS) quickly generates K-1 dummies in the perturbed region, and then sends the request to the LBS. In *CloakedK*, once a mobile user's request is received, before dummies are generated, the algorithm first verifies if other requests are close enough to this mobile user's request and can be cloaked together. Finally, if after cloaking with neighbor requests, *K-1* real mobile user requests are still not in the region, the proposed solution will generate realistic dummies as the remaining requests. Finally, the region request is sent to the LBS. The proposed algorithms maintain the following good contributions.

— All queries are given a response. In previous privacy models, some queries may not be given a response because *K-1* queries are not available to meet the QoS required, or *K-1* users are not available in the system. In these models [6, 8, 15, 19, 29], the mobile user defines their own personalized spatial tolerance, temporal tolerance, and privacy requirement. If the query requirement demanded by the mobile clients cannot be satisfied, the queries are dropped from the system. In MobiPriv, we have a drop rate of 0%. In the evaluation by experiments, s*uccess rate* is used as a matrix to measure reliability.

— Elimination of the temporal cloaking problem present in [8, 15, 19], whereby the system waits for K-1 other requests to be available before anonymizing the request. Some LBS systems provide emergency services and cannot tolerate delays.
— Elimination of the spatial cloaking problem whereby the system continues to extend the region to find K-1 other queries. The maximum size of the region is defined by the mobile user (i.e. *dx* and *dy*) from the query submission phase. We cannot expand the region beyond this restriction.
— Communication Cost Reduction, the proposed work presents two privacy aware models *CloakLessK* and *CloakedK*. *CloakLessK* has a fixed communication cost regardless of the number of mobile users submitting request. *CloakedK* has a much lower communication cost, the communication cost of *CloakedK* improves much more than *CloakLessK* as the number of request increases.
— Query linking in continuous queries is eliminated. The algorithms generate realistic and diverse dummy requests using a dynamic dummy profile strategy. This strategy eliminates this kind of attack. Experimental results revealed that the proposed work is resilient to this category of attack.

5.1 Mobile request

In *MobiPriv*, a mobile user submits a *request* in the form <*user_id, msg_num, {t, x, y}, {dx, dy, dt} ,P, C*>. The *user_id* is the unique identification of the user and *msg_num* to be the message identification. The combination of *user_id* and *msg_id* is unique for all messages. Also, *{t, x, y}* is the temporal and spatial property of the request. Additionally *dx, dy , dt* are the spatial and temporal resolution demanded by the mobile user. *P* is the percentage of privacy desired, and *C* is the request or query content. Recall, large spatial tolerances produces less accurate responses and high temporal tolerances result in longer message delays.

The point request above is converted to a region request of the form <*user_id', msg_num', {x1,x2}, {y1,y2}, {t1,t2}, C*> where *user_id', msg_num'* are hashed versions of *user_id*, and *msg_num* respectively. The parameters *x1, x2* are the request region's *x* coordinates, *y1, y2* are the request region's *y* coordinates, *t1* and *t2* represents the request region *z* coordinates. The three coordinates are used to form the cloaking box. *C* is the request content that should always be preserved. The size of the cloaking box is bounded as follows:

$$x2 - x,\ x - x1 <= dx$$
$$y2 - y,\ y - y1 <= dy$$
$$t2 - t1 <= dt$$

We refer to this process of converting a request from an exact point to a rectangular region as spatial and temporal cloaking.

Mobile request example:
<user_101, msg_num_004, {11:15am,-87.653, 41.85}, {60m, 50m, 5s}, 90% Privacy, "Shortest route from current location (latitude = -87.653, longitude = 41.85) to XXX rehab center (latitude = -87.6215, longitude = 41.210)">

5.2 Transformation and mapping function

MobiPriv allows mobile users to define the percentage value of privacy that they desire. Then it uses a mapping function to determine a suitable value of *K* based on the desired percentage privacy specified. The conversion from a percentage level of privacy to a suitable value of K is referred to as the *transformation* phase. AS administrators are not limited to one mapping function, they may define their own mapping function. A mapping function may reflect the nature of the underlying location based system. For example, in the proposed algorithm, a mapping function such that percentage privacy is related to the certainty with which an association between a user and a message can be ascertained is considered. Let *P* be the percentage of privacy desired by a real user U_r, a corresponding K-anonymity value is computed as follows:

$$K = ceiling\ (100/\ (100\text{-}P)),\ P < 100$$

6 Algorithms

We presented the realistic dummy generation algorithm in Section 4.1. Next, two other algorithms in MobiPriv, *CloakLessK* (i.e. Algorithm 2) and *CloakedK* (i.e. Algorithm 3) are discussed.

6.1 CloakLessK

First, the mobile user submits a request to the anonymization server (line 2). In line 4, we hash the user identification and the message identification. Each request contains the personalized privacy requirement of the mobile user that is converted to some *K*. This step is referred to as the transformation phase (line 5). The proposed algorithm then creates regions depending on *dx*, *dy*, *dt* (spatial and temporal tolerance) of the user request. The total number of users (real user and dummy users) in the region is determined by *K* from the *transformation* phase.

The real user is randomly placed in the bounding region (line 7), the AS then generates *K-1* dummies (line 8). The dummies generated are realistic and diverse. Realistic diverse dummies are used to protect against the corollary history attack. However, this protection comes with a cost (see experimental results). Finally, the cloaked region request is sent to the LBS for processing in line 9.

```
/* Algorithm 2: CloakLessK */
1. pre-condition: request!null
2. input: request <u_id,msg_num {t,x,y},{dx,dy,dt},P,C>
3. method:
4. hash (u_id ,msg_num )
5. K = transformation (P)
6. createGrid(request.dx, request.dy ,request.dt)
7. insertRealUser()
8. insertDummies (request.u_id, request.K-1)
9. sendRegionRequestToLBS()
10. end
```

Next, *CloakedK*, an algorithm that reduces the communication cost of CloakLessK is discussed.

6.2 CloakedK

The principal difference between *CloakLessK* (Algorithm 2) and *CloakedK* (Algorithm 3) is the addition of line 8 in the algorithm shown below. The input to the algorithm is a mobile user's request in line 2. In line 4, we hashed the identification parameters such as *user_id* and *msg_num*. The percentage of privacy demanded by the mobile client from line 2 is transformed to a suitable value of *K* in line 5. We create the bounding rectangle (region) around the location point in the request in line 6 and line 7.

Additionally, at line 8, of the *CloakedK* algorithm, instead of sending one real mobile user request along with dummies in a region request, the AS now aggregates multiple real user's requests in the same cloaking region before sending to the LBS. Only if real mobile users are less than K-1, then dummies are considered.

We now discuss the cloaking methodology of line 8. Once a user submits a request to the LBS, we generate a perturbed box as discussed in Section 5.1. The size of the box is dependent on *dx, dy, dt.* Before inserting any dummies inside the box, we query if this mobile query can be aggregated with other mobile queries currently in the system. Two mobile queries can be aggregated only if they are intersectable. If they can be aggregated, we then take a constraint on box size for the two queries. In line 9, the system verifies if the number of real mo-

bile user request (*real_user_request_cnt*) in the cloaked region is less than the privacy requirement (K) of the mobile user that submitted the request. If *real_user_request_cnt* is less than K, the proposed algorithm generates the remaining request as realistic diverse dummy request. Finally, in line 12, the request is sent to the LBS for processing. The LBS processes the request and send the response to the AS, the AS then filters the response and sends the results to the client.

/ *Algorithm 3: CloakedK */

1. **pre-condition**: *request!null*
2. **input:** *request <u_id,msg_num {t,x,y},{dx, dy, dt},P,C>*
3. **method**:
4. *hash (u_id ,msg_num)*
5. *K = transformation(P)*
6. *createGrid(request.K, request.dx, request.dy, request.dt)*
7. *insertRealUser()*
8. *intersectAndMerge()*
9. ***if*** *(real_user_request_cnt < K)*
10. *insertDummies()*
11. ***endif***
12. *sendRegionRequestToLBS()*
13. **end**

6.3 Complexity analysis discussion

In this section, the asymptotic complexity of the algorithms is discussed. Let n be the total number of requests in the system and K be the privacy requirement of a single mobile user submitting a new location based request.

In the case of CloakedK, the time complexity is bounded by the number of requests in the system. In line 8 of CloakedK, a linear search is performed to find at least K-1 other requests that are compatible with this incoming request. After this linear search if K-1 other requests cannot be found, then the outstanding requests are generated as dummy requests. Hence, the worst case complexity of CloakedK is O (n+(β *ε)), where ε is the time to generate a single dummy requests, and β is the number of outstanding requests to be generated.

In the case of CloaklessK, a search for other requests is not performed. Instead, the CloaklessK algorithm quickly generates K-1 other dummy requests instantaneously. Therefore, the complexity of CloaklessK is bounded by the time that it takes to generate the K-1 dummy requests (i.e. O((K-1) ε)). Consequently, CloaklessK should have a much better run time performance than CloakedK, especially for small K, and for large values n.

6.4 Proof of correctness for MobiPriv in snapshot queries

We will show that MobiPriv can satisfy any privacy requirement. By induction on the size of K, we will show that the proposed algorithm stays with the optimal solution, hence can guarantee high levels of location based privacy.

Let S = {r_2, r_3, r_4,… r_{k-1}, r_k} be the set of requests that are chosen to be anonymized with r_1 by the proposed algorithm for a privacy requirement K. Also, let T = {r_2', r_3', r_4'…r_{k-1}', r_k'} be the set of requests chosen by the optimal solution to be anonymized with r_1. That is, for the request r_1, the optimal solution chooses the requests r_2', r_3', r_4'…r_{k-1}' and r_k'.

For the case of K=1, the mobile client does not wish for their request be anonymized with other requests. In this case, both the optimal solution and the proposed solution would submit request r_1 only to the location based server. Assume true for the case of $K \geq 1$, that the proposed algorithm can satisfy a privacy requirement up to and including K (*inductive hypothesis*). Thus, for a privacy requirement of K, the hypothesis assumes that {r_1, r_2, r_3, r_4, … r_{k-1}, r_k} is satisfied for a snapshot. Now, for a privacy requirement demand of K+1, the optimal chooses a new request r_{k+1}' to be added to T. As a result, if we added r_{k+1}' to the set S (i.e. {r_1, r_2, r_3, r_4…r_{k-1}, r_k}) that was chosen by the proposed work according to the hypothesis, the privacy requirement of K+1 is satisfied correctly. However, for the $(K+1)^{th}$ privacy demand, the proposed algorithm will generate a new dummy request r_{k+1} to satisfy the privacy requirement. Therefore, the inclusion of r_{k+1} to {r_1, r_2, r_3, r_4,… r_{k-1}, r_k} (i.e. *inductive hypothesis*) will satisfy the privacy requirement of K+1.

7 Evaluation

We performed experimental evaluations of the proposed algorithms (*CloakLessK* and *CloakedK*) against the three PrivacyGrid approaches (*Bottom Up, Top Down, Hybrid*) [15] and also against the *pyramid* based approach such as Casper [19]. First, the three PrivacyGrid techniques [15] are briefly discussed. In the *bottom-up* cloaking the mobile user cell is expanded to meet the K-anonymity and diversity requirement. Mobile users are considered for the K-anonymity count and static objects e.g. gas stations, supermarkets are used for the diversity count. For a given cell it may expand to its immediate neighbor east, west, north, or south. The next cell to be chosen is the cell with the highest mobile user count. This cell will be included in cloaking box. The *top-down* approach first selects the largest possible cloaking box that satisfies the users QoS requirement. If *K* users cannot be found in this cloaking box the query cannot be satisfied. If *K* mobile users are found, the algorithm will attempt to prune the cloaking box to see if *K* can still be satisfied. If K can still be satisfied in a smaller

cloaking box then the latter and smaller cloaking box is chosen. The *hybrid* approach makes a decision to use the bottom up, or top down, depending on the value of *K* and the QoS. Hybrid combines the strength of both (top down and bottom up) approaches.

The pyramid based scheme used [19] is now discussed. In [19], for anonymization and cloaking a pyramid structure is maintained. The cells in the region maintain the number of mobile users present in each cell. If the current cell of user cannot satisfy the value of *K*, then neighboring cells are considered. A cell is considered a neighbor if they have the same parent. If expansion to neighbors does not satisfy the user requirements, then the parent expansion is considered. One can envision a balance quad-tree where the users are leaves in the system.

7.1 Evaluation criteria

In this Section, the evaluation criteria that we used to evaluate the efficiency and effectiveness of *MobiPriv* and previously proposed algorithms are discussed.

7.1.1 Success rate

One of the most important evaluation criteria is the *success rate*. The main goal of any anonymization server is to maximize the number of messages that can be successfully anonymized with the personalize quality of service and privacy requirement desired. The *success rate* is measured as the ratio of the number of successful anonymized request, by the total number of incoming mobile request.

A success rate of 100% implies that all the requests that are sent by the mobile clients are safely anonymized. In some systems [8, 15, 19, 29], the request is dropped because the privacy requirement cannot be satisfied. Let *N* be the total number of mobile requests send to any cloaking algorithm *CloakM*. Then, the set of mobile requests that can be successfully anonymized with the personalized quality of service and privacy requirement can be calculated as $\{m_t \mid m_t = CloakM(m_s), m_s \in N\}$ where m_s is the request sent to $CloakM(m_s)$. The success rate of any algorithm $CloakM(m_s)$ is given by:

$$Success\ rate = \left(\frac{|\{m_t \mid m_t = CloakM\ (m_s \in N)\}|}{|N|}\right) * 100$$

7.1.2 Performance measure

The run time performance of the algorithms is measured as the cloaking time. The cloaking time of an algorithm is the time taken to perturb and privatize the mobile requests. An algorithm with a lower cloaking time does better, because the cloaking time is a measure of the temporal complexity. Efficient cloaking

implies that the algorithm spends less time processing the incoming mobile requests from the mobile clients. We define a function *startTime (cloaking Algorithm)*, that returns the time the cloaking algorithm start the anonymization process. Also, *endTime (cloakingAlgorithm)*, which returns the time the cloaking algorithm completes the anonymization process.

$$cloaking\ time = endTime(CloakM(m_s)) - startTime(CloakM(m_s))$$

7.1.3 Communication cost

We were interested in measuring the communication cost of the proposed algorithm and previously proposed algorithms. The communication cost is a measure of the number of messages sent by the anonymization server to the LBS. For example, in CloakLessK, for each request received from the mobile client, the algorithm immediately aggregates that request with dummy requests and then forwards the aggregated request (i.e. region request) to the LBS. Therefore, if N requests are submitted to the anonymization server, then we also have N region requests being submitted to the LBS. In the case of other cloaking algorithms such as Casper [19], CloakedK, or PrivacyGrid [15], for N requests sent to the anonymization server, the anonymization server may send less than N region requests to the LBS. In these schemes, multiple real requests can be aggregated together and forwarded to the LBS. Let N be the total number of mobile request submitted at time *t* by the mobile clients. The number of *region requests* submitted by the anonymization server (AS) for the N incoming request is *m*, *1<=m<=N*. We measure the communication cost as the ratio:

$$communication\ cost = \frac{m}{N}, 1 \leq m \leq N$$

7.1.4 Quality of service

Some QoS evaluation variables considered are *spatial tolerance* and *anonymity level*. Spatial tolerance is the user defined spatial resolution that should be satisfied in conjunction with the anonymity level. The anonymity level is the user defined K-anonymity requirement.

7.2 Experimental setup and road network

The experiments are conducted on a Windows machine running the P8400 Intel DUO 2.27 GHz processor with 4GB of RAM. The six algorithms (Bottom up PrivacyGrid, Top down PrivacyGrid, Hybrid PrivacyGrid, Casper Pyramid approach, MobiPriv CloakLessK, MobiPriv CloakedK) were implemented using

Java. We refer to these algorithms as *B*, *T*, *H*, *Py*, *CLK*, and *CK* respectively in the experiments.

The mobile object generator that was considered is an extension of mobile object generator used in [8, 15]. A map of Chamblee in the state of Georgia USA was used for the experiments. The map covers a region of 160 km², see Figure 4. Moving object traces were generated based on real world traffic volume data extracted from [10] for 10,000 cars traveling along the road network. Three types of roads are considered in the simulation; expressway, arterial, and collector roads (see Figure 4). Cars are placed randomly on the road network initially, and then continue to move along a road trajectory making a decision at each intersection. The properties of each road type considered in the experiment are shown in Table 1. Each car (mobile user) generates multiple requests to the anonymization server (AS) running MobiPriv algorithms. The anonymization server then anonymizes the request before forwarding it to the service provider (LBS). The experimental results using the previously described evaluation matrices are discussed next in Section 8.

TABLE 1- Road Properties

Properties	Road categories		
	Expressway	*Arterial*	*Collector*
Mean speed (km/h)	90	60	50
Std. Dev (km/h)	20	15	10
Traffic volume (cars/h)	2916.6	916.6	250

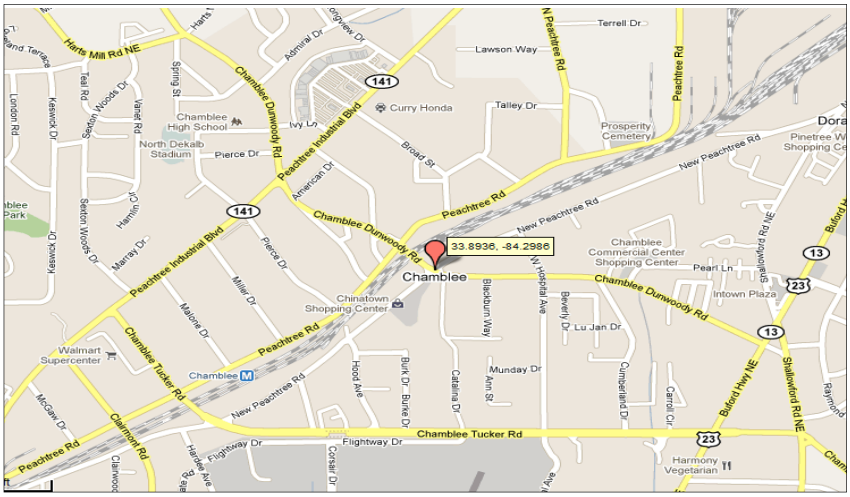

Figure 4 -Evaluation area (Map of Chamblee Region, Georgia, USA)

8 Experimental results

The experimental results for the proposed algorithms compared against previously proposed algorithms are presented in this section. The algorithms' effectiveness and efficiency under varying user requirements are studied.

8.1 High privacy requirement and high spatial tolerance

For these experiments, the algorithms' performance when the mobile user demands a high privacy level and a high spatial tolerance is studied. A high spatial tolerance implies a low quality of service. In Figure 5, 6, and 7 we evaluate the success rate, cloaking time, and communication cost with *high spatial tolerance* (i.e. 700m*700m) and *high anonymity level* (i.e. K=50). Figure 5 plots the success rate with varying number of mobile requests sent to the anonymization server. The x- axis represents the number of mobile request sent by the mobile object generator to our anonymization server. The y-axis represents the success rate (see Section 7.1.1) of the algorithms in percentage. We saw that *MobiPriv* algorithms achieved 100% success rate. This implies that all the requests sent can be safely anonymized with the *MobiPriv* algorithms.
The PrivacyGrid (top down, bottom up, hybrid) [15] and Pyramid approaches [19] only anonymized 70%-80% of the mobile requests sent to the anonymization server. PrivacyGrid and Pyramid schemes will drop some of the requests because the QoS demanded by the client cannot be satisfied. An increase in the number of mobile requests sent to the anonymization server showed a slight increase in the success rate of the other algorithms (*B*, *T*, *H*, *Py*). This make sense, since there are more requests in the system, it is easier to locate K-1 other request to anonymize with even under tighter constraints.

Figure 6 expresses the run time performance against increasing number of mobile requests sent to the anonymization server. We refer to the run time performance as "cloaking time" (see Section 7.1.2). *MobiPriv* algorithms achieve fast cloaking time. In particular, *CloakLessK* has a lower cloaking time than *CloakedK* since it can quickly generate K-1 dummy requests instead of searching for real requests initially. The fastest cloaking time under these settings is the pyramid based in [19]. Also, MobiPriv algorithm's cloaking time is hardly affected by an increase in the number of mobile requests. On the other hand, the cloaking time of the PrivacyGrid schemes are severely affected by an increase in number of mobile user request. As the number of requests increases from 200 to 1000, the cloaking time of the PrivacyGrid schemes [15] increases rapidly. This is not the case for the proposed MobiPriv algorithms.

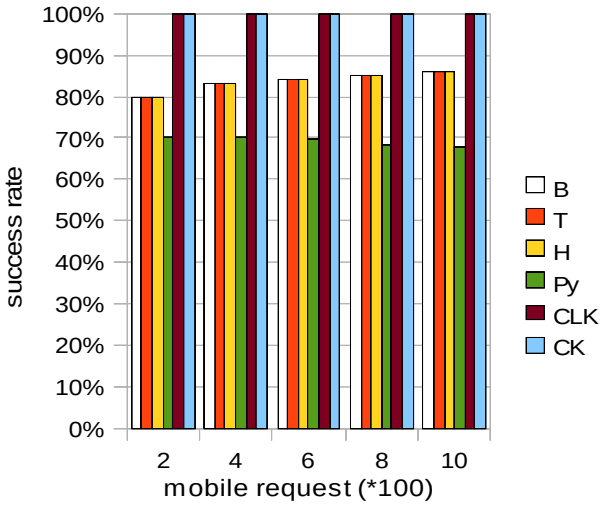

Figure 5 – success rate and number of mobile requests

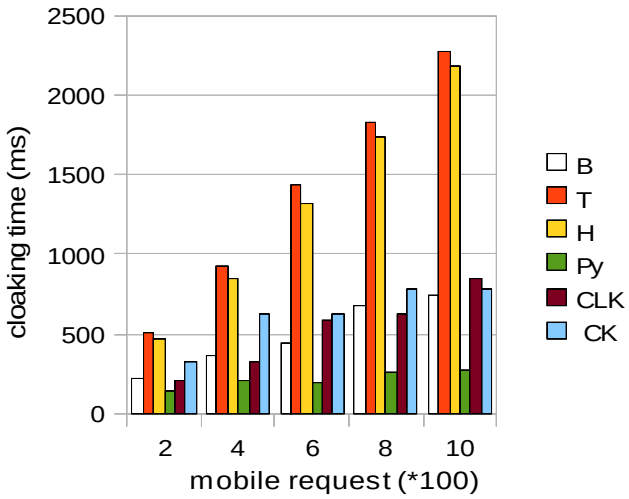

Figure 6 – cloaking time and number of mobile requests

In Figure 7, the graph depicts the communication cost (see Section 7.1.3). We saw that MobiPriv algorithms have a higher communication cost. More specifically, CloakedK does better than CloakLessK and continues to do much better as the number of request increases.

We conclude the discussion of high anonymity level and high spatial tolerance by claiming that MobiPriv algorithms guarantee that all messages can be anonymized safely at a relatively fast speed for high anonymity level and high spatial tolerance. MobiPriv algorithms on the other hand have higher communication cost. *MobiPriv CloakedK* has a better communication cost than *CloakLessK*. Additionally, the other trusted third party schemes such as PrivacyGrid [15] and Pyramid based [19] dropped 20%-30% of the user requests. In these schemes, if the privacy requirement and QoS cannot be satisfied, the request is dropped. MobiPriv algorithms on the other hand are a guarantee service, they will never drop a request. Instead, MobiPriv always achieves 100% success rate.

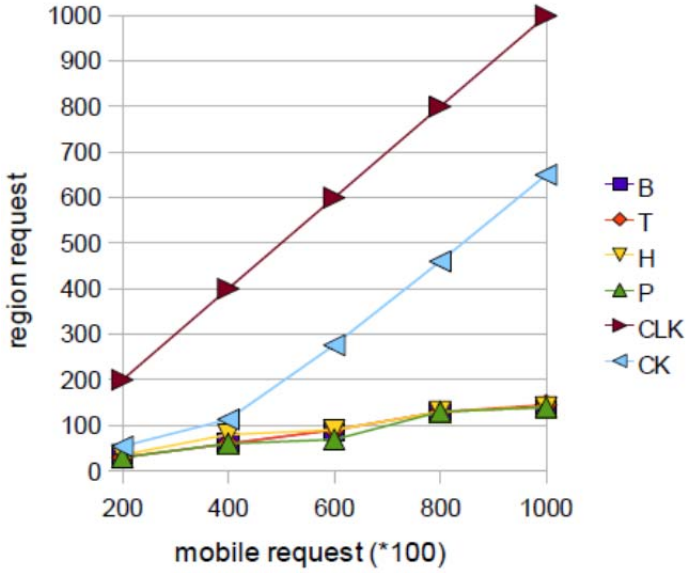

Figure 7 communication cost and number of mobile requests

8.2 Lower privacy requirement and high spatial tolerance

For these experiments, the algorithms' performance when the mobile user demands a lower privacy level and a high spatial tolerance is studied. We evaluate algorithms' success rate, cloaking time, and communication cost with *low average anonymity level* (K=10) and *high spatial tolerance* (700m*700m) in Figures 8, 9, and 10. The primary difference between these experiments and the previous experiment (Figure 5, 6, and 7) is the fact that the privacy requirement is now *K=10* instead of *K=50*.

Figure 8 plots the success rate with different number of user requests. In general, all cloaking algorithms should have a higher success rate with a reduction in the average anonymity level (i.e. from K=50 to K=10). *MobiPriv* algorithms still anonymized all the requests. This is not the case for the other algorithms (B, T, H, Py). However, the three PrivacyGrid and the Pyramid based systems showed an improvement in the success rate compared to Figure 5. This improvement in success rate is because it is less challenging to discover a smaller number (*K=10*) of requests to cloak with than K=50. Pyramid scheme anonymizes 98.5% of the total request received, this indicates that a few of the requests are still discarded. MobiPriv algorithms achieve 100% success rate, it does not discard requests.

In Figure 9, the graph shows the cloaking time (run time performance) on the y-axis with increasing number of mobile request on the x-axis. The proposed *MobiPriv* algorithms have good cloaking time. In particular, *CloakLessK* has an

exceptionally fast cloaking time for low anonymity level (e.g. K=10) since it can quickly generate dummies. While the run time performance of the other schemes, such as PrivacyGrid's top down and hybrid approaches is very high.

From Figure 10, all the privacy algorithms submit more region requests to the LBS with an increase in the number of incoming requests. *CloakLessK* communication cost is not affected by a reduction in anonymity level. We end the discussion of the algorithms under low anonymity level and high spatial tolerance by concurring that MobiPriv algorithms still maintain a 100% success rate and also the fastest cloaking time. However, we pay a price for communication cost.

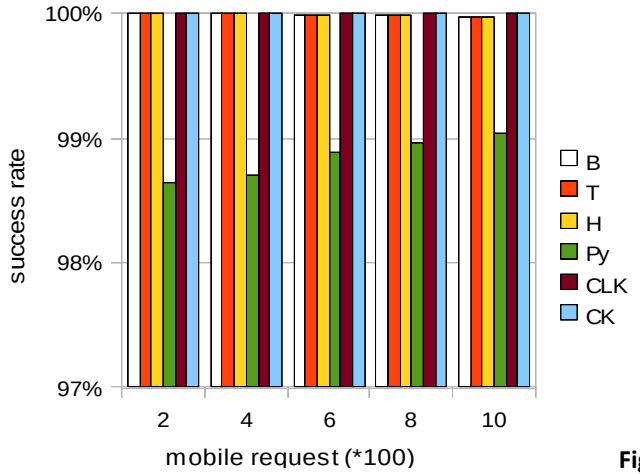

Figure 8 – success rate and number mobile requests

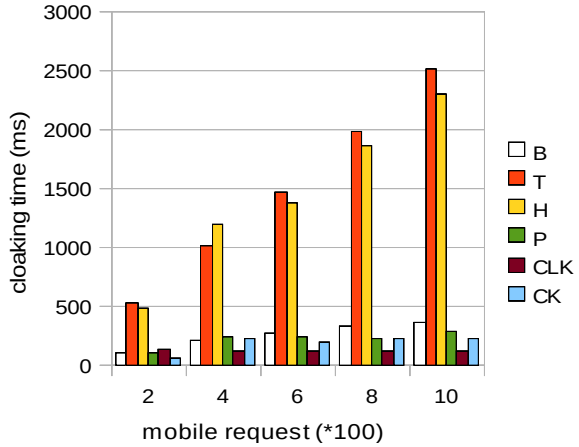

Figure 9 – cloaking time and number of mobile requests

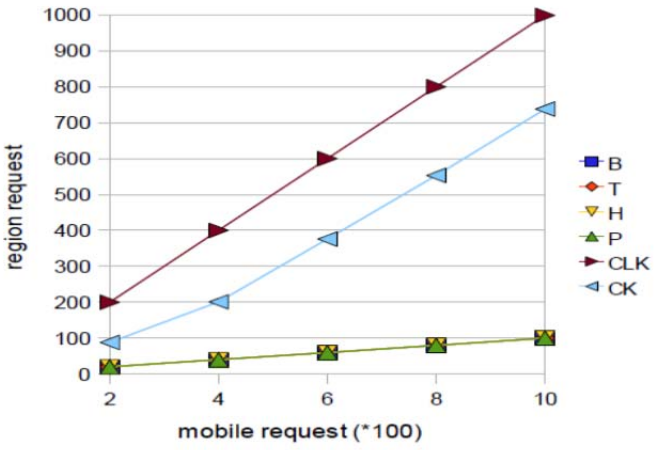

Figure 10 – communication cost and number mobile requests

8.3 Lower privacy requirement and low spatial tolerance

For these experiments, the algorithms' (i.e. B, T, H, P, CLK, CK) performance when the mobile user demands a lower privacy level and a low spatial tolerance is studied. A low spatial tolerance implies a high quality of service. We evaluate the algorithms under *low average anonymity* level (i.e. K=10) and *low spatial tolerance* (100m * 100m) in Figures 11, 12, and 13. This experiment studies the algorithms when the mobile clients demand high quality of service.

Given a low spatial tolerance (i.e. 100m * 100m) and low anonymity level, Figure 11 shows the success rate with different number of request. We make two important observations. First, the success rate of the PrivacyGrid and Pyramid approaches is less than 60%. This implies that these algorithms running on the anonymization server dropped over 40% of the total mobile requests received. This is undesirable, especially in critical location based systems. Secondly, MobiPriv algorithms still maintains a 100% success rate even under such high personalized QoS requirements. This is one of the main strengths of MobiPriv, it never discard a request. The main reason why the previous approaches [15, 19] drops over 40% of the incoming location based requests is because the privacy (i.e. K) and the QoS (i.e. spatial tolerance) requirements cannot be satisfied.

Figure 12 plots the run time performance (cloaking time) with varying number of mobile requests. The most obvious observation is that the pyramid scheme now has a very high cloaking time for low spatial resolutions. The *MobiPriv* algorithms have a fast cloaking time. Specifically, *CloakLessK* does slightly better than *CloakedK.* The runtime of the other algorithms (i.e. B, T, H, Py) are poor if we consider that they only anonymized 60% of the total incoming mobile requests.

Figure 13 highlights the communication cost. It is observed that *CloakedK* has a better communication cost than C*loakLessK.* In general, *MobiPriv* algorithms have a higher communication cost, it ensures that all the messages are anonymized. The other algorithms (i.e. B, T, H, Py) anonymize 60% of the requests. Consequently, the other 40% was discarded, thus did not contribute to the communication cost.

We end the discussion of low average anonymity level and low spatial tolerance by claiming that MobiPriv algorithms maintain a 100% success rate while the PrivacyGrid and Pyramid based approaches achieved 60% success rate. Thus, MobiPriv guarantees a 100 % success rate under high personalized quality of service or privacy requirements. The proposed MobiPriv algorithms have a fast cloaking time that is comparable to any of the algorithms that we studied. Specifically, *MobiPriv CloakedK* does better in communication cost than *CloakLessK*. Additionally, the communication cost of the other schemes (i.e. B, T, H, Py) is better than MobiPriv's.

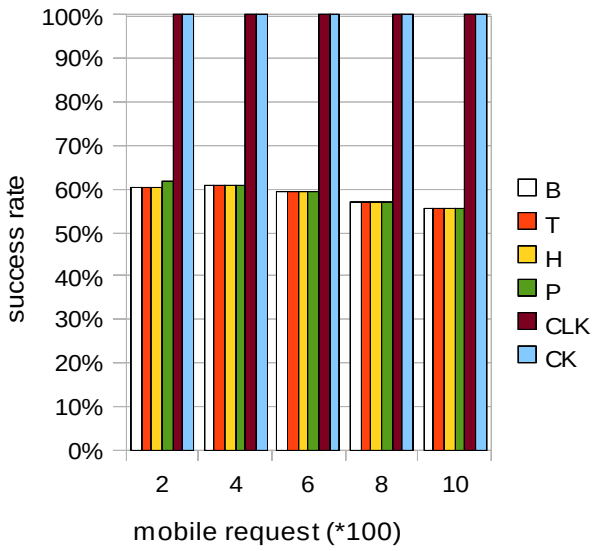

Figure 11 – success rate and number of mobile requests

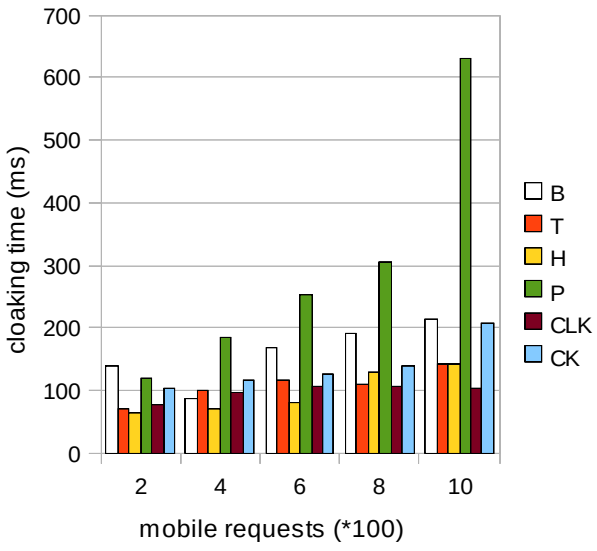

Figure 12 – cloaking time and number of mobile requests

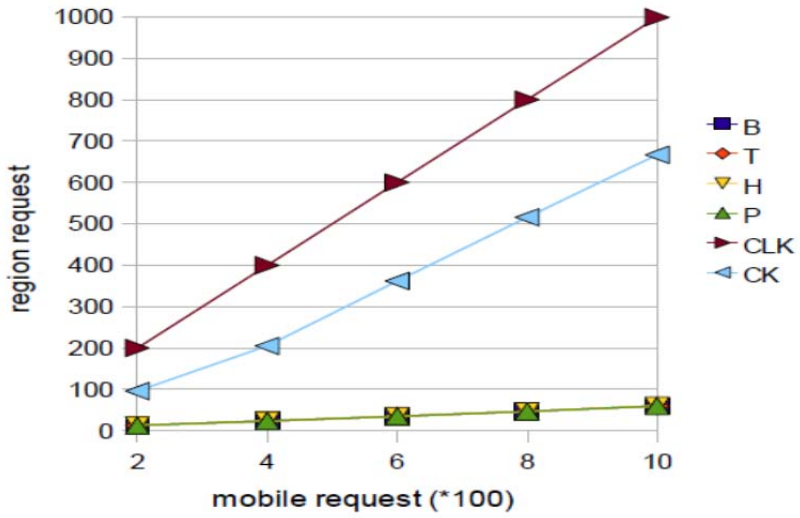

Figure 13 – communication and number of mobile requests

8.4 Privacy and Quality of Service (spatial tolerance)

For these experiments, we study the algorithms' performance under variety of spatial tolerances. In Figures 14, 15, and 16, the effects of spatial resolution (QoS) on the success rate, performance (i.e cloaking time) and communication cost is evaluated. The average anonymity level is set to K=50, and the number of

mobile requests submitted to the anonymization server is 1000. Figure 14 plots the success rate with the personalized spatial tolerance. Again, *MobiPriv* algorithms achieve 100% success rate for any spatial tolerance, even extremely low (i.e. less than 50m) spatial tolerance requirements. The *PrivacyGrid* and *Pyramid* approaches all have very low success rate for low spatial tolerance (see Figure 14). For a spatial tolerance of 50m*50m, the *PrivacyGrid* (top down, bottom up, hybrid) [15] and pyramid based (Casper) [19] dropped over 98% of the total mobile requests sent to the anonymization server. With an increase in spatial resolution to 100m*100m, PrivacyGrid and Pyramid schemes still drop 70% of the total incoming location based requests.

Figure 15 shows the cloaking time with different spatial tolerances. The *MobiPriv* algorithms all have a higher cloaking time because all the mobile requests that are sent to the anonymization server are anonymized successfully, unlike the other algorithms (i.e. B, T, H, Py) that only anonymize a small fraction (i.e. 2%) of the total mobile requests.

Figure 16 highlights the chart of communication cost with different spatial tolerances. From Figure 16, it is obvious that the spatial tolerance does not affect the communication of *CloakLessK*. As the spatial tolerance increases, the communication cost of *CloakedK* decreases. This is related to the fact that more real mobile users can be included in the *region request*. The other schemes, PrivacyGrid and Pyramid, all have low communication cost because most of the requests that they received are discarded.

We conclude the study on the effects of low spatial tolerance on success rate, cloaking time and communication cost. The study showed that under low spatial tolerances (e.g. 50m*50m) PrivacyGrid and Pyramid schemes discard most (98%) of the mobile request received. An example of a low spatial requirement in real life location based system is a transit itinerary system where the user of the system may request the shortest or fastest route from an origin to a destination and does not want to walk more than 50m. Additionally, apart from the high success rate, the MobiPriv algorithms have good run time performance.

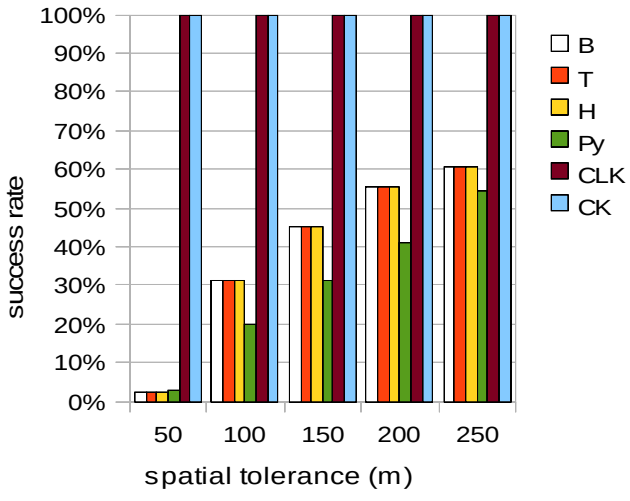

Figure 14 – success rate and spatial tolerance

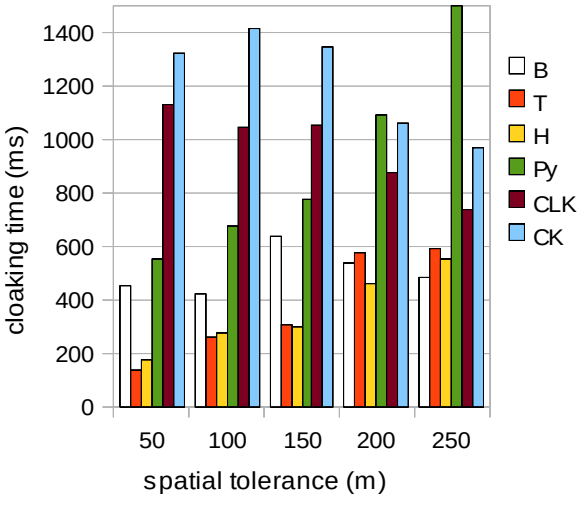

Figure 15 – cloaking time and spatial tolerance

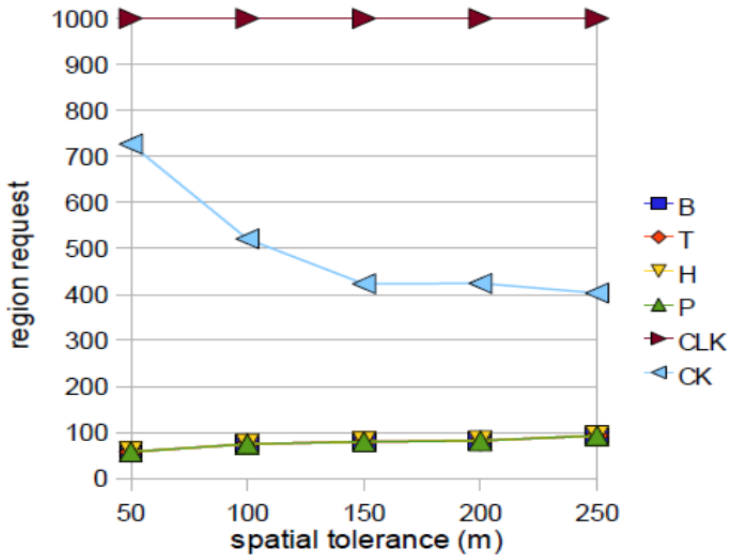

Figure 16 – communication cost and spatial tolerance

9 Continuous queries

In this section, continuous queries are studied. We will show that current snapshot solutions cannot overcome the privacy challenges in continuous querying systems. The proposed technique and experimental results for privacy preservation in continuous location based queries are presented.

9.1 Query linking in continuous queries

Some anonymization techniques that involve a trusted third party AS [8, 15, 19, 26] cannot protect against the corollary history attack. In these models, when a user *u* submits a query to the AS there is a search for K-1 other requests. If the K-1 other requests are not found immediately, the cloaking region is expanded with the intention to locate these other requests. If *u* moves to another location and submits another query, *K-1* other requests have to be rediscovered again. However, the *K-1* requests in the latter region request may be different from the former *K-1* requests. Consequently, taking the intersection of the latter and the former regions, enables the query to be linked to the mobile client *u* in such systems. Furthermore, rediscovery of the same user requests for subsequent requests reduces the QoS as observed by Chow et al [29].

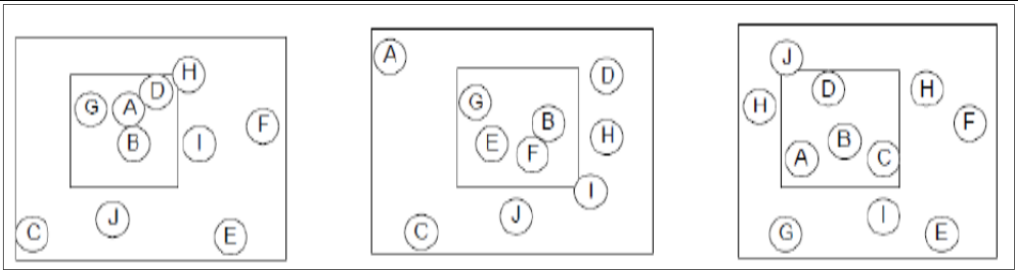

Figure 17 (a, b, c) - Corollary history attack in continuous queries

Consider the following scenario in Figure 17 with 10 mobile clients (i.e. *A, B, C, D, E, F, G, H, I, J*). The queries are submitted by mobile client B and consist of three discrete timestamp readings t0, t1, and t2 (i.e. Figure 17(a), Figure 17(b) and Figure 17 (c)) respectively. The large rectangles in Figures 17(a, b, c) represent the map of the "world" where all the mobile clients reside. The small rectangles represent a region request, it contains the mobile users that are anonymized together. The desired local anonymity level for mobile user B corresponds to K=4 for explanation purposes.

At time t0 (i.e. Figure 17a), the K-anonymity value of 4 submitted by the mobile client *B* is satisfied because mobile client *B* is anonymized with mobile clients *A*, *G*, and *D* as shown by the small rectangle in Figure 17 (a). Also, at time t_1, the local *K-anonymity* of the mobile client is satisfied (*K* =4). However, the adversary may take the intersection of the two snapshots (t0, t1) and conclude that only mobile clients *B* and *G* are present in both snapshots. Hence, the query linking is reduced to ½. Further, at time t_2, if the intersection of all three snapshots (t_0, t_1, t_2) is taken, mobile client *B* will be positively linked to the query. We refer to this query linking in continuous queries as the *corollary history attack*. Many snapshot solutions [8, 15, 19, 26] cannot overcome this category of attack since previous anonymize candidates are not taken into consideration by these models. Then we will show how MobiPriv prevents this kind of attack.

9.2 Discussion on K_{global} and K_{local}

For continuous queries users are expected to define two privacy requirement parameters, K_{global} and K_{local}. K_{local} is the local privacy for each snapshot, and K_{global} is the global privacy across snapshots. In snapshot systems we refer to K_{local} as K.

Given K_{global} and region requests $R_1, R_2 \ldots R_{n-1}, R_n$ MobiPriv ensures that the following holds.

- $|\{R_1 \cap R_2 \cap \ldots R_{n-1} \cap R_n\}| \geq K_{global}$
- $K_{local} \geq K_{global}\ \forall K_{local}$ where K_{local} is the local snapshot anonymity

9.3 MobiPriv prevents corollary history attacks

Let U be the set of real users in the system. For each real user $U_r \in U$, we maintain a dummy profile. This dummy profile associates U_r with a set of *profileCount* amount of dummy users A_r. When a user submits the required privacy level as a part of the request, we associate this privacy level with a value of K in K-anonymity via a phase we refer to as transformation. If the derived of value $K <$ *profileCount* we generate *K-1* dummies such that the *K-1* dummies $\subset A_r$. An example is discussed below.

Assume a user U_r submits a query to the AS with a privacy requirement corresponding to K_{local}=5. Let the privacy profile (dummy profile) of U_r be the set {U_{a1}, U_{a2}, U_{a3}, U_{a4}, U_{a5}, U_{a6}, U_{a7}, U_{a8}} where *profileCount*=8. In the first region request R_1, we maintain the set {U_r, U_{a1}, U_{a2}, U_{a3}, U_{a4}}. Assume U_r submits another query in R_2, with a privacy requirement corresponding to K_{local} =6 we maintain the set {U_r, U_{a1}, U_{a2}, U_{a3}, U_{a4}, U_{a5}}. If U_r submits a third query in cloaking region 3 (R_3) with a privacy requirement corresponding to K_{local} =7, we maintain the set {U_r, U_{a1}, U_{a2}, U_{a3}, U_{a4}, U_{a5}, U_{a6}}.

Observe, if the adversary takes the intersection of $R_1 \cap R_2 \cap R_3$ = {U_r, U_{a1}, U_{a2}, U_{a3}, U_{a4}}. This implies that the user has a 1/5 chance of been identified, regardless if the adversaries can aggregate region requests across multiple regions. Clearly, there is a correspondence between the lowest value of K specified by the user and the chance of been linked to a query in *MobiPriv*.

In general, for region requests R_1, R_2, R_{n-1}, R_n with corresponding privacy requirements K^1_{local}, K^2_{local}, K^{n-1}_{local}, and K^n_{local} the below constraint holds.

$$|\{R_1 \cap R_2 \cap \ldots R_{n-1} \cap R_n\}| \leq min\,(K^1_{local}, K^2_{local}, \ldots, K^{n-1}_{local}, K^n_{local})$$

We prove this by contradiction. Since K_{local} determines the number of elements in R, the intersections cannot contain more than the number of elements in any set. Let $\{R_1 \cap R_2 \cap \ldots R_{n-1} \cap R_n\}$ be the set of regions and $K_{min} = min_{1 \leq i \leq n}\{|R_i|\}$. Let $A = \{R_1 \cap R_2 \cap \ldots R_{n-1} \cap R_n\}$. Suppose to the contrary $|A| > K_{min}$, we know that all elements of A are contained in all sets R_i $1 \leq i \leq n$ by the definition of intersection. Therefore, all sets R_i $1 \leq i \leq n$ have at least $|A|$ elements, which contradicts $|A| > K_{min}$

9.4 Proof of correctness for continuous queries

We will prove by induction on the size of n, where n is the number of region requests, that for any number of requests, MobiPriv can satisfy the global privacy constraints. In previous sections of the paper, we proved that MobiPriv can satisfy the local constraints using induction. For any value of n, the proposed algorithm can guarantee both local and global privacy. Therefore, for n =

$|\{R_1 \cap R_2 \cap \ldots R_{n-1} \cap R_n\}|$ we will show that MobiPriv will preserve global privacy across all the snapshots and also show MobiPriv algorithm is aligned with the optimal solution.

Let S = $\{R_1, R_2, \ldots R_{n-1}\ R_n\}$ denote the set of regions generated by the proposed algorithm, and T = $\{R_1', R_2', \ldots R_{n-1}'\ R_n'\}$ be the set generated by the optimal solution. This is true for the case of n=1. For this case, we only have one region request (i.e. R_1) been submitted by user u_{real}. The optimal solution would have anonymized with any K_{local}-1 amount of different requests if they are available. Likewise, in MobiPriv, we generate at least K_{local}-1 different dummy requests for the first snapshot (i.e. R_1). Thus, K_{local} for the first snapshot is satisfied. For both the optimal and the proposed algorithm, the global privacy is not taken into account since *n=1*.

Assume that it is true for some value $n \geq 1$(*inductive hypothesis*). This implies that the algorithm's first *n* regions have properties of the optimal. That is, the local K-anonymity is satisfied and the global anonymity is satisfied up to and including R_n.

As a result, if we added R_{n+1}' *(i.e. n+1th optimal region)* to T $\{R_1, R_2, \ldots R_{n-1}\ R_n\}$ (i.e. *inductive hypothesis*) we would not violate the local or global constraints. But in the (n+1)th region, the proposed algorithm generate the same set of dummies as in previous regions (i.e. $\{R_1, R_2, \ldots R_{n-1}\ R_n\}$) for the (n+1)th region. Hence, taking the intersection of hypothesis $\{R_1, R_2, \ldots R_{n-1}\ R_n\}$ and R_{n+1} will not violate the local or global constraints.

9.5 Resilience in continuous queries

In this section, MobiPriv's privacy preservation guarantee for continuous queries is studied.

9.5.1 Resilience

This evaluation matrix is a *percentage* measure of the resilience to the corollary history attack that causes query linking in continuous queries. As we explained previously, corollary history attack is present if multiple mobile requests are submitted by the same mobile user. Consider, a mobile user submitting requests U with a privacy level of *K*=5. If for the first snapshot, we get (U, B, C, D, E), where B, C, D, and E are other requests that request U is anonymized with. For the 2nd snapshot we get (U, B, C, G, H). Globally, we have an overlap of three common mobile requests, U, B, and C. Therefore, the resilience to the corollary history attack in this case becomes 3/5 or 60%. Formalized as follows; let the personalized privacy requirement be *K* in K-anonymity. Also, let the number of common clients in the different snapshots be given by the function *intersect (SN1, SN2, SN3…SNn)*, where *SN1* is the first snapshot, and *SNn* is the last snapshot submitted by the same mobile client.

$$\text{resilience} = (|\textit{intersect}\ (SN1, SN2, SN3 \ldots SNn)| / K) * 100$$

In this section, the algorithms' resistivity to the corollary history attack in continuous queries is evaluated. As mentioned previously, some anonymization techniques cannot protect effectively against corollary attacks because these algorithms greedily selects the nearest mobile requests, without any regard for previous anonymized regions. Mobile requests from previous anonymized regions should be used as selected candidates for future anonymized region requests.

9.5.2 Resilience across snapshots

Figure 18 highlights the graph of resilience against the number of queries. The number of mobile requests in the system is 10000, the average K value in K-anonymity is 5 and the average spatial tolerance is 100m * 100m. Recall, for corollary history attacks to be possible, a single mobile user must submit multiple requests at discrete time points to our anonymization server. In Figure 18, the mobile user issues 5 distinct location based requests.

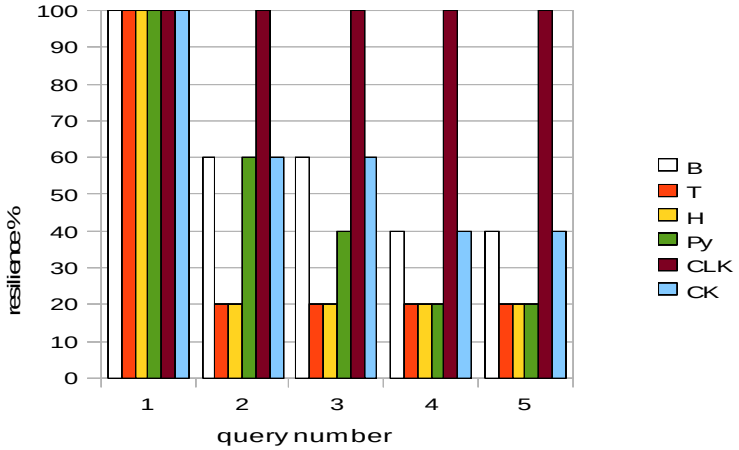

Figure 18 – resilience across five snapshot queries within a continuous query

We measure the resilience of each request using previously anonymized requests as candidates for the current request. From Figure 18, it is observed that for the first query issued, the resilience of all the algorithms is 100% because there are no previously anonymized entrants. In general, CLK algorithm has the best resilience because it will always generate the same set of dummies for mobile requests from the same mobile user. Also, for the other algorithms, the resilience decreases as the number of queries increases since it becomes more chal-

lenging for the intersection of the different region requests to contain the same mobile requests.

9.5.3 Resilience and Quality of Service (spatial tolerance)

Figure 19 depicts the graph of average resilience against spatial tolerance. The mobile client submits 5 mobile requests similar to Figure 18, and then we take the average resilience of the 5 mobile requests under varying spatial tolerance. The number of mobile requests in the system is 5000, the average K value =5, and we vary the average spatial resolution from 50m*50m to 500m * 500m. First, we observe that for high quality of service (low spatial tolerance) such as 50m * 50m, the resilience of *B*, *T*, *H* and *Py* is 0 because the privacy requirement (K=5) and QoS (50m * 50m) desired cannot be satisfied, hence, the query is discarded. More specifically, *CLK* algorithm has the best resilience for any QoS desired. The *CLK* algorithm will never discard a request. Instead, *CLK* will always generate the same set of dummy requests for mobile requests from the same mobile client. *CLK* maintain a 100% resilience against the corollary history attack in continuous queries under any spatial tolerance demanded by the mobile client. For the algorithms *B*, *T*, *H* and *Py*, an increase in spatial tolerance implies stronger resilience against corollary attacks. In general, as the spatial tolerance increases, it becomes easier to locate the same mobile clients that were used as previous anonymization candidates.

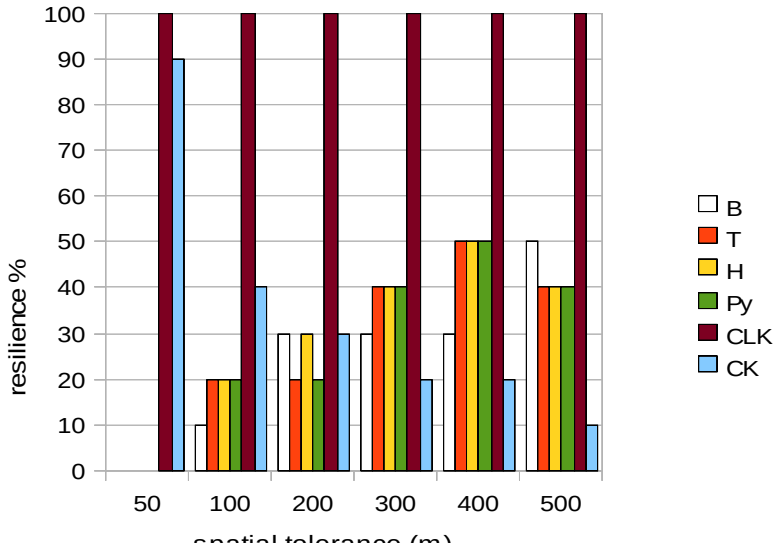

Figure 19 – resilience and quality of service

Additionally, *CK* also achieves high resilience when the spatial tolerance is very low (Qos high). This is attributed to the fact that under low spatial tolerances (50m * 50m) the privacy requirement (K=10) may be difficult to achieve.

However, instead of discarding the request, *CK* will generate the same set of dummy requests, similar to *CLK*. The main difference between *CLK* and *CK* is that *CK* will first make an attempt to anonymize with real mobile requests. However, if there is limited real mobile requests, *CK* generate the same set of dummy requests for mobile requests from the mobile client. On the other hand, *CLK* does not search for real mobile requests. Instead, *CLK* quickly generates the same set of dummy requests for mobile requests from the same client.

9.5.4 Resilience and global privacy requirement

In Figure 20, we evaluated the effects of K on the resilience of the algorithms. We configured *profileCount*=10, this implies that the proposed algorithms (*CLK*, *CK*) will not generate more than 10 realistic diverse dummy requests not exceeding K =10 from the dummy profile (*see Section 4.1*).

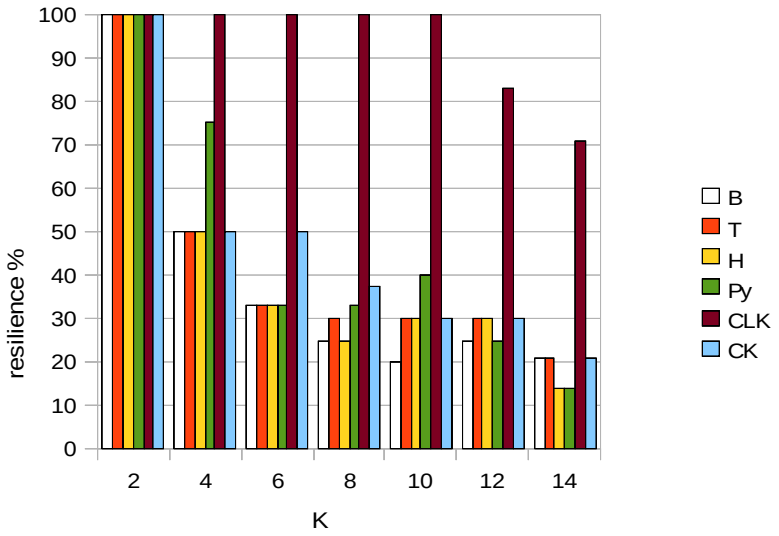

Figure 20 – resilience and K_{global}

For this study, 5000 mobile requests where submitted and the average spatial tolerance is 200m * 200m. First, it is observe from Figure 20 that for very low values of *K* (e.g. *K*=2), all the algorithms provided a high level of resilience to the corollary attacks. In general, as K increases the resilience of the algorithms decrease. As *K* increases, it becomes difficult to locate the previous anonymized candidates. For the *CLK* algorithm, we observe that as the privacy requirement (i.e. *K*) surpasses *profileCount*, the resilience decreases (e.g. at *K*=12, *K*=14). It therefore makes sense in the *CLK* algorithm to ensure that the *profileCount* system parameter setting is configured to a large value.

10 CloakLessK and CloakedK

We introduced the *MobiPriv* suite of algorithms. Both algorithms *CloakLessK* and *CloakedK* are able to achieve the maximum (100%) success rate under any personalized or system wide anonymity level or spatial resolution. This is not the case for previously proposed solution to the same problem. Instead, under certain circumstances as depicted by the experiments, these previously proposed models will underperform.

Both algorithms (i.e. CLK and CK) can guarantee a very high success rate. In terms of run time performance, CLK outperforms CK since there is no search for K-1 other requests in CLK. Instead, CLK can quickly generate K-1 other fake requests.

With regards to communication cost, *CloakedK* outperforms *CloakLessK*. Moreover, we saw that the communication cost of *CloakLessK* is not affected by spatial resolution unlike *CloakedK*. The communication cost of *CloakedK* improves as the anonymity level increases or if the spatial tolerance increases. *CloakedK* does much better than *CloakLessK* as the number of request sent to our middleware increases.

CloakLessK is effective for privacy preservation in continuous queries. The resilience of *CLK* is not affected by the number of continuous snapshot requests sent or the quality of service demanded by the mobile client. *CLK* offers very high resilience to corollary attacks even under increasing privacy requirement (K) by the mobile user. When the privacy requirement of the mobile client surpasses *profileCount*, the resilience is reduced. It is therefore recommended that the parameter *profileCount* be a very large value. For example, *profileCount* could be the maximum K value allowed in the system.

In summary, the proposed algorithms are simple yet effective for privacy preservation in snapshot and continuous querying systems. Additionally, we satisfied both location and query privacy.

11 Conclusion

In this paper, we propose and evaluate a suite of privacy preserving algorithms for location based systems called MobiPriv. MobiPriv is compared against previously proposed privacy preservation algorithms for mobile location based systems [15, 19]. Results indicate that CLK and CK substantially improved the success rate in location based privacy systems. Even with high quality of service requirement, CLK and CK can achieve high success rate. With previously proposed algorithms, under certain conditions, such as high quality of service requirement, or high anonymity level, these privacy systems discard a number of

mobile requests that cannot be anonymized. On the other hand, MobiPriv provides a "guaranteed service", and all the mobile requests sent to our anonymization server can be anonymized. There is also no temporal delay for *K-1* other clients to become available in the proposed work.

Further empirical evaluation showed that *MobiPriv* algorithms have a good run time performance. In particular, for low anonymity level, MobiPriv's *CK* supersedes the other privacy algorithms such as Casper [19] and PrivacyGrid [15] in run time performance. This run time result conforms to the algorithm complexity asymptotic analysis. The communication cost of MobiPriv algorithms is higher. However, a communication cost reduction strategy is used in MobiPriv's *CK*.

Using a dynamic dummy generation strategy that preserves the K-anonymity and diversity requirement, *CLK* and *CK* is effective in a continuous querying environment and guarantees resilience against query linking via corollary history attack. The experimental results indicate that MobiPriv's *CLK* is the best privacy in continuous querying environment. These results also indicate that previously proposed snapshot privacy algorithms, such as Casper and PrivacyGrid cannot overcome the privacy challenges in continuous queries. In general MobiPriv is effective under everyday usage, it guarantees that all queries are successfully anonymized in a fast time.

12 Future work

We intend to deploy the proposed *MobiPriv* system to be used as the middleware for TransitGenie [20]. TransitGenie is a context aware transit itinerary planner for the city of Chicago, Illinois USA. This deployment will provide TransitGenie clients with privacy aware location based trip planning and routing.

13 Acknowledgements

This work is supported in part by NSF through grants IIS-0914934, IIS-0905215, and DBI-0960443.

14 References

[1] The cellular telecommunication and internet association, CTIA Website . http://www.wow-com.com/.

[2] Rakesh Agrawal, Edward Wimmers. *A Framework for Expressing and Combining Preferences.* In Proceedings of the ACM International Conference on Management of Data. SIGMOD, 2000.

[3] Federal Communications Commission http://www.fcc.gov/cgb/consumerfacts/wireless911srvc.html

[4] Sergio Mascetti, Claudio Bettini, Dario Freni, Sean Wang. *Spatial Generalization Algorithms for LBS Privacy Preservation*. Journal of Location Based Services ISSN 1748-9725/ISSN 1748-9733, 2007.

[5] Google Latitude Website - http://www.google.com/latitude/intro.html

[6] Ling Lui. *From Data Privacy to Location Privacy: Models and Algorithms.* VLDB, 2007.

[7] Fuyu Liu, Kien Hua, Ying Cai. *Query l-Diveristy in Location-Based Services.* International Conference On Mobile Data Management, 2009.

[8] Bugra Gedik, Ling Lui. *Location Privacy in Mobile Systems: A Personalized Anonymization Model.* ICDS, 2005.

[9] Pierangela Samarathi, Latanya Sweeney. *Protecting Privacy when disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression*. SRI-CSL-98-04.

[10] Marco Grusteser, Dirk Grunwald. *Anonymous usage of location based services through spatial and temporal cloaking.* ACM/USENIX MobiSys, 2003.

[11] Man Yiu, Christian Jensen, Xuegang Huang, Hua Lu. *SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services*. 24th International Conference on Data Engineering , 2008.

[12] Hidotoshi Kido, Yutaka Yanagisawa, Tetsuji Satoh. *An Anonymous Communication Technique using Dummies for Location Based Services*. Second International Conference on Pervasive Services, 2005.

[13] Tun-Hao You, Wen Peng, Wang Lee. *Protecting Moving Trajectories Using Dummies*. International Workshop on Privacy-Aware Location-Based Mobile Services, *2007*.

[14] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, Kian Tan. *Private queries in Location Based Services: Anonymizers are not Necessary.* SIGMOD, 2008.

[15] Bhuvan Bamba, Ling Liu, Peter Pesti, Ting Wang. *Supporting anonymous Location Queries in Mobile Environments with PrivacyGrid* . World Wide Web, 2008.

[16] Benny Chor, Oded Goldreich, Eyal Kushilevitz, Madhu Sudan. *Private Information Retrieval*. IEEE Symposium on Foundations of Computer Science, 1995.

[17] Eyal Kushilevitz, Rafail Ostrovsky. *Replication is NOT needed: Single Database, Computationally-Private Information Retrieval*. IEEE Symposium on Foundations of Computer Science, 1999.

[18] Chi-Yin Chow, Mohamed Mokbel, Tian He. *Tiny Casper: A Privacy-Preserving Aggregate Location Monitoring System in Wireless Sensor Networks.* SIGMOD, 2008.

[19] Mohamed Mokbel, Chi-Yin Chow, Walid Aref. *The New Casper: Query Processing for Location based Services without Compromising Privacy*. 32nd International Conference on VLDB, 2006.

[20] TransitGenie Website- *Your Personal Transit Navigator* (November 2009) www.transitgenie.com

[21] Chi-Yin Chow, Mohamed Mokbel, Xuan Liu. *A peer to peer spatial cloaking algorithms for Anonymous Location Based Services.* ACM GIS, 2006.

[22] NextBus Website-http://www.nextbus.com/predictor/agencySelector.jsp (2009)

[23] USA Today. *Authorities: GPS system used to stalk woman*. http://www.usatoday.com/tech/news/2002-12-30-gpsstalkerx.htm

[24] Fox News. *Man Accused of Stalking Girlfriend with GPS*. http://www.foxnews.com/story/0,2933,131487,00.html

[25] Anonymous surfing. http://www.anonymizer.com

[26] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, Muthuramakrishnan Venkitasubramaniam, "*L-diversity": Privacy beyond k anonymity.* ICDE, 2006.

[27] PayPal Website - https://www.paypal.com/

[28] Leon Stenneth, Philip S. Yu. *Global Privacy and Transportation Mode Homogeneity Anonymization in Location Based Mobile Systems with Continuous Queries*. 6th International Conference on Collaborative Computing: Networking, Applications and Work Sharing, 2010.

[29] Chi-Yin Chow, Mohamed Mokbel. *Enabling Private Continuous Queries for Revealed User Locations.* International Symposium on Advances in Spatial and Temporal Databases, 2007.

[30] John Voelcker. *Stalked by satellite: An alarming rise in GPS-enabled harassment*. IEEE Spectrum, 2006.

[31] Latanya Sweeney. *K-anonymity: A model for Protecting Privacy.* International Journal on Uncertainty and Fuzziness and Knowledge Based Systems, 2002.

[32] Leon Stenneth, Philip S. Yu, Ouri Wolfson. *Mobile Systems Privacy: "MobiPriv" A Robust K-Anonymous System for Location Based Mobile Systems*. 6th IEEE WiMob, 2010.

[33] Leon Stenneth, Ouri Wolfson, Philip S. Yu, Bo Xu. *Transportation mode detection using mobile phones and GIS*. ACM SIGSPATIAL GIS, 2011.