# LBS (k,T)-Anonymity: A Spatio-Temporal Approach to Anonymity for Location-Based Service Users

Amirreza Masoumzadeh, James Joshi, and Hassan A. Karimi
School of Information Sciences, University of Pittsburgh
135 N. Bellefield Ave., Pittsburgh, PA 15213, USA
[amirreza,jjoshi,hkarimi]@sis.pitt.edu

## ABSTRACT

We propose a location-based query anonymization technique, LBS *(k,T)*-anonymization, that ensures anonymity of user's query in a specific time window against what we call *known user attack*. We distinguish between our technique and related work on $k$-anonymity for LBSs by showing that they target different privacy inference attacks. Also, we analyze the inconsistency of the existing predominant approach with the original definition of $k$-anonymity and its implications on the anonymization. Finally, we present an evaluation framework that assess the applicability and performance of the proposed technique using an evaluation framework.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

## General Terms

Algorithms, Security

## Keywords

privacy, anonymity, k-anonymity, LBS

## 1. INTRODUCTION

Location-based services (LBSs) deal with large amounts of spatio-temporal data related to user movements, among other privacy-sensitive information in user queries. Once collected by LBSs, such privacy-sensitive data are at risk of further analysis for malicious purposes. Although the use of pseudonyms instead of real identifiers may enhance privacy preservation, recent research in the area of data anonymization shows that other pieces of information can also be used to identify user records. In the context of LBS, user's location provided in the queries may be used to link a request, with its user's identity removed, to a user. The idea of employing anonymization for LBSs is to anonymize user queries by cloaking the location area before submitting them to an LBS. The cloaked area is a coarse-grained location information that results in uncertainties, and therefore anonymity, in case an adversary attempts to relate the queries to the users. A query is submitted to a trusted anonymizer that submits an anonymized version of the query to the LBS on behalf of the user, and later relays back its responses.

$K$-anonymity, as one of the principal anonymization approaches [7, 8], has been predominantly adopted by researchers for use in LBSs. It essentially ensures that any linking attack cannot succeed by a probability exceeding $1/k$. Most of the proposed approaches, such as New Casper [6], PRIVÉ [4], and PRIVACYGRID [1], choose a cloaked area as the location context of a query such that there are at least $k$ users in the area at the time of its submission. We shall refer to this approach as *LBS k-anonymity* hereafter. We observe that the lack of complete compliance with the original $k$-anonymity idea may make it difficult for the existing approaches mentioned to provide acceptable anonymity in practice. More specifically, LBS $k$-anonymity neglects to follow a *safe* approach regarding user population in $k$-anonymity (discussed in Section 2.1) that imposes an unrealistic implicit assumption on the adversary's background knowledge. Exception to this is Gedik's et al.'s approach in [3] that strictly follows the original definition of $k$-anonymity. However, its drawbacks include introducing delays in query anonymization. Note that it does not address how to minimize the size of cloaked areas, either.

We note (and show in deatil later) that LBS $k$-anonymity does not provide safeguard against an important type of attack as described in the following. Suppose Oscar knows that Alice has issued a query to an LBS at noon from her workplace. If Oscar has access to the anonymized queries received by the LBS, he can check every query in a relevant time window, say 12pm to 1pm, and identify a subset of queries with cloaked locations that include Alice's workplace; this subset would include Alice's query. As the value $k$ in LBS $k$-anonymity does not control the size of this subset, such approaches fail to provide proper anonymity in case of this attack. Therefore, Alice's query may be identified, either exactly or with a high probability. In this paper, we refer to this attack as *known user attack*, as analogy to the fact that the user is known to have issued a query by the adversary. We believe that such an attack is very likely to occur in practice in the LBS context.

In this paper, we propose LBS *(k,T)*-anonymity as another approach to $k$-anonymity for LBS that has safe as-


ACM GIS '09 , November 4-6, 2009. Seattle, WA, USA (c) 2009 ACM
ISBN 978-1-60558-649-6/09/11...$10.00.

sumptions regarding user population and aims to thwart the *known user attack*. We compare and analyze it against the predominant interpretation of $k$-anonymity in the LBS context. Moreover, We formulate LBS *(k,T)*-anonymization as a spatio-temporal problem, and provide a greedy solution and some experimental results related to its deployment.

The rest of the paper is organized as follows. In Section 2, we present the generally conceived interpretation of $k$-anonymity in the LBS context and analyze its drawbacks and limitations. We propose our approach, LBS *(k,T)*-anonymity, in Section 3 and compare it to the previous approach. We formulate LBS *(k,T)*-anonymization problem, propose a greedy solution for it, and present the experimental results in Section 4. We conclude the paper and provide future directions in Section 5.

# 2. LBS K-ANONYMITY

In this section, we formally present an interpretation of $k$-anonymity in LBSs that is widely captured by the existing approaches such as in [6, 4, 5, 1]. In our formalism, we use the relations $AQ(location, time, query)$ and $UL(user, location, time)$ to represent, respectively, the submitted anonymized queries to the LBS and the exact locations of the potential users. As the LBS is not supposed to be trusted, relation $AQ$ is considered known to the adversary. Moreover, as the worst-case adversary's background knowledge, the exact locations of the users at the time the query is submitted, i.e., $UL^t$ (selection from $UL$ where $time = t$), is assumed to be known. The idea is to assert a query's location area such that at least $k-1$ users other than the one submitting the query are enclosed in the location area. Therefore, an adversary cannot associate a query to a user with a probability more than $1/k$. We formally capture this as follows.

DEFINITION 1 (LBS $k$-ANONYMITY). *Relation AQ is LBS* k*-anonymous iff for every query at a given time there exist at least k users whose locations match the query's location. Formally:*

$$\forall t \forall q \in AQ^t, |\{u \in UL^t | q.location \text{ covers } u.location\}| \geq k.$$

## 2.1 Consistency with Original k-Anonymity

In order to analyze consistency, we provide a brief background on original definitions of $k$-anonymity. Central to the $k$-anonymity principle is the concept of *quasi-identifier*. A quasi-identifier is a combination of a relation's attributes that can be used to uniquely identify at least one individual (while the unique identifier is removed from the relation) with the help of other externally available data sets. $K$-anonymity has been proposed to protect against such a linking attack by proposing the following requirement [7].

DEFINITION 2 ($k$-ANONYMITY REQUIREMENT). *Every combination of values of quasi-identifers must indistinctly match with those of at least k individuals.*

However, as the exact population of individuals that are represented in an external relation is not known to the data anonymizer, a safe approach has been followed to assure $k$-anonymity, captured in the following definition [7, 8].

DEFINITION 3 ($k$-ANONYMITY). *Let P be a relation and QI be the quasi-identifier associated with it. P is said to satisfy* k*-anonymity iff each sequence of values in $P[QI]$ occurs at least k times in $P[QI]$.*

For LBS $k$-anonymity, $AQ^t$ is the privacy-sensitive relation with the quasi-identifier $\{location\}$ (which can be linked to *location* in $UL^t$). We observe that LBS $k$-anonymity captures the $k$-anonymity requirement (Definition 2) by matching at least $k$ user locations in $UL^t$ for every query's location in $AQ^t$. However, it fails to follow the safeguard implied in Definition 3. Note that Definition 3 requires at least $k$ occurrences of each sequence of quasi-identifier in order to rule out any assumptions regarding the population in the linkable external information. LBS $k$-anonymity clearly does not ensure this property. This incompliance to the original definition influences the practicality of the approach as described next.

## 2.2 Adversary's Background Knowledge

The incompliance mentioned above imposes a strong implicit assumption on the adversary's background knowledge that results in an unsafe anonymization scheme. This assumption is as follows: the adversary believes that all the users located in the area enclosed by a query's location at the time of submission are potential issuers of the query. There are two major issues with such an assumption. First, it is very likely that the adversary does not have access to the exact location of every user. Note that collecting such information is not even feasible for a trusted party. Therefore, the anonymizer's belief regarding the population may simply not match the adversary's. Second, in a real-world scenario, an adversary usually obtains her background knowledge through observations. Such background knowledge may help narrow down the set of the actual candidates the adversary needs to consider. For example, an adversary may obtain the background knowledge by sighting a person in a place, or based on the observation that a person is regularly present at workplace or home at specific times.

## 2.3 Known User Attack

As described in Section 1, the *known user attack* is very likely in practice. As LBS *(k,T)*-anonymity does not comply with Definition 3, it remains vulnerable to the attack. An adversary can search relation $AQ$ for a specific time period and location to narrow down to a set of queries related to the victim. In that case, the number of search hits is very likely to be less than $k$.

# 3. LBS (K,T)-ANONYMITY

In this section, we propose a different approach that provides anonymity against the *known user attack*. Here, the adversary is assumed to know that a specific user has issued a query and aims to identify the submitted query in the set of anonymized queries. The adversary's background knowledge includes user's location and also a time window in which the query is believed to have been issued. LBS *(k,T)*-anonymity, formally defined next, ensures that at least $k$ queries' locations enclose an issuing user's location in any time window of at least size $T$ around the time the query has been issued. The time window size $T$ must be chosen smaller or equal to the potential size that an adversary may consider.

DEFINITION 4 (LBS *(k,T)*-ANONYMITY). *Relation AQ is LBS* (k,T)*-anonymous iff for each submitted query at time $t_0$, i.e., $q_0 \in AQ^{t_0}$, issued by user $u_0 \in UL^{t_0}$, there exist at least $k-1$ other queries in the time window of size at least*

$T$. *Formally:*

$$\forall t_1 \forall t_2 (t_1 \leq t_0 \leq t_2) \wedge (t_2 - t_1 + 1 \geq T)$$
$$\implies |\{q \in AQ^{[t_1,t_2]} | q.location \text{ covers } u_0.location\}| \geq k.$$

In contrast to LBS $k$-anonymity, our method does not have implicit assumption on the adversary's background knowledge. There is only the explicit assumption that the adversary knows about the time and place that a query is issued. Also as mentioned, LBS *(k,T)*-anonymity protects against the *known user attack*. It is worthwhile to note that LBS *(k,T)*-anonymity is somehow dual of LBS $k$-anonymity; LBS $k$-anonymity ensures $k$ users for each tuple in $AQ$, while LBS *(k,T)*-anonymity ensures $k$ queries in $AQ$ for each issuing user. Figure 1 depicts a sample behavior of the two approaches for the highlighted, newly-submitted query where $k = 4$. LBS $k$-anonymity sets the cloaked area of the new query such that 4 users are inside it, while LBS *(k,T)*-anonymity ensures 4 queries (including itself) cover the location of the issuing user. We emphasize that the approaches address different aspects of anonymization, and could ideally be employed complementary to each other.

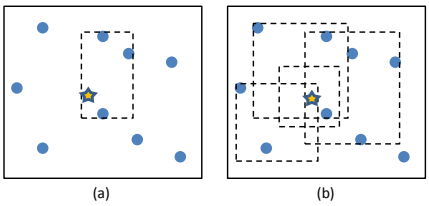


**Figure 1: LBS $k$-anonymity (a) vs. LBS *(k,T)*-anonymity (b)**

# 4. LBS (K,T)-ANONYMIZATION

In this section, we formulate the problem of LBS *(k,T)*-anonymization as per Definition 4, i.e., cloaking query locations such that the location of every user issuing a query is enclosed in at least $k-1$ other anonymized queries in any time window of size $T$ (and greater) that includes the query. Ensuring that the submitted queries to the LBS comply with the LBS *(k,T)*-anonymity principle while performing minimum cloaking for quality of service purpose is a complex spatio-temporal problem.

## 4.1 Problem Formulation

LBS *(k,T)*-anonymization is an optimization problem spanning over both spatial and temporal dimensions. Ideally, the cloaked locations should be optimized not only according to current queries, but also previous and future queries. However, we avoid further complexities by breaking the problem into several iterations. In each iteration, we try to ensure LBS *(k,T)*-anonymity for queries within a time window of size $T$ that ends in the current time point. Figure 2 depicts the time window of size T that ends in the current time point ($t_c$). The queries in the time window can be categorized into two groups: newly issued queries by users at the current time, and the queries issued and processed in the past $T-1$ time points. According to the LBS *(k,T)*-anonymity principle, the location of the issuer of a previously issued query such as $q_x$, that is issued at time $t_i$, should be covered by at

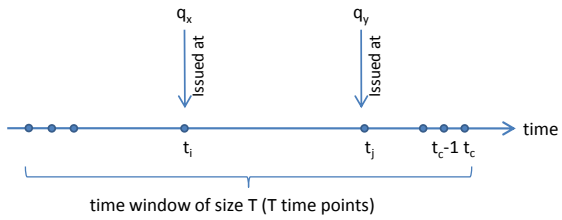


**Figure 2: Time window of size T, ending at current time, that is considered in *(k,T)*-anonymization**

least $k$ query locations. There could be a number of previous queries such as $q_y$, issued at time $t_j$, that cover the issuer of $q_x$. Any remaining coverage for the issuer of $q_x$ towards $k$ coverage needs to be provided by the cloaked locations of the newly issued queries. Analogously, locations of the issuers of the newly issued queries may be covered by locations of the previously issued queries in the time window. The remaining coverage for such issuers should be provided by the newly issued queries themselves. The problem is to determine the cloaked locations for newly submitted queries such that all the coverage requirements are fulfilled, while the total area of such queries are minimized. Iteratively solving this problem at each time point ensures LBS *(k,T)*-anonymity for all queries. We formally define the simplified LBS *(k,T)*-anonymization as follows.

DEFINITION 5 (SIMPLIFIED LBS *(k,T)*-ANONYMIZATION). *Let collection $L$ be the issuers' locations of the newly issued queries; let collections $L'$ and $CL'$ be the issuers' locations and the cloaked locations of the queries issued in the past $T-1$ time points, respectively. The simplified LBS* (k,T)*-anonymization problem is to determine the cloaked locations for the newly issued queries, i.e., mapping $A : L \rightarrow CL$, such that*

- $\forall l \in L, A(l)$ *covers* $l$,
- $\forall l \in L \cup L', |\{cl \in CL \cup CL' | cl \text{ covers } l\}| \geq k$, *and*
- $\sum_{cl \in CL} Area(cl)$ *is minimum, where* $Area(cl)$ *represents the area of the cloaked location* $cl$.

## 4.2 A Greedy Solution

We present a greedy strategy to solve the simplified LBS *(k,T)*-anonymization problem. Due to space limitations, only the general idea of the algorithm is described. The algorithm iteratively expands the cloaked area for the set of newly submitted queries so that every query issuer's location in the past $T$ time points is sufficiently covered. The algorithm is greedy in that it computes every candidate expansion (cloaked area of newly submitted query to cover location of an issuer) and applies the best expansion in each iteration as follows. For every issuer that needs a coverage, the least costly expansion is selected. Among those expansions the most costly one is chosen to be applied. The rationale is that such a choice may end up covering more other issuers also, or at least dramatically reduce the cost of their coverage in future iterations. Note that the ultimate goal is to provide enough coverage for every query issuer anyway. The time complexity of the algorithm is related quadratically to the number of queries in the time window.

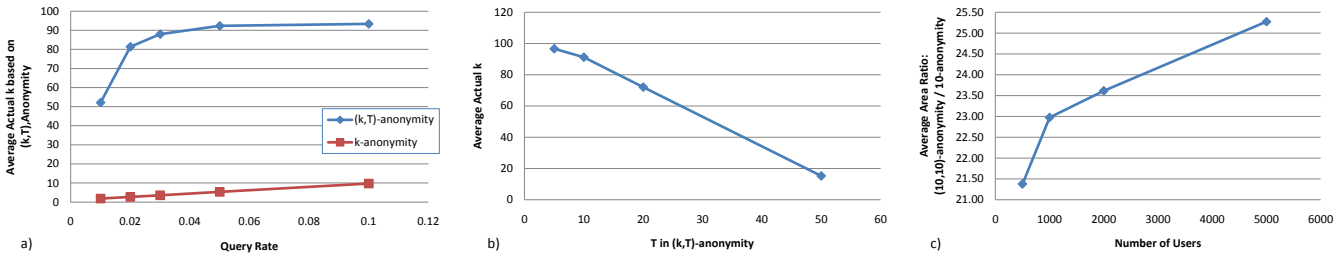


**Figure 3: a) Anonymization performance b) Effect of time window size c) Cloaking performance**

## 4.3 Experimental Evaluation

We have implemented a modular discrete clock-based simulation environment in Java that leverages the *Network-based Generator of Moving Objects* [2] to simulate generation of queries by mobile users on a given road network. In order to enable comparison with LBS $k$-anonymity approach, we have implemented PrivacyGrid [1], a recent work that employs this idea. We have also implemented a grid-based version of the algorithm described in Section 4.2. As the input to the moving object simulator, we used the road network of SF Bay Area (approx. $26k\ km^2$). The area is divided into a grid network of $270 \times 358$ square-shaped cells. We simulated user movements for 100 time units (increase in simulation time did not show any significant effect). Users generate queries with probability $p_q$ with a uniform distribution, that varies between 0.01 and 0.2 at each time point. The number of users vary from 500 to 5000. The parameters $k$ and $T$ were by default set to 10, unless otherwise mentioned. Note that $k$ is the minimum requirement proposed by the scheme. As a performance measure, we analyzed the actual $k$ and the cloaked area sizes based on the anonymized queries.

Figure 3-a shows the actual $k$ measurement, with regards to LBS *(k,T)*-anonymity, on our proposed algorithm and PrivacyGrid. The actual $k$ for all queries was averaged for 1000 users with $p_q$ ranging from 0.01 to 0.1. As expected, LBS $k$-anonymity did not support the proposed $k = 10$ on average. Even when the average actual $k$ meets proposed $k$, not all the LBS $k$-anonymized queries can prevent the *known user attack*. On the other hand, our algorithm, while conforming completely to the principle, seems to provide much larger actual $k$ values than the proposed. This can be attributed to the heuristic-based greedy algorithm that cannot perform good optimization. Setting the $k$ value to lower values may improve the results, but will void the foolproof protection against the *known user attack*. However, our technique shows better performance for larger window sizes. Figure 3-b shows the improvement trend of actual $k$ by adjusting parameter $T$, ranging from 5 to 50 ($k = 10$, $n = 1000$, and $p_q = 0.05$). Note that smaller values for $T$ assumes that the adversary would have more certainty about the time that the victim may have issued a query.

Figure 3-c shows the cloaking performance of LBS (10,10)-anonymization using our algorithm compared to that of LBS 10-anonymization using PrivacyGrid's algorithm. The ratio of the cloaked locations are measured for $p_q = 0.05$ and variable number of users, from 500 to 5000. The results show that the LBS *(k,t)*-anonymization algorithm generates more than 20 times larger areas than PrivacyGrid, which is slightly increased by increasing the number of users. This is partly because of the difficult-to-solve optimization issues related to LBS *(k,t)*-anonymization, compared to the much simpler problem in LBS $k$-anonymization approaches. We emphasize that any direct comparison like this is not very strong as we are dealing with two completely different problems, although in the very much the same context.

# 5. CONCLUSIONS

We proposed LBS *(k,T)*-anonymization that ensures indistinguishability of a user's query among at least $k$ queries in a given time window of size at least $T$. The approach is unique from previous contributions in this area in two ways. It addresses temporal dimension of the anonymization problem in LBS while most previous approaches consider only system snapshots, i.e., a spatial-only problem. Moreover, it addresses a different perspective of $k$-anonymity in LBSs, which is neglected in most of the existing approaches, and has more realistic assumptions about adversary. A future work is to explore the simultaneous enforcement of the two approaches, adding uncertainty to the adversary's background knowledge regarding the issuer's location, and independent parameters $k$ and $T$ per query.

**Acknowledgements.** This research has been supported by the US National Science Foundation award IIS-0545912. We thank the anonymous reviewers for helpful comments.

# 6. REFERENCES

[1] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In *WWW'08*, pages 237–246, 2008.

[2] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.

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

[4] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: anonymous location-based queries in distributed mobile systems. In *WWW'07*, pages 371–380, 2007.

[5] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[6] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new Casper: Query processing for location services without compromising privacy. In *VLDB'06*, pages 763–774.

[7] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[8] L. Sweeney. k-anonymity: a model for protecting privacy. *Int'l Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.