Privacy-Aware Mobile Location-Based Systems

Leon O. Stenneth
University of Illinois, Chicago
Department of Computer Science
851 South Morgan, Chicago, 60607
lstennet@cs.uic.edu
312.450.1870

Philip S. Yu
University of Illinois, Chicago
Department of Computer Science
851 South Morgan, Chicago, 60607
psyu@cs.uic.edu
312.996.0498

ABSTRACT
Privacy in location-based systems is a major concern, since many mobile phones have a GPS sensor that can report location within 10 meters of accuracy. The contributions of this paper are in three folds. First, we examine privacy issues in snapshot queries, and present our work and results in this area. The proposed method can guarantee that all queries are protected, while previously proposed algorithms only achieve a low success rate in some situations. Next, we discuss continuous queries and illustrate that current snapshot solutions cannot be applied to continuous queries. Then, we present results for our robust models for continuous queries. Finally, we show evaluation results when we add another dimension to privacy in location based systems, referred to as *transportation mode homogeneity*.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database applications, Spatial databases and GIS

General Terms
Algorithms, Experimentation

Keywords
GPS, location-based services, location privacy, query privacy

INTRODUCTION
A GPS sensor is available on many smart phones. Several reports are available where GPS devices are used to stalk user locations [11]. There is also a rapid rise in the number of GPS based harassments [11]. Knowledge of location 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 advertisements sent to his mobile device with unwanted marketing of products or services.

MLBS'11, September 18, 2011, Beijing, China.

Location privacy in location-based systems is to prevent adversaries from learning a mobile user's past or current locations, or the times of the visits. To protect 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. Therefore, if Alice submitted a query via Google Maps requesting the closest parking lot to a psychiatric hospital, revealing her visit to the psychiatric hospital can be embarrassing. Concerning location privacy, Alice would like her exact location, whether in a psychiatric hospital or strip-bar or Mosque, to be hidden. We also want to protect users against request/query linking, by preventing an adversary from knowing the mobile user that has submitted a query. Sensitive queries, such as, "*Where is closest XXX Rehab Center to my current location?*" should not be linkable to the mobile user. In query privacy, the location component inside the query is the quasi-identifier [2]. 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 mobile user request cannot be distinguished from at least K-1 other mobile user requests. The technique is to expand the query location point until we locate K-1 other queries. This way, the exact query locations are hidden. The K value of the message specifies the desired minimum anonymity level. A value of K = 1 means that anonymity is not required for the message. A value of K > 1 means that the perturbed 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 link the message with location/identity association or binding. This certainty is given by 1/K.

Chow and Mokbel [2] observed that while in some systems the location of the user is public knowledge, the queries should remain private [2]. In these systems, from the user location one can infer the mode of transportation (bus, car, walking, running, bike or stationary) since continuous location is reported [7, 12]. The query can often be linked to the user, if the mode of transportation is known. For

example, the query, *"Where is the closest parking lot to the HIV rehab?"* may be related to the driving mode more than bus, walking or running. Hence, the query may be easily linked to the submitter if he is driving**,** and he is anonymized with only non-motorized based requests. For this reason, we add another dimension to privacy called *transportation mode homogeneity*.

To summarize, this paper's contributions are as follows: (1) We address the weakness of previous approaches to location K-anonymity, where requests whose privacy requirement cannot be satisfied are discarded. We introduce a simple and effective method to ensure that all requests can be satisfied with any personalized privacy requirement, in a snapshot environment; (2) We illustrate that current snapshot techniques cannot overcome the privacy challenges in a continuous querying environment. Next, we devise solutions to address the privacy challenges in continuous query systems; and (3) We discuss and evaluate a new extension to privacy preservation in mobile location based systems, referred to as *transportation mode homogeneity*.

RELATED PAPERS

K-anonymity was originally used within the realm of relational databases [13]. The idea of location K-anonymity was first used in [1], where K was static and has the same value for all users in the system. The framework of a personalized value of K was first introduced in [6]. L-Diversity, a second dimension to K-anonymity, was proposed in [10]. In our work, we aim for *transportation mode homogeneity* instead of diversity.

Dummies [8] were proposed by Kido et al. This work is different, we generate dummy requests on the anonymization server with respect to K in K-anonymity. Kido's work in [8] uses the strict client server architecture without the anonymization server. Also, [8] did not consider K-anonymity or L-diversity.

Most of the previous work done on privacy in mobile location- based systems focused on snapshot queries [3, 5, 6]. The work in [3, 5, 6] did not distinguish between location privacy [1] and query privacy [2].The first work that distinguishes between location privacy and query linking privacy in a continuous querying environment is [2]. The authors in [2] addressed continuous queries, but they did not consider *transportation mode homogeneity*. Furthermore, in [2], the algorithm continues to anonymize the same set of mobile clients over the entire continuous set. This enlarges the request region (R_i) and reduces the QoS (quality of service). The proposed work is different from [2] in the way we achieve global privacy and the way we select anonymization candidate sets. In [9], the authors focus on continuous queries, but they rely on entropy as the anonymity measure.

Some prior works did not consider the trusted third party architecture. For example, in [14,8], the authors did not use the anonymization server framework. Instead, they relied on the strict client server architecture. The algorithm proposed in [14] allowed the client to specify a false location called an anchor. In [14], the authors use 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 client will incrementally send more request to the server increasing its supply space and reducing its demand space. When the supply space totally covered the demand space the algorithm completes and the client is guaranteed to have gotten a correct response.

Inferring mobile user's behavior from sensors on the mobile device is an important area of work [7]. In [7, 12], from continuous GPS locations, a user's mode of transportation can be detected with high accuracy. For example, in [12], the transportation mode such as bus, car, or walking was detected with over 80% accuracy. We will show that knowledge of transportation mode by an adversary stimulates query linking. Hence, to prevent query linking in an environment where the modes of transportation may be detected, we introduce *transportation mode homogeneity*.

ARCHITECTURE

This work is focused on the trusted third party architecture [2, 3, 4, 5, 6]. In general, the system consists of mobile devices with positioning capabilities, location based services (LBS), wireless networks, and our algorithms running on a privacy aware middle-ware that we called an anonymization server (AS), see Figure 1. The adversaries loiter between the AS and the LBS.

Mobile Device (Clients): Mobile devices include mobile phones, 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. In addition to location, users also specify the privacy requirement (K) that they desire. Both the personalized privacy requirement and the query containing the location data are forwarded to the anonymization server.

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 alongside the mobile user's query. The role of the anonymization server is to anonymize location and the request before submitting it to the location based system. Recall, 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 AS is assumed to be secured wireless connections.

Service Provider (LBS): The service provider provides location based services to its subscribed mobile users. Upon receiving 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 cloak 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 a request incorporating positioning information such as current location (latitude and longitude) as a parameter of the request to the AS. The AS then hashes any identification fields (e.g. user name), and then cloak the client's query location point into a region containing *K-1* other mobile user requests. The adversaries are not aware of the hash function the AS uses.

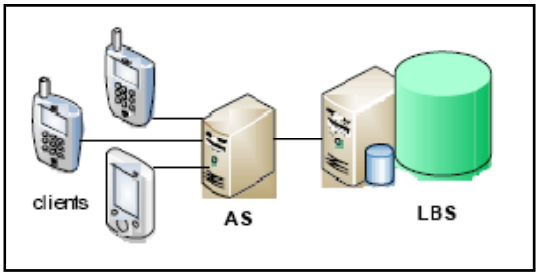

Figure 1- System Architecture

THREAT MODEL

Mobile users communicate precise personal location information, while adversaries' intention is to decipher the physical location of the mobile clients. For example, a person submits a query/request from their house will immediately reveal the mobile client as a residence of the house.

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 has two main goals: (1) correlate a sensitive query to a mobile user, and (2) discover the location that the submitted requests are related to, since location is an identifier in location based systems.

SNAPSHOT QUERIES

A snapshot query is a "one time" request submitted by the mobile user to the AS. For example, "*Where is the closest sushi restaurant to my current location?*" The general anonymization technique is to expand the region around the query location point until K-1 other requests are found [3, 5]. We refer to this region containing the K requests as a region request (R_i). The value of K and the size of R_i (spatial tolerance) are defined by the user at request submission [6].

The main shortcoming of these privacy aware snapshot systems [3, 5, 6] is, if K-1 other requests cannot be found in R_i at the time of the request, 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. 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 approaches in Figure 2 and Figure 3.

Definition 1.1 (Local K-anonymity) - A region request (R_i) satisfies Local K-anonymity (K_{local}) if for every mobile client $m \in R_i$, there exist at least K_{local} -1 other requests $m_1, m_2, m_3, m_4 ..... m_{Klocal-1} \in R_i$ such that any identifier that influences query linking is the same for $m_1, m_2, m_3, m_4 .....m_{Klocal-1}$.

For snapshot queries, Local K-anonymity is required. Additionally, we refer to the privacy requirement K, in each snapshot as K_{local}. We describe two of our approaches that guarantee a very high success rate. The first algorithm is called CLK (CloaklessK) and the second is CK (CloakedK) [4].

In CLK, for each request submitted to the anonymization server (AS), we then generate K_{local}-1 fake/dummy requests on the anonymization server to satisfy the desired K-anonymity. This way CLK can satisfy any privacy requirement (K_{local}), and also ensure that all queries can be anonymized to the required level.

CK, on the other hand, will first attempt to find K_{local} -1 other real requests. Then, if K_{local} -1 other real requests cannot be found in R_i, the remaining requests are generated by our AS as fake/dummy requests.

This work with dummy requests is the first on the trusted third party framework [4]. For evaluation purposes, we compared the CLK and CK against previously proposed privacy solutions in the snapshot environment.

For simulation, we extended the mobile object simulator from [6] to produce mobile users moving on a road network. The users travel along the road network and make a random decision at each road intersection. Each user submits location based requests revealing some location of interest; they also define their privacy as K_{local}, and QoS requirements as spatial tolerance (i.e. size of R_i). The default settings for the simulator are: K_{local} = 10 and spatial tolerance = 200m* 200m. See Appendix for the details on the experiment setup.

Specifically, we compared our work against the approaches in [3, 5]. To validate the effectiveness of the proposed algorithms, we evaluated the success rates, spatial tolerance (size of R_i), and privacy requirement K_{local}. In the experiments, we referred to the Bottom Up, Top Down and Hybrid privacy approach in [3] as B, T and H respectively. Also, we refer to the Pyramid Casper in [5] as Py.

Figure 2 plots the success rate with varying number of mobile requests sent to the anonymization server at each instance. The success rate is defined as the fraction of requests that can be successfully anonymized to satisfy the privacy requirement. The x-axis represents the number of mobile requests sent to the anonymization server. The y-

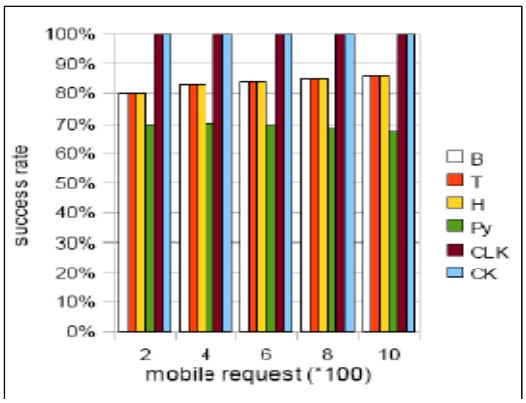

Figure 2 - success rate evaluation, K_{local}=50

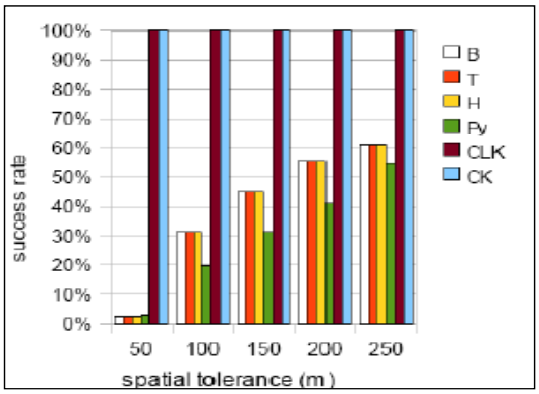

Figure 3 - success rate and spatial tolerance, K_{local}=50

axis represents the success rate of the algorithms in percentage. The average privacy requirement, K_{local}, was configured to K_{local}=50. This is a large value of K_{local}, in [15] it was observed that most people are satisfied with K=5 (i.e. K_{local} = 5). We want to study the behavior of the algorithms when the clients demand a high privacy level.

We observe in Figure 2 that CLK/CK algorithms achieved 100% success rate. This indicates that all requests sent can be safely anonymized with these algorithms. On the other hand, PrivacyGrid's (top down T, bottom up B, hybrid H) and Casper's Pyramid (Py) approaches can only anonymize 70%-80% of the total mobile requests sent to the anonymization server. PrivacyGrid and Pyramid schemes will drop some of the requests because the quality of the anonymization service demanded by the requests cannot be satisfied. For example, if a request desires K=50, and the number of requests in the system or in R_i is less than 50, the request will be dropped. We also observe that an increase in the number of mobile requests sent to the anonymization server, shows a slight increase in the success rate of the other algorithms (B, T, H, Py).

Figure 3 highlights the success rate with the personalized spatial tolerance defined by the user (size of R_i). Again, CLK/CK algorithms achieved 100% success rate for any spatial tolerance, even extremely low (e.g. 50m * 50m) spatial tolerance requirement. The PrivacyGrid [3] and Pyramid [5] approaches have very low success rate for low spatial tolerance. At these low spatial tolerances, since K-1 other requests are difficult to find, the query is often dropped. For a spatial tolerance of 50m*50m, the PrivacyGrid (top down, bottom up, hybrid) and pyramid based (Casper) dropped over 98% of the total mobile requests sent to the anonymization server. With an increase in spatial tolerance to 100m*100m, PrivacyGrid [3] and Pyramid [5] still dropped over 70% of the total mobile requests sent to the AS.

We conclude the discussion of snapshot queries by claiming that CLK and CK can satisfy any privacy requirement, and will always have a success rate of 100%. CLK and CK are effective for any spatial tolerance, any privacy requirement, and any number of mobile requests. This is not the case for previously proposed algorithms for snapshot queries, as they drop requests whose privacy requirement cannot be satisfied.

CONTINUOUS QUERIES

A continuous query is submitted at discrete time points by the mobile user. For example, *"Continuously send me gas price coupons as I travel through the inter-state highway I-90?"* Most of the current work focused on snapshot queries [3, 5, 6]. 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 [2].

For continuous queries, users define another parameter $K_{global.}$ K_{global} indicates the number of candidate requests that should be in the intersection of the different region requests ($R_1, R_2, R_3, \ldots R_n$) in the continuous query.

Definition 1.2 (Global K-anonymity) A continuous query (CS) satisfies Global K-anonymity if the intersection of all the region requests $R_1, R_2 \ldots R_n$ in the continuous query, is at least K_{global} . Therefore, $|R_1 \cap R_2 \cap R_3 \cap R_4 \cap \ldots R_n| \geq K_{global}$ and $K_{global} \leq K_{local}$.

In a continuous querying environment, *Global K-anonymity* should be preserved to prevent query privacy breaches. Consider a mobile user U submitting a request Q, with a privacy level of K_{local}=5 and K_{global}=5. If for the first snapshot, we get (R, B, C, D, E), where B, C, D, and E are the other requests that Q is anonymized with. If for the second snapshot we get (R, B, C, G, H). Globally, we have an overlap of three common mobile requests, R, B, and C. So, the resilience to query linking in this case becomes 3/5 or 60%, even though the Local K-anonymity is satisfied.

In the proposed approach, we generate the same set of fake requests to satisfy the privacy requirement. This same set of fake requests is used across the continuous query for subsequent requests from the same user. This way the intersection of regions will reveal at least K_{global} different request for CLK.

Let the number of common requests/queries in the different region requests, within the continuous query, be given by the function *intersect (R_1, R_2, R_3...R_n)*, where R_1 is the first region request, and R_n is the last region request submitted by the same mobile client. We define the evaluation metric resilience below. Resilience gives an indication of the achievable global privacy level in the continuous query environment.

$$\textbf{resilience} = (|\textbf{intersect } (R_1, R_2, R_3 \ldots R_n)| / K_{global}) * 100$$

Since we generate the *same* set of fake requests for each user across all the snapshots (i.e. region requests) in the continuous query, if for a request X, we generated the fake mobile requests W, P, Y, Z in snapshot 1. For subsequent snapshots, we will still generate the same fake requests (i.e. W, P, Y, Z). Hence, we can satisfy any global privacy requirement. Therefore, the intersection of multiple snapshots within the continuous query will reveal at least K_{global}-1 other requests. This approach is simple, yet effective for privacy preservation in continuous queries.

We now discuss a motivational example. In Figure 4, ten mobile requests (A to J) are in the system. The continuous query is related to mobile request B, and consists of three different timestamp readings (t_0 , t_1, t_2). The large rectangle in Figures 4 (a, b, c, d, e, f) represents the map of the region where the mobile requests are related. The small rectangles represent a region request containing the mobile requests that are anonymized together. The desired local anonymity level is K_{local}=4 and global anonymity is K_{global}=2. K_{local} is the Local K-anonymity constraint for each snapshot, while K_{global} is the Global K-Anonymity across all snapshots. For this example, consider the query *"Where is the closest fast food restaurant to my current location on my route to the 'abc' strip bar?"*. This query is sensitive, we refer to this query as Q1.

Additionally, in TABLE 1, the transportation modes for the corresponding mobile users that submitted each requests (A to J). To make our discussion clear to the reader, we separate the observations from the example above into three sections: (1) Transportation Mode, (2) Global Privacy, and (3) Aggregation of Global Privacy and Transportation mode.

(1) Transportation Mode

In continuous queries, multiple location-based queries are submitted at discrete time points [2]. In some systems, the client's location is public knowledge, but the query should be private [2]. In this case, the mobile clients would like the query to be private even though the user's location is public. From different location points, it is possible for the transportation mode to be determined [7, 12]. For example; from location information, one can compute speed and acceleration. With speed and acceleration, we can clearly distinguish if a person is stationary, walking, or driving [7, 12]. Thus, if a person is stationary, and is anonymized with others that are driving, the query may still be linked to the stationary user, because of the queries' context. We therefore require persons with the same mode of transportation to be anonymized together for all snapshots in a continuous query environment.

We refer to the policy of anonymizing with at least K_{local}-1 of the same transportation mode as *transportation mode homogeneity* [16].

Definition 1.3 (Transportation mode homogeneity anonymity) A continuous query satisfies *transportation mode homogeneity* if all the region requests (snapshots) are transportation mode anonymous. A region request (R_i) is transportation mode anonymous iff the region contains at least K_{local} -1 other requests with the same transportation mode.

In Figure 4 (a, b, c), the query of interest is B, if the transportation modes of the mobile clients are known by the adversary, at t_0, the adversary may filter out the mobile clients G and D, since G is on a train, and D is running. Knowing the transportation mode at t_0 will relate the query to either mobile user A or mobile user B since they both are traveling by car (see Table 1). Hence, the linking possibility is now ½. Moreover, at time t_1, the query may be positively linked to mobile client B, since the intersection at t_0 and t_1 produces (G, B), and G is on train.

If at least K_{local} -1 other mobile users in the region were traveling by the same mode of transportation, query linking via transportation mode would have been eliminated.

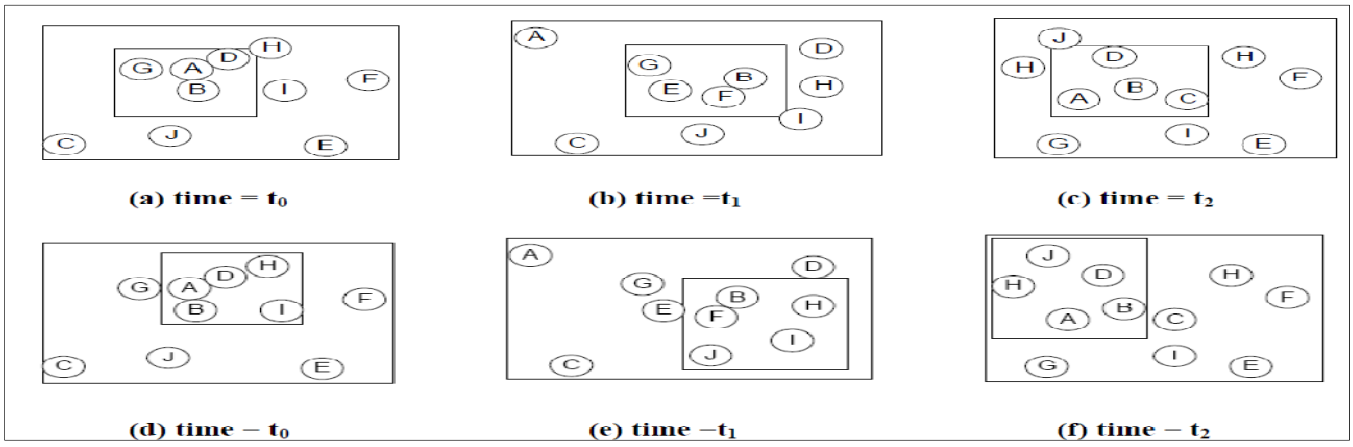

Figure 4-Continuous query and transportation mode homogeneity

Mobile Request	Transportation Mode
A	Car
B	Car
C	Walk
D	Run
E	Bus
F	Stationary
G	Train
H	Car
I	Car
J	Car

Table 1-Transportation modes of request submitter

For clarity on *transportation mode homogeneity*, we discuss Figure 4 (d, e, f). First, observe that in Figure 4(d) the local K-anonymity required by mobile request B (K_{local} =4) is satisfied. Furthermore, the cloaking region at t_0 contains at least three other mobile request with a transportation mode of "car". Query linking is not possible at this stage even if the transportation modes are known.

Second, in Figure 4(e), K_{local} =4 and K_{global} =2 is satisfied, as the intersection of t_0 and t_1 gives mobile clients {B, H, I}. Observe that {B, H, I} is related to car mode. At this point, if the adversary aggregates both t_0 and t_1, the adversary cannot link the query directly to a particular mobile client. Alternatively, the approach could have considered including mobile client A in the region request in Figure 4(e). The inclusion of mobile client A, implies a larger compromise on spatial tolerance. We also take a constraint on spatial tolerance when we attempt to cloak our mobile clients.

Finally, we analyze Figure 4(f), and observe that the required local anonymity of K_{local} =4 is satisfied at the transportation mode level, as proposed in this paper. Furthermore, the global anonymity (K_{global}) by aggregating t_0, t_1, and t_2 produces {B, H} of size 2, both of mode car. Therefore, the request is linked with 50% probability, for global privacy.

Transportation mode homogeneity adds a new dimension to privacy aware mobile location based systems, since knowledge of transportation mode may be ascertained by adversaries from continuous location tracking. This is common in many context aware ubiquitous systems [7, 12]

(2) Global Privacy

Local privacy ensures that each individual snapshot is transportation mode anonymous, with respect to some local K-anonymity value. Global privacy ensures that the aggregation of all the submitted snapshots adhere to some global K-anonymity value.

In Figure 4(a), at time t_0, the local K-anonymity requirement of K_{local}=4 is satisfied. Since B is anonymize with A, G, and D as shown by the small rectangle in Figure 4 (a). Also, at t_1 (i.e., Figure 4(b)), the local K-anonymity of the mobile client is satisfied (K_{local} =4). The adversary may take the intersection of the two snapshots (t_0 , t_1), and conclude that only B and G are present in both snapshots. Hence, the query linking is reduced to ½. Furthermore, at time = t_2 (i.e. Figure 4 (c)), if the intersection of all three snapshots (t_0, t_1, t_2) is taken, B will be positively linked to the query. We refer to this as a reduction of global privacy of the mobile client.

Since there are fewer adversaries with more sophisticated knowledge, it is effective to provide a stronger level of privacy protection against the more common adversaries with weaker knowledge, i.e., maintaining a higher local K-anonymity. An adversary with weaker knowledge may be capable of deciphering only individual snapshots hence higher local K-anonymity is a deterrent. An adversary with sophisticated knowledge may be able to intersect or aggregate multiple snapshots. This will be addressed by the global K-anonymity, which is smaller than the local K-anonymity value.

(3) Global Privacy and Transportation mode

It should be clear to the reader that if we consider transportation mode and global privacy simultaneously, the mobile client may be linked to the query much easier.

The example above highlights two important issues that are the main motivations behind our work. The first issue is, if the adversaries have knowledge of the transportation mode, queries can be linked to the mobile client even if regular (local) K-anonymity is satisfied. The second issue is the reduction of global privacy, if multiple snapshots at different timestamps are aggregated. This reduction of global privacy affects query linking. We therefore introduce an approach that guarantees global privacy and transportation mode granularity anonymization. We aim for *transportation mode homogeneity* instead of diversity.

In CLK and CK, we use dummy requests to prevent query linking. As explained earlier, we generate the same set of dummies for each request. The maximum number of common dummies that can be generated in the continuous query, on behalf of any request, is defined by a system parameter called *profileCount*, which is used to control the extra load to the LBS due to dummy requests. Below, in the experiments, we show how CLK and CK compare in the continuous environment with the prior work. We configured *profileCount*=10, this means the algorithms (CLK, CK), will only generate common fake/dummy requests not exceeding 10.

For experiments, 5000 mobile requests were available to perform K-anonymity. These requests were generated by an extension of the mobile object generator used in [6]. Additionally, the average spatial tolerance (i.e. size of R_i) was set at 200m. See Appendix for details on the experimental setup.

In Figure 5, the x-aixs represents K_{global} and the y-axis denotes the resilience. From Figure 5, it is observed, for very low values of K_{global} (e.g. K=2), all the algorithms provide a high level of resilience to query linking. In general, as the K (K_{global}) increases the resilience of most algorithms decreases. As K (i.e. K_{global}) increases, it becomes difficult to locate the previous anonymization

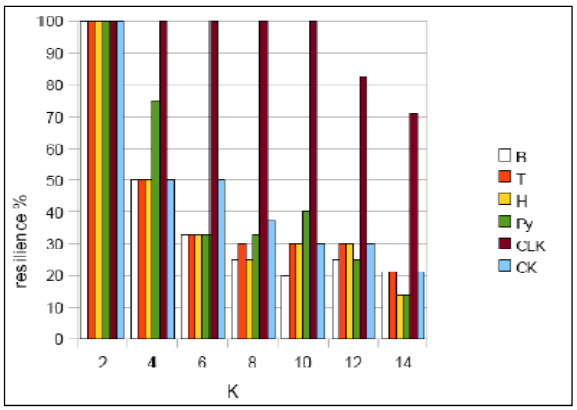

Figure 5- The resilience against query linking

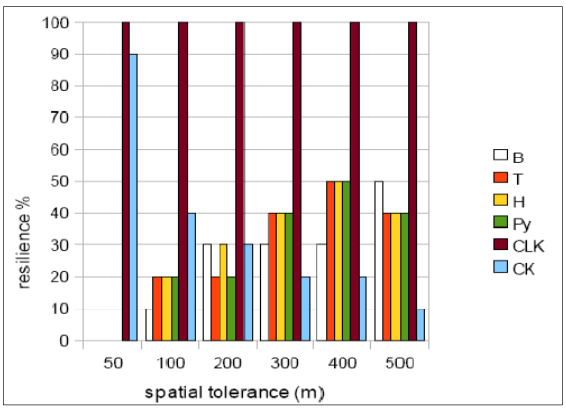

Figure 6 – The resilience against query linking

candidates. For the CLK algorithm, we observed that when the global privacy requirement surpasses *profileCount*, the resilience decreases (e.g. 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. As long as K_{global} is less than *profileCount*, we can achieve perfect query linking protection. The CLK algorithm outperforms all the other algorithms.

In Figure 6, the x-axis represents the spatial tolerance and the y-axis signifies the resilience. From Figure 6, when spatial tolerance is low (e.g. 50m * 50m), the algorithms CLK/CK both satisfied the global privacy. This was not the case for previously proposed algorithms in which the resilience drops to zero. For previously proposed algorithms, under low spatial tolerance, the resilience to query linking decreases.

In general, as the spatial tolerance increases, the resilience in the prior work also increases. This makes sense, as the region gets larger; it becomes easier to locate previously anonymized candidates.

The proposed algorithms are effective for privacy preservation in the continuous querying environment. Even under low spatial tolerances, or high privacy requirement, the global privacy can still be satisfied. The maximum achievable global privacy is controlled by *profileCount*.

Anonymization without Dummy Requests

So far, the anonymization schemes considered, CLK and CK, explore dummy requests. We next consider schemes without utilizing dummy requests.

To evaluate the effectiveness of the schemes, we will consider the most challenging case with *transportation mode homogeneity* on privacy preservation. We developed an algorithm called D-TC (Dynamic Transportation Cloaking). Then, we compared D-TC algorithm to the work in [2], we refer to [2] as R-SC (Robust Spatial Cloaking).

The R-SC algorithm in [2] continues to search for the same set of mobile clients until the query expires. However, the same mobile clients anonymized before may be much further apart in future snapshots. This implies the enlargement of the spatial region to find the same set of mobile clients as before. We note that enlargement of the spatial regions has several implications. First, large regions overwhelm the LBS with load, also reduce the QoS of the results sent back by the LBS.

In D-TC, a novel dynamic layered approach to guarantee global privacy across snapshots is used. The dynamic layered approach separates the local privacy on each snapshot and global privacy across snapshots with different privacy goals, and exploits the local privacy anonymization group as candidates to obtain global anonymization group candidates. It uses a dynamic snapshot suppressing strategy to guarantee global privacy. It will strike a balance between the QoS and the number of snapshots issued for the continuous queries, referred to as the completeness of the continuous queries. The method for selecting anonymized candidates in D-TC is different from R-SC [2]. The D-TC cloaking methodology is a bottom up cloaking strategy [3], where it continues to expand the region around the mobile user that submitted the request until it finds the K_{local}-1 closest mobile requests from the same transportation mode as the request submitter. The region chosen to submit to the LBS will be the bounding rectangle surrounding at least K_{local} mobile clients with similar transportation mode. If K_{global} common mobile clients can be found from the intersection with all previous snapshots, D-TC proceeds to issue the snapshot to the LBS. Otherwise, if K_{global} common mobile clients cannot be found in this snapshot, D-TC considers cloaking with the mobile clients from the previous snapshot that had just satisfied the global constraint. Thus, enlarge the spatial region. A snapshot that cannot satisfy the global constraint will be suppressed, instead of reducing the global privacy. This way D-TC guarantees global privacy. If a snapshot is suppressed, D-TC continues to check subsequent snapshots. For example, if K_{global} is satisfied at R_1, R_2 , R_3 and D-TC cannot satisfy K_{global} at R_4, it suppresses R4 and continues to check subsequent snapshots R_5,.....,R_n. Thus D-TC guarantees global privacy, though it may suppress some snapshots.

We measure the number of snapshots submitted that meet the global privacy requirement divided by the expected number of snapshots as the *completeness* of the privacy algorithms. For example, if the expected number of

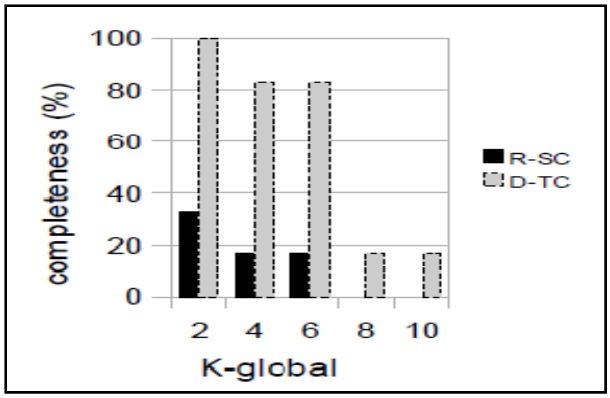

Figure 7 - completeness against K_{global}

snapshots is n=10, and only 8 of the 10 snapshots meet the global privacy requirement and are submitted by the AS, the *completeness* will be 8/10 = 80%. However, the global privacy would still be 100%, since only snapshots that can satisfy the local and global constraints are submitted by the AS to the LBS.

The local and global privacy constraints are dependent on the mode of transportation. Only similar modes should be aggregated together. It should be very clear to the reader that D-TC always anonymize together requests coming from the same transportation mode. R-SC has no regard for transportation mode. Hence, in R-SC, the local or global privacy may not be satisfied.

We evaluated D-TC and R-SC for *completeness*, since it gives an indication of the total achievable privacy across all the snapshots in the continuous query.

In D-TC, the completeness is at least *1/n* since at least the first snapshot will be submitted to the LBS. The AS was configured with K_{local} =10, the number of snapshot queries in the continuous set was 5 (n=5). In addition, 4000 mobile requests with transportation modes randomly assigned from the set {*stationary, walk, run, bus, car, train*}, is available in the system at each instance. The most important observation from Figure 7 is that as K_{global} increase and approaches K_{local} the *completeness* of the algorithm reduces. Furthermore, D-TC has a much higher *completeness* than R-SC. More specifically, for R-SC, as K_{global} increases and approaches K_{local} the *completeness* is zero. This means that none of the snapshots met the global constraints, since they relied on *transportation mode homogeneity* and R-SC has no regard for transportation modes when selecting possible anonymization candidates. For low K_{global}, the completeness of D-TC is high. Note that D-TC does not use dummy requests.

CONCLUSION

We present and evaluate two snapshot algorithms, CLK and CK. We compared them against previously proposed privacy preservation algorithms [3, 5] for mobile location based systems. Results indicate that CLK and CK substantially improved the success rate. Even with high quality of service requirement, CLK and CK achieve high success rate.

We then illustrate that CLK and CK can be used in a continuous querying environment. The results indicate that CLK is the best in continuous querying environment. These results also indicate that previously proposed snapshot privacy algorithms, such as Casper [5] and PrivacyGrid [3], cannot overcome the privacy challenges in continuous queries. We further consider anonymization schemes without utilizing dummy requests. The proposed D-TC method substantially outperformed previous approaches such as R-SC in [2].

Additionally, we contribute and evaluate a new dimension to privacy in location based systems, referred to as *transportation mode homogeneity*. This extension to location based privacy preservation is effective and strengthens K-anonymity. *Transportation mode homogeneity* ensures that mobile users travelling by the same mode of transportation are anonymized together.

In ubiquitous computing, it is more difficult to distinguish between motorized modes such as bus, car, and train. It is easier to distinguish between motorized modes, versus non-motorized modes, such as car, versus walk [7]. We therefore recommend that persons using the exact same mode of transportation be anonymized together. Or, at least, motorized modes and non-motorized modes users to be anonymized separately.

ACKNOWLEDEMENT

This work is supported in part by Google Mobile 2014 Program.

REFERENCE

[1] M. Gruteser, D. Grunwald. Anonymous usage of location based services through spatial and temporal cloaking. *ACM/USENIX MobiSys, 2003.*

[2] C. Chow, M. Mokbel. Enabling Private Continuous Queries For Revealed User Locations. *International Symposium on Advances in Spatial and Temporal Databases, 2007.*

[3] B. Bamba, L. Liu, P. Pesti, T. Wang. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid. *World Wide Web, 2008.*

[4] L. Stenneth, P. Yu, O. Wolfson. Mobile Systems Location Privacy:"MobiPriv" a Robust K-Anonymous System. *IEEE WiMob, 2010.*

[5] M. Mokbel, C. Chow, W. Aref. The New Casper: Query Processing for Location based Services without Compromising Privacy. *32nd International Conferecne on VLDB, 2006.*

[6] B. Gedik, L. Lui. Location Privacy in Mobile Systems: A Personalized Anonymization Model. *ICDS, 2005*

[7] S. Reddy, M. Mun, J. Burke, D. Estrin, M Hansen, and M. Srivastava. Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensor Networks, 2010.*

[8] H. Kido, Y. Yanagisawa, T. Satoh. An Anonymous Communication Technique using Dummies for Location Based Services. *Second International Conference on Pervasive Services, 2005.*

[9] T. Xu, Y. Cai. Location Anonymity in Continuous Location Based Services. *ACM GIS, 2007.*

[10] A. Machanavajjhala, J Gehrke, D. Kifer and M. Venkitasubramaniam. "L-diversity": Privacy beyond k anonymity. *ICDE, 2006.*

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

[12] Y. Zheng. Q. Li, Y. Chen, X. Xie and W. Ma. Understanding Mobility Based On GPS Data. Ubiquitous Computing. *ACM UBICOMP, 2008.*

[13] P. Samarathi. L. Sweeney. Protecting Privacy when disclosing Information: K-Anonymity and its Enforcement through Generalization and Suppression. *SRI-CSL-98-04*

[14] M. Yiu, C. Jensen, X. Huang, H. Lu. SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile *Services. ICDE , 2008.*

[15] http://www.physorg.com/news116603576.html

[16] L. Stenneth, P. 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 Worksharing (CollaborateCom), 2010*

APPENDIX: EXPERIMENT SETUP

For the empirical results discussed, we extended the mobile object simulator from [6] to generate mobile users moving on a map of the Chamblee region of Georgia, which covers an area of approximately 168 km^2. When transportation mode homogeneity is used, the mobile users are randomly assigned transportation modes from the set {stationary, walk, run, car, bus, train}. Furthermore, the mobile users randomly generate location based requests to the AS. On receiving each request, the AS anonymizes the request before forwarding it to the LBS. We measure our evaluation criterions against the anonymized request. Three types of road are considered in the simulator; expressway, arterial, and collector road.

The properties (e.g. mean speed, standard deviation and traffic volume) of each road type are shown in Table 2. We used real world traffic volume data for the Chamblee Region to generate the vehicles on the road. The traffic volume data was taken from [1]. Vehicles are placed randomly on the road network initially, and continue to move along a road trajectory making a random decision at each intersection. The simulator attempts to keep the number of vehicles on each type of road constant with time. Each car generates a set of messages (i.e. requests or queries) during the simulation.

The experiments were conducted on a HP Notebook PC running Windows Vista , and contained a P8400 Intel DUO 2.27 GHz processor with 4GB RAM. The privacy preservation algorithms (Top Down [3], Bottom up [3], Hybrid [3], Casper [5], Robust Spatial Cloak (R-SC) [2], CLK, CK, and D-TC) were implemented in JAVA. A Google Map image of the experiment region (Chamblee, Georgia USA) is shown in Figure 8.

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

Table 2 – Road properties

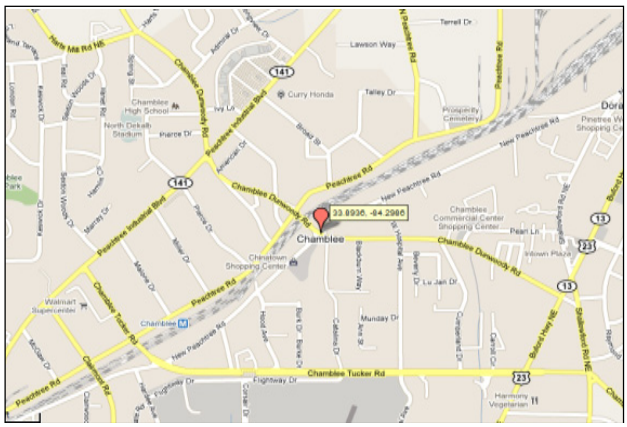

Figure 8 –The experiment region (Chamblee, Georgia, USA)