Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2011) 1–10

Procedia Computer Science

An Alternative Approach to *k*-Anonymity for Location-Based Services

Amirreza Masoumzadeh, James Joshi

School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260, USA

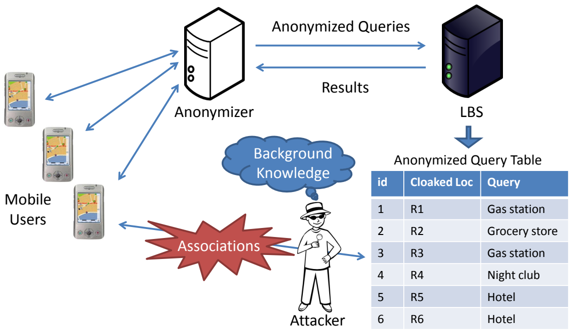
Abstract

Users of location-based services (LBSs) may have serious privacy concerns when using these technologies since their location can be utilized by adversaries to infer privacy-sensitive information about them. In this work, we analyze the mainstream anonymity solutions proposed for LBSs based on *k*-anonymity, and point out that these do not follow the safe assumptions as per the original definition of *k*-anonymity. We propose an alternative LBS anonymity property, LBS *(k,T)*-anonymity, that ensures anonymity of a user's query against an attacker who knows about the issuance of the user query within a time window. We evaluate the vulnerability of the approaches in the literature to this type of attack that we believe is very basic and important, and assess the performance of our proposed algorithm for achieving LBS *(k,T)*-anonymity in terms of providing optimal solution.

1. Introduction

Location-based services (LBSs) are able to provide location specific information to (mobile) users, enabling more convenient and effective ways to access information. However, despite rapid technical developments in the area, it seems that they are lagging behind in deployment and leverage by providers and consumers. Privacy concerns are believed to be one of the major obstacles to the full-fledged emergence of these services. People are becoming more aware of privacy implications of using technologies nowadays, and LBSs are not exception to that. In fact, LBSs deal with large amounts of spatio-temporal data related to user movements, among other privacy-sensitive information in user queries. Once possibly collected by LBSs, such privacy-sensitive data are at risk of further analysis for malicious purposes. Although removing real identifiers (de-identification) or using pseudonyms instead may enhance privacy preservation, researchers have shown that other pieces of information may be used to re-identify user records in a de-identified table. In the context of LBS, user's location provided in the queries may be used to link a query 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. In a typical anonymization scenario, users submit their queries to a trusted anonymizer, which submits an anonymized version of the query to the LBS on behalf of the user, and later relays back its responses.

The *k*-Anonymity principles [1, 2] has been predominantly adopted by researchers for use in the context of LBSs. It essentially ensures that any linking attack cannot succeed by a probability exceeding $1/k$. Most of the proposed approaches for LBS privacy in the literature, such as New Casper [3], Privé [4], and PrivacyGrid [5], 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 makes it difficult for these approaches to provide acceptable anonymity for LBS users in practice. More specifically, LBS *k*-anonymity neglects to follow a *safe* approach regarding user population in *k*-anonymity (discussed in detail in Section 3) that imposes an unrealistic implicit assumption on the adversary's



id	Cloaked Loc	Query
1	R1	Gas station
2	R2	Grocery store
3	R3	Gas station
4	R4	Night club
5	R5	Hotel
6	R6	Hotel

Figure 1: General threat model

background knowledge. This unsafe assumption leads to vulnerability of these approaches to a very simple and basic anonymity attack as follows. 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 from 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, the above-mentioned 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. We believe that such an attack is very likely to occur in practice in the LBS context. Exception to this trend is Gedik's et al.'s approach in [6] that strictly follows the original definition of k-anonymity. However, their approach introduces delays in query anonymization in order to submit a group of k queries together, and does not formulate an optimization problem to minimize the size of a cloaked area.

In this paper, we analyze the predominant interpretation of k-anonymity in the LBS context and point out its subtle but important nonconformance to the original definition of k-anonymity, and its impractical, implicit assumptions as a result of this. We propose LBS *(k,T)*-anonymity which has safe assumptions regarding user population and aims to thwart an adversary's attack based on the knowledge of the existence of victim's query. Moreover, we formulate LBS *(k,T)*-anonymization as a spatio-temporal problem, and provide a greedy solution and some experimental results. We emphasize that our approach to temporal aspect of anonymization is different than what is considered in the related work on anonymizing continuous LBS queries [7, 8]. In this paper, we do not consider such scenarios, and limit our scope to one-time LBS queries.

The rest of the paper is organized as follows. We provide an overview of privacy threat model in LBSs in Section 2. In Section 3, we discuss and analyze the limitations of the generally-conceived interpretation of k-anonymity in the LBS context. We propose our approach, LBS *(k,T)*-anonymity, and formulate the problem of achieving it and a greedy solution in sections 4 and 5, respectively. The proposed algorithm is evaluated in a simulation framework in Section 6. Section 7 highlights the related work. We conclude the paper and provide future research directions in Section 8.

2. LBS Privacy Threat Model

We consider a general anonymization architecture consisting of mobile users, an LBS, and an anonymizer, depicted in Figure 1. From users' point of view, the anonymizer is a trusted entity that mediates queries between them and the untrusted LBSs; users submit their queries to the anonymizer; the anonymizer performs query anonymization and sends the anonymized queries to the LBS on behalf of the users. Any responses to the queries are sent back to the anonymizer, which can forward them to the corresponding users. The anonymizer may also perform some post-processing on the responses before sending them back to the users in order to reduce the uncertainties in the answers as a result of query anonymization.

In the setting described above, we assume that an adversary may be able to access to the queries at the LBS side, and therefore making LBS an untrusted entity in the system. The goal of adversary is to identify the query that a specific target user has issued. We assume the attacker knows the exact location of the target user as background knowledge. Figure 1 summarizes the architecture described and the threat model.

3. Analyzing The Predominant *k*-Anonymity Approach for LBSs

In this section, we present an interpretation of k-anonymity in LBSs that is widely captured by the existing approaches such as in [3, 4, 9, 5]. Let relations $AQ(location, query)$ and $UL(user, location)$ represent, respectively, the submitted anonymized queries to the LBS and the exact locations of the LBS users. Note that these relations are snapshots of the data at a specific time point. As the LBS is not considered trusted, relation AQ is considered known to the adversary. In the predominant interpretation of k-anonymity for LBSs, which we call *LBS* k*-anonymity*, the idea is to cloak 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 $1/k$.

Definition 1 (LBS *k*-anonymity). *Anonymized queries AQ is LBS* k*-anonymous iff for every query in AQ there exist at least k users in UL whose locations match the query's location. Formally:*

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

We show that the above interpretation of k-anonymity in the LBS context is not consistent with the original definitions of the k-anonymity principle [1, 2], which in turn result in not delivering the expected anonymity to the LBS users. In order to analyze the (in)consistency, we provide a brief background on the 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 [1].

Definition 2 (*k*-anonymity requirement). *Every combination of values of quasi-identifiers 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 [1]. Assuming that an individual is only associated with one tuple in a privacy-sensitive relation, the following definition ensures that for each tuple in a k-anonymous relation, there are at least k individuals that would match based on the quasi-identifier [2, 1].

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* $P[QI]$.

Mapping LBS k-anonymity to the above-mentioned definitions, AQ is the privacy-sensitive relation with the quasi-identifier $\{location\}$ (which can be linked to *location* in UL). We observe that LBS k-anonymity captures the k-anonymity requirement (Definition 2) by matching at least k user locations in UL for every query's location in AQ. 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. In the context of LBSs, this means that there should be at least k queries with the same cloaked location for every existing location in the AQ relation. Definition 1 clearly does not ensure this property.

Let us illustrate the privacy issue of such inconsistency in an LBS anonymization scenario. Suppose user queries are anonymized according to LBS k-anonymity. This means that if victim Alice has issued a query, her query's location has been cloaked to include at least k users' locations. However, if adversary Oscar simply knows that Alice has in fact issued a query and there are no other anonymized queries matching Alice's location, he can easily associate the query to Alice! The knowledge of existence of a query by Alice is a basic assumption of k-anonymity approaches and linking attacks for publishing datasets [2], which seems to have been neglected in the approaches that support LBS k-anonymity.

In fact, LBS k-anonymity implicitly considers a very strong assumption regarding adversary's background knowledge: the adversary believes that all the users located in the area enclosed by a query's location are potential issuers of the query. This is an impractical assumption. Because, first, it is very likely that the adversary does not have access to the exact location of every LBS user; access to such data is impossible maybe except for mobile network operators. Second, in a real-world scenario, the adversary may simply obtains knowledge of existence of the victim in the query table by observing/monitoring the victim, which can help him easily associate victim's record as mentioned above, without the need for much more complex background knowledge.

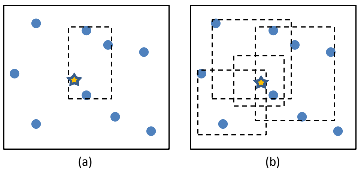


Figure 2: LBS k-anonymity (a) vs. our approach (b)

4. An Alternative Approach to Anonymity in LBSs

In this section, we propose an alternative formulation of anonymity principle for LBSs, that better adopts the original k-anonymity property (Definitions 2 and 3) compared to LBS k-anonymity. The idea is to avoid an adversary from being able to link less than k anonymized queries to a target user's location. Intuitively, this can be achieved by ensuring that every query issuing user's location is covered by at least k queries in AQ. Suppose the star-shaped point in Figure 2 is the location of the victim that issues a query and $k = 4$. Figure 2b depicts requirement of our approach in terms of k-anonymity, i.e., four queries should cover the victim's location, while Figure 2a shows the approach of LBS k-anonymity, i.e., assuring four users in the victim's cloaked location. It is worthwhile to note that these two approaches are somewhat dual of each other; LBS k-anonymity ensures k users for each tuple in AQ, while our approach ensures k queries in AQ for each issuing user.

Although our approach enjoys a more practical assumption regrading background knowledge, it might be too rigid to enforce in practice. As shown in Figure 2, our approach requires cloaking of $k - 1$ other query locations to anonymize a query, while LBS k-anonymity involved anonymizing only the same query. In order to compensate for this complexity to some extent, we relax our approach to consider less precise adversary's knowledge of the issuance time of the query as follows. The intuition is that even the attacker knows the exact location of the victim, he might not know he exact point in time that the query has been issued. For instance, Oscar may know that Alice has requested a service around noon, between 12:00pm to 12:10pm. But he might know that the exact time up to the second granularity. We consider that our anonymized query table AQ also includes a *time* field that indicate the time query has been submitted. Note that this is not considered as a new background knowledge for attacker, since the untrusted LBS knows about the time of query submission. We formally propose our anonymity approach as follows.

Definition 4 (LBS *(k,T)*-anonymity). *Relation AQ is LBS* (k,T)*-anonymous iff for each submitted query at time t_i, i.e., $q_i \in AQ^{t_i}$, issued by user $u_i \in UL^{t_i}$, there exist at least $k - 1$ other queries in the time window of size at least T. Formally:*

$$\forall t_1 \forall t_2 (t_1 \le t_i \le t_2) \wedge (t_2 - t_1 + 1 \ge T) \Rightarrow |\{q \in AQ^{[t_1,t_2]} | q.location \text{ covers } u_i.location\}| \ge k.$$

In the above definition, a table's superscript represents a selection of records in the table in a specific time point/window. The time window size T should be chosen in the way that it is less or equal to the potential size accuracy that is expected for an attacker. Note that considering $T = 1$, i.e., *(k,1)*-anonymity, essentially removes any assumption regarding less precise time background knowledge.

5. 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.

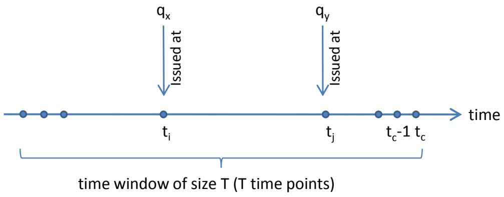


Figure 3: Time window of size T, ending at current time, that is considered in *(k,T)*-anonymization

5.1. 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 3 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 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* $A : L \to CL$*, such that*

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

5.2. A Greedy Algorithm

In this section, we propose a greedy approach to solve the LBS *(k,T)*-anonymization problem. Algorithm 1 shows a pseudo code of the proposed approach that should run at every time point. The inputs to the algorithm include the set of newly issued queries, the set of issued queries in the past $T-1$ time points. For convenience, we are using a number of notations. For a query q, $q.cl$ represents its cloaked location and $q.ul$ represents its issuer's location. Also, $coverage(q)$ represents the number of queries which their cloaked locations cover the issuer's location of the query q. It is initialized based on the cloaked locations of previously issued queries. The cloaked locations for newly issued queries, which are the actual output of the algorithm, are initialized to the corresponding locations of the query issuers.

The algorithm iterates until all queries (both previously and newly issued) have coverage degree of k. In each iteration, the cost for expanding a cloaked location of a newly issued query to cover an insufficiently covered query issuer is computed. For each insufficiently covered query issuer, the the potential expansion with minimum-cost is selected as a candidate. After selecting the candidate expansions (one for each an insufficiently covered query), our heuristic is to choose the one with maximum-cost to be applied in the iteration.

Algorithm 1 KT-Anonymize(Q_N, Q_I, k, T)

Input: new queries Q_N, issued queries in past $T - 1$ time points Q_I, parameters k and T in LBS *(k,T)*-anonymity
Output: cloaked locations in $Q_N.cl$

```
1:  $Q \leftarrow Q_I \cup Q_N$
2:  for each $q \in Q$ do
3:      $coverage(q) \leftarrow |\{q_i \in Q_I | q_i.cl \text{ covers } q.ul\}|$
4:  for each $q_n \in Q_N$ do
5:      $q_n.cl \leftarrow q_n.ul$
6:  while $\exists q \in Q, coverage(q) < k$ do
7:      for each $q \in Q, coverage(q) < k$ do
8:          for each $q_n \in Q_N$ do
9:              Compute cost of expanding $q_n.cl$ to cover $q.ul$
10:         Let $\langle q'_n, q' \rangle$ be the expansion with the minimum-cost among the possible expansions in line 9 ($q'_n.cl$
            expands to cover $q'.ul$)
11:     Let $\langle q''_n, q'' \rangle$ be the expansion with the maximum-cost among $\langle q'_n, q' \rangle$s computed in line 9
12:     Expand $q''_n.cl$ to cover $q''.ul$
13:     $coverage(q'') \leftarrow coverage(q'') + 1$
14:     Updates coverage for all other queries due to the above expansion
```

The algorithm is greedy in the way that it chooses the best expansion to be applied one at a time without considering possible optimizations using previous or possible future expansions. The expansion will increment the coverage for one query's issuer, obviously. However, as this is spatial expansion, it may result to possibly covering more query issuers. This is checked and updates are applied at the last step of each iteration of the algorithm. A little more explanation on the heuristic might be helpful. Since the ultimate goal is to provide enough coverage for every query issuer, in each iteration we choose to cover an issuer that has worst-cost choice considering all the possible potential expansions. The idea is that by covering such a location, since it results into a greater query location expansion among other candidates, it might end up covering other issuers also or at least dramatically reduce the cost of their coverage in future iterations. Note that we have to provide the coverage for such issuers according to the problem definition, regardless of the cost. The time complexity of Algorithm 1 is $O(k|Q|^2|Q_N|)$, where Q is the set of all queries in the past T time units that have inadequate coverage in the time window, and Q_N is the set of newly issued queries.

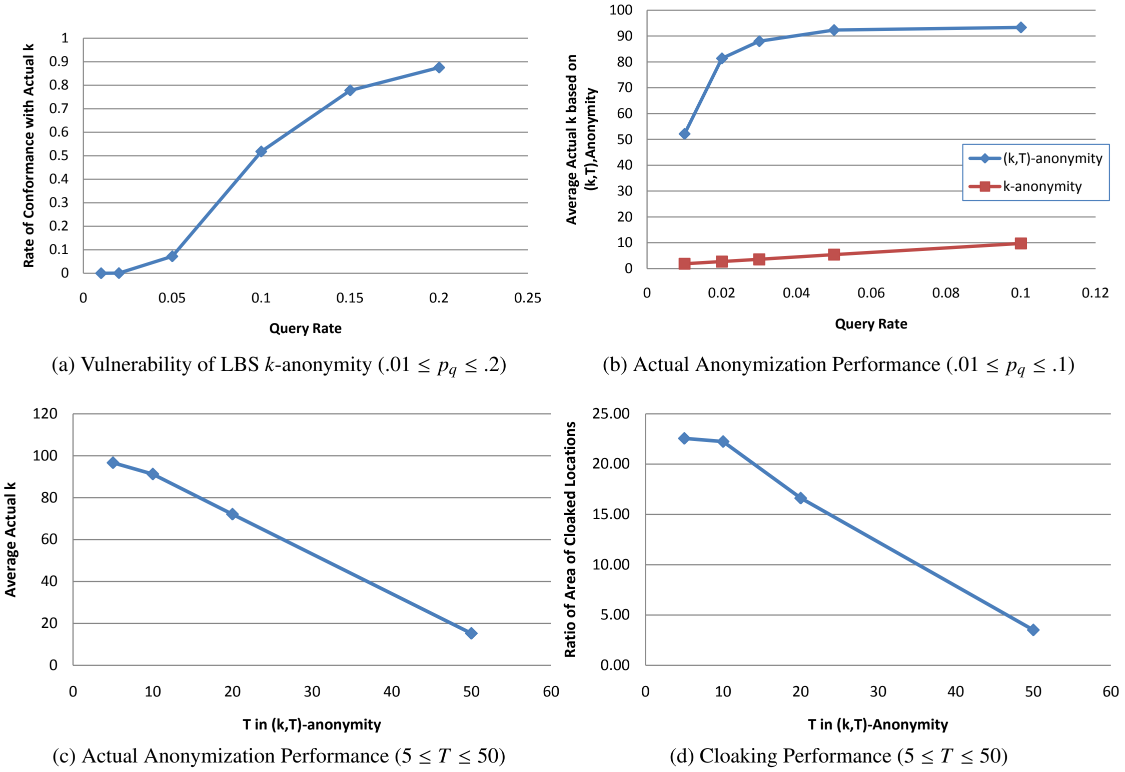
6. Experimental Evaluation

We have implemented an LBS anonymization evaluation framework in Java that leverages *Network-based Generator of Moving Objects* [10] to simulate generation of queries by mobile users on a given road network, performs anonymization schemes, and measures statistics regarding the quality of the anonymization and anonymized queries.

6.1. Evaluation Setup

In order to compare our approach with LBS k-anonymity, we implemented PrivacyGrid [5], a recent work that employs this idea. We also implemented a grid-based version of our proposed algorithm described in Section 5.2. As input to the moving object simulator, we used the road network of SF Bay Area (approx. $26k\ km^2$). The area was divided into a grid network of 270×358 square-shaped cells. We simulated movement of 1000 users 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. The parameters k and T were by default set to 10, unless otherwise mentioned.

In the evaluation of the anonymization techniques, we distinguish between k as input parameter to an algorithm, and its post measurement after performing the anonymization, which we call *actual* k. In our results, we report the average of this value for the collection of all submitted queries. In LBS k-anonymity, the number of users in any query's cloaked location is its actual k value. In LBS *(k,T)*-anonymity, the average number of queries that their location enclose the location of the user who issued a specific query, in any time window with size T that includes the query, is considered as its actual k.



(a) Vulnerability of LBS k-anonymity ($.01 \leq p_q \leq .2$)

(b) Actual Anonymization Performance ($.01 \leq p_q \leq .1$)

(c) Actual Anonymization Performance ($5 \leq T \leq 50$)

(d) Cloaking Performance ($5 \leq T \leq 50$)

Figure 4: Experimental Results ($n = 1000$ and $k = 10$)

6.2. Results

We investigate the vulnerability of the LBS k-anonymity approach when the adversary knows about the issuance of the query by the victim. This is a more practical adversary's background knowledge than what assumed in LBS k-anonymity, as explained in Section 3. We consider the relaxed version that is captured by LBS *(k,T)*-anonymity property, setting parameters $k = T = 10$, i.e., the attacker knows that the victim has issued a query in a specific time window of size 10, and should not be able to identify the victim's query with a probability higher than 0.1. Figure 4a depicts the rate of actual k, in the sense of LBS *(k,T)*-anonymity, matching parameter k in the PrivacyGrid algorithm at different query rates. The results show that even for a very high rate of query generation such as 0.15 roughly 23% of the queries are vulnerable based on the above-mentioned background knowledge. Note that 0.15 is considered a very high query generate rate, i.e., every user issues a query at each time point with 15% chance. Query rate is expected to be much lower in practice, which as shown in the figure has even much lower conformance to LBS *(k,T)*-anonymity .

Figure 4b shows the average actual k measurement with regards to LBS *(k,T)*-anonymity on our proposed algorithm and PrivacyGrid, at different query rates. As expected, LBS k-anonymity does not support the proposed $k = 10$ on average. Even when the average actual k meets proposed k, many individual queries fail to conform with *(k,T)*-anonymity property. 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 provision of the anonymity property. However, our technique shows better performance for larger window sizes. Figure 4c shows the improvement trend of actual k by adjusting parameter T, ranging from 5 to 50 ($p_q = 0.05$). A window size of 50 brings down the actual k to about 17, which is over 800% improvement compared to $T = 10$ in achieving closer value to the expected k value. Note that if for instance time is measured in the scale of seconds, such a window size is completely reasonable to be considered as attacker's uncertainty regarding issuance time.

Although we are dealing with a different problem than LBS k-anonymization techniques, we found it interesting to compare the size of cloaked areas generated by our method with the ones generated by PrivacyGrid. Figure 4d shows the average area ratio of the cloaked locations using our algorithm to the ones using PrivacyGrid, for a fixed query rate of 0.05. The results show that for low T values our algorithm generates cloaked area of about 22 larger than PrivacyGrid's. However, as we increase T, we get more acceptable cloaking performance (about 4 times larger at $T = 50$). This is partly because of the hard optimization problem that LBS *(k,t)*-anonymization deals with, compared to the much simpler problem in LBS k-anonymization approaches. We emphasize again that any direct comparison like this is not very insightful as we are dealing with two completely different problems, although in the very much the same context.

7. Related Work

Anonymization techniques have originated from the relational database community, where there is need to anonymize data before publishing for research or other purposes. In the k-anonymity approach [2], it is required that each record is indistinguishable from at least $k - 1$ others. L-diversity [11] proposes to prevent attacks threatening k-anonymized data with low diversity in sensitive attributes. It requires that within each k-anonymized equivalency class, there exist at least l different values for privacy-sensitive fields. T-closeness is a subsequent approach to overcome l-diversity limitations by requiring that the distribution of a sensitive attribute in any equivalence class be close to the distribution of the attribute in the overall table [12].

Research in the location privacy area can be categorized into two group: work that deals with location privacy from a different perspective than anonymization, and work that addresses anonymity issues and adopt anonymization techniques. In the former group, several approaches exist for obfuscating user location in order to hide the exact location information from LBS [13, 14]. Also, in a recent work, Ghinita et al. propose using private information retrieval (PIR) techniques [15]. PIR relies on cryptographic techniques to submit a query to a data server and retrieve the results without revealing either the query or the exact response. The advantage of such an approach is totally unrevealed location and no reliance on other entities to ensure location anonymity. However, PIR is inherently computationally intensive and involves exchanging a large amount data, which is most probably not suitable for mobile devices that use LBSs. In the latter group, as explained in Section 1, most work follow the interpretation of k-anonymity that we have formally defined as LBS k-anonymity (Definition 1) [3, 4, 9, 16, 5].

The work by Gedik et al. is an exception to the above fact as it more strictly follows the original definition of k-anonymity [17, 6]. The authors propose *CliqueCloak* algorithm that groups at least k number of queries together and submits all the queries in a group together to the LBS, using a cloaked region that encloses all issuing users' locations. They consider maximal spatial and temporal QoS parameters to constrain the grouping of queries. The algorithm relies on finding a clique in a constraint graph, and faces severe performance issues as a result of constraint graph formation and search for cliques on that.

Mokbel et al. propose Casper that anonymizes user requests, based on user-specified k-anonymity and minimum acceptable cloaked region, through a location anonymizer and empowers the server with a query processor to handle anonymized queries [3]. The anonymization scheme is based on a pyramid data structure comprising of grid of location cells with ascending resolution towards higher levels, i.e., each grid cell at a level corresponds to four cells in the next level. The users are registered in the highest level cells. The cloaking algorithm traverses the levels from the highest level all the way to a cell that contains at least k number of users and its area is larger than A_{min}, where k and A_{min} are user privacy preferences. The authors also propose algorithms to process the anonymized queries to provide inclusive answers with minimal size.

Ghinita et al. propose Privé as an anonymization technique with a distributed architecture, in contrast with centralized anonymizers [4]. Privé relies on a location cloaking algorithm based on Hilbert filling curve to ensure LBS k-anonymity in the case of attacker's full knowledge of users' locations. The authors define a reciprocity requirement for cloaked locations, based on which if a user happens to be in a cloaked location of another user's query, the latter user should also be located in queries issued by the former user. Although this requirement makes this approach more consistent with the original definition of k-anonymity, it does not completely remove the limitations regarding attacker's belief of user population and vulnerability to known user attack. In a complementary work, Kalnis et al. propose an alternative approach to Hilbert cloak, called Nearest Neighbor Cloak, which protects against heuristic attacks such as considering a user closer to the center of a query's cloaked area as the query issuer [9]. The authors analyze and compare the two algorithms using various experiments.

PrivacyGrid is a framework that supports user privacy preference and efficient cloaking algorithms [5]. It allows users to define a location privacy preference profile that specifies both preferred location hiding measures (location k-anonymity and l-diversity) and location service quality measures (maximum spaial/temporal resolution). The authors propose three approaches for location cloaking approaches: bottom-up, top-down, and hybrid grid cloaking. The basic idea behind the grid cloaking algorithms is similar to the one in [3]. However, by allowing dynamic expansion of grid cells (instead of static quad-tree scheme) it provides better result in terms of minimal cloaked region and performance.

8. Conclusion and Future Work

In this paper, we studied the conformance of the predominant interpretation of k-anonymity in the LBS context. Our analysis shows negligence of this interpretation in its assumption about the user population, and consequently vulnerability to a very basic attack against LBS users. We formulated a generalized alternative solution, called LBS *(k,T)*-anonymity, in order to avoid the mentioned limitations. We empirically showed the vulnerability of the predominant interpretation, and presented acceptable performance of our approach for reasonably large window sizes.

The proposed LBS *(k,T)*-anonymization in this paper is generally providing much larger actual k values than requested. This is partly due to the non-optimal, greedy nature of the proposed solution, and partly because of our conservative approach fool-proof against attacks. In short, in each step of the algorithm full expected coverage for all the queries in the past T time points is tried to be achieved. But this is only really necessary for the queries in the exact T points before. The rest will still have chance to be provided with remaining coverage in the future time points. However, since we have no guarantee that there will be enough future supporting queries, we take the conservative approach to fulfill required coverage sooner. This results to excess of coverage in later time points. We plan to investigate a less-conservative approach that relies on possibility of future queries for coverage purpose and therefore address both the coverage excess and large cloaking area issues. We will also consider uncertainty in the adversary's background knowledge regarding the issuer's location, and independent parameters k and T per query, as future work.

References

[1] P. Samarati, Protecting Respondents' Identities in Microdata Release, IEEE Trans. on Knowledge and Data Eng. 13 (6) (2001) 1010–1027.

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

[3] M. F. Mokbel, C. Y. Chow, W. G. Aref, The new Casper: Query processing for location services without compromising privacy, in: Proc. 32nd Int'l Conference on Very Large Data Bases, ACM, 2006, pp. 763–774.

[4] G. Ghinita, P. Kalnis, S. Skiadopoulos, PRIVE: anonymous location-based queries in distributed mobile systems, in: Proc. 16th Int'l Conference on World Wide Web, ACM, New York, NY, USA, 2007, pp. 371–380.

[5] B. Bamba, L. Liu, P. Pesti, T. Wang, Supporting anonymous location queries in mobile environments with PrivacyGrid, in: Proc. 17th Int'l Conference on World Wide Web, ACM, 2008, pp. 237–246.

[6] B. Gedik, L. Liu, Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms, IEEE Transactions on Mobile Computing 7 (1) (2008) 1–18.

[7] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, W. H. Wang, Anonymizing moving objects: how to hide a MOB in a crowd?, in: Proc. of the 12th Int'l Conference on Extending Database Technology: Advances in Database Technology, EDBT '09, ACM, 2009, pp. 72–83.

[8] G. Andrienko, N. Andrienko, F. Giannotti, A. Monreale, D. Pedreschi, Movement data anonymity through generalization, in: Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS, SPRINGL '09, ACM, New York, NY, USA, 2009, pp. 27–31.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, D. Papadias, Preventing Location-Based Identity Inference in Anonymous Spatial Queries, IEEE Transactions on Knowledge and Data Engineering 19 (12) (2007) 1719–1733.

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

[11] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, l-diversity: Privacy beyond k-anonymity, ACM Transactions on Knowledge Discovery from Data 1 (1) (2007) 3.

[12] N. Li, T. Li, S. Venkatasubramanian, t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, in: 2007 IEEE 23rd International Conference on Data Engineering, IEEE, 2007, pp. 106–115.

[13] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. di Vimercati, P. Samarati, Location Privacy Protection Through Obfuscation-Based Techniques, in: Proc. 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security,, no. 4602 in LNCS, 2007, pp. 47–60.

[14] M. Duckham, L. Kulik, A Formal Model of Obfuscation and Negotiation for Location Privacy, in: Proc. 3rd Int'l Conference on Pervasive Computing, Vol. 3468 of Lecture Notes in Computer Science, Springer, 2005, pp. 152–170.

[15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, K. L. Tan, Private queries in location based services: anonymizers are not necessary, in: Proc. ACM SIGMOD Int'l Conference on Management of Data, ACM, 2008, pp. 121–132.

[16] A. Solanas, A. M. Ballesté, A TTP-free protocol for location privacy in location-based services, Computer Communications 31 (6) (2008) 1181–1191.

[17] B. Gedik, L. Liu, Location Privacy in Mobile Systems: A Personalized Anonymization Model, in: Proc. 25th Int'l Conference on Distributed Computing Systems, IEEE Computer Society, 2005, pp. 620–629.