Defend Seclusion Against Location-Based Personal Recognition

T Senthil Kumar[1], Udaya Lakshmi Jaladi[2]

1 Assistant Professor, Dept of Computer Science and Engineering,SRM University,Chennai,India

2 Research scholar, Dept of Computer Science and Engineering,SRM University,Chennai,India,

***Abstract*—*Privacy protection has recently received considerable attention in location-based services. A large number of location cloaking algorithms have been proposed for protecting the location privacy of mobile users. Then consider the scenario where different location-based query requests are continuously issued by mobile users while they are moving. Then the most of existing k-anonymity location cloaking algorithms are concerned with snapshot user locations only and cannot effectively prevent location-dependent attacks when users' locations are continuously updated. Therefore, adopting both the location k-anonymity and cloaking granularity as privacy metrics, The new incremental clique-based cloaking algorithm, called ICliqueCloak, to defend against location-dependent attacks is introduced. The main idea is to incrementally maintain maximal cliques needed for location cloaking in an undirected graph that takes into consideration the effect of continuous location updates. Thus, a qualified clique can be quickly identified and used to generate the cloaked region when a new request arrives. The efficiency and effectiveness of the proposed ICliqueCloak algorithm are validated by a series of carefully designed experiments. The experimental results also show that the price paid for defending against location-dependent attacks is small and maintaining security along with the data management of the database for the user end.*

Keywords- Location privacy, data management, location-based services,cloaking-granularity.

1.INTRODUCTION

To provide an security for the Location Based Services(LBS). The privacy threat of revealing a mobile user's personal information through his/her location has become a key issue to be concerned. Example, using her PDA phone, Alice wants to find out "the nearest hospital with specialty in ophthalmology" while hiding her exact location (e.g., being in a clinic or at home) and the sensitive information that it is her (Alice) who made this query. A straightforward method is to replace her identity with a pseudonym before sending the query to the service provider. But this is not enough. Location information included in the query can be used as a quasi-identifier to reidentify the user . To address the location privacy issue, location k-anonymity and cloaking granularity are two commonly used privacy metrics.

- **Location k-anonymity**:A mobile user is considered location k-anonymous if and only if the location information sent to the service provider is made indistinguishable from that of at least k _ 1 other users. To achieve location k-anonymity, exact user locations are extended to cloaked regions such that each region covers at least k users.
- **Cloaking granularity**: It requires the area of cloaked region to be larger than a user-specified threshold.While the location k-anonymity protects the user identity (out of k users), it may not be able to prevent the location disclosure (e.g., a cloaked region

covering k users in populated areas could be very small). On the other hand, the cloaking granularity prevents the location disclosure but cannot defend against attacks for user identifies in the cases where user locations are publicly known and there is only one user in the cloaked region .

The prior solutions in only considered the cloaking granularity as the privacy metric, which, as discussed earlier, may fail to protect the user identity in case there is only one user in the cloaked region. The cloaking granularity and location k-anonymity as privacy metrics. The proposal of a new location cloaking algorithm, called ICliqueCloak, to incorporate the effect of continuous location updates in the process of location cloaking.

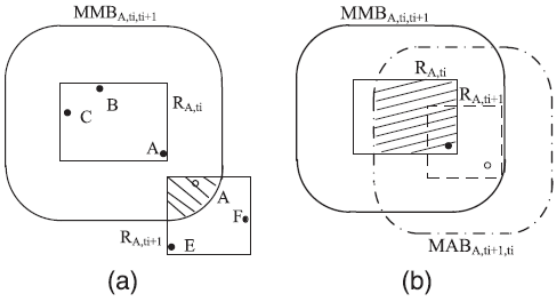


Fig. 1. Example of location-dependent attacks ($k = 3$).

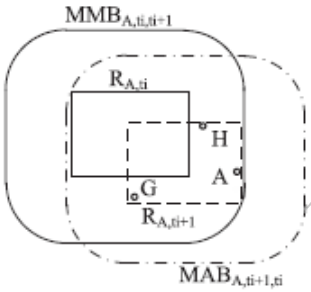


Fig. 2. ICliqueCloak algorithm ($k = 3$).

As illustrated in Fig. 2, at time ti+1, the cloaking algorithm is aware of RA,ti and MMBA,ti ,ti+1 , and attempts to find the cloaked region for A within MMBA,ti,ti+1 . If the attacker knows each user's speed limit, he/she still cannot tell the exact location of A in RA ,ti and RA,ti+1 . it is better to use a graph model to formulate this problem. Each location-based query request is represented by a node in the graph, an edge exists between two nodes only if they are within the MMB of each other and can be potentially cloaked together. To meet the location k-anonymity requirement, the problem becomes to find k-node cliques in the graph such that all the nodes within a clique form a cloaking set. To reduce the computational complexity, and to maintain the maximal cliques incrementally. That is, all maximal cliques are identified at the beginning of the process, they are then incrementally maintained based on three classes: positive candidates, negative candidates, and non candidates. Thus, a qualified clique can be quickly identified and used to generate the cloaked region when a new request arrives.to conduct a series of experiments to evaluate the performance of the proposed ICliqueCloak algorithm using both location data generated from a well-known road network simulator and adapted from a real data set . Experimental results show that ICliqueCloak is efficient in terms of various performance

metrics including the cloaking time, the request processing time, and the cloaking success rate, while its anonymization cost is only slightly increased in comparison with the existing algorithm.

Objectives

- Preventing Snapshot Location Attacks.
- Preventing Query Tracking Attacks.
- Preventing Location-Dependent Attacks.
- Preventing Trajectory Attacks.
- Location Privacy.
- Mobile Data Management.

2.MODULES

2.1 Preventing Snapshot Location Attacks.

When exact snapshot locations are disclosed, two kinds of attacks may happen: location linking attacks and query sampling attacks . Location linking attacks refer to the scenario where the location information included in a user query is used as a quasi-identifier to re identify the user. For example, if a location exclusively belongs to some owner. The corresponding query can thus be linked to the location owner. The location k-anonymity model was proposed to prevent this kind of attacks . The basic idea is to extend an exact user location to a cloaked region that covers at least k users. In a Quad-tree-based cloaking algorithm is used to generate cloaked regions. In each user can specify the minimum tolerable area of a cloaked region as well as the smallest privacy level k, and a variant Quad-tree is used to compute cloaked regions. It models user-specified privacy requirements using an undirected graph and identifies the cloaking sets through finding cliques in the graph. Our work also employs a graph model to facilitate location cloaking, but differs from in several aspects. First concerned with QOS support in privacy protection for snapshot locations and hence could suffer from location dependent attacks, whereas the aim of our cloaking algorithm is to protect location privacy against location dependent attacks. Second, the methods for finding a new request arrives, the algorithm needs to recursively search the neighbors of the node representing the request. In contrast, in our approach is incrementally maintain the maximal cliques in the graph, thus a cloaking set is found in some maximal clique instead of recursively searching the neighbors. Third, the requests with privacy levels higher than that of the newly arrived request cannot be cloaked.

2.2 Preventing Location-Dependent Attacks

Location-dependent attacks (illustrated in Fig. 1) are the main focus. To prevent location-dependent attacks, proposed two simple solutions, namely patching and delaying. The patching, enlarges the current cloaked region to cover the last one so that the overlapped area with the MMB is at least as large as the last cloaked region. The drawback is that the size of the cloaked

region would increase significantly as time evolves. The delaying, suspends the request by t time until the MMB grows large enough to fully contain the current cloaked region. However, the user may have already changed her location and is no longer in the cloaked region at time (ti+1) ,t.the postpone requests, and they considered the scenario where the attacker has prior knowledge about the placement of sensitive regions on a map. Developed a mobility-aware cloaking technique by considering mobility patterns in location cloaking. However, different from our work, the privacy metric employed in these previous studies is only the granularity of cloaked regions (without considering the location k-anonymity). The next is which employed entropy of information theory to measure the location anonymity level by considering the probabilities of users being in a cloaked region. As entropy does not care whether user locations are actually different, the exact user location would be disclosed if all k users are at the same location. To overcome the limitations of the previous work, employ both the cloaking granularity and location k-anonymity as privacy metrics.

2.3 Preventing Query Tracking Attacks

The location privacy of continuous queries has been considered for a continuous query, the query results would be continuously returned for a designated time period (called query lifetime). For example, consider a sample query "finding the nearest gas station in the next five minutes." The query lifetime is 5 minutes. Query tracking attacks become possible if a user is cloaked with different users at different time instances during the query life time . Consider a continuous query CQA issued by a user A. If the user is cloaked with different sets of users for these two continuous queries, the location-dependent attack shown in Fig. 1 may still happen. On the other hand, if the user is always cloaked with the same set of users, the cloaked region would eventually expand to the whole service region when the users move apart and issue more and more queries over time.

2.4 Preventing Trajectory Attacks

When a trajectory is published, the owner might be inferred by attackers, even though the identifier has been removed. These types of attacks are called trajectory attacks. The problem of trajectory anonymization is to publish trajectories in such a way that the anonymity of each trajectory is preserved, while the utility of published data is maximized. The existing work can be classified into two categories: trajectory anonymization in free space. trajectory anonymization problem. In addition, trajectory anonymization is typically an offline process. In contrast, location cloaking considered in this paper is an online process that is invoked during processing of location-based queries. Moreover, trajectory anonymization requires a series of locations on a trajectory to be cloaked all at once But in our location cloaking problem, user locations are cloaked on the fly along with new requests. Therefore, existing methods for anonymizing trajectories are not applicable to our problem.

3 PRELIMINARIES

3.1 System Architecture

A mobile user sends location-based query requests (e.g., "finding the nearest gas station"), in the form of (id, l, P, q),to the anonymizing proxy through an authenticated and encrypted connection, where id is the real user identity, l is the user location, P contains the privacy parameters (to be detailed in Section 3.2), and q represents the query content. The anonymizing proxy consists of a cloaking engine, a cloaked repository, and a results refiner. Upon receiving a location-based query request, the checking that whether the user had made any queries before. If this is a first-time user, the cloaking engine replaces the user's id with a new pseudonym id', otherwise it replaces id with the user's existing pseudonym id'. Next, the query request is forwarded to the cloaking process to generate a cloaked region R in accordance with the user's privacy requirements. When the cloaking succeeds, the cloaked request is saved in the cloaked repository in the form of (id, id0, P, Rti , ti), where Rti is the user's cloaked region at time ti. Afterward, the anonymizing proxy forwards the modified query request (id', R, q) to the service provider. On the service provider side, upon receiving a location based query request (id', R, q), it will search and return all candidate results that are potentially query results of some location point within R. After that, these candidate results will be further refined by the anonymizing proxy using the user's exact location l. Finally, the refined results will be securely returned to the mobile user. The main focus is on the location cloaking algorithm, which is concerned with how to extend a location l to a cloaked region R without violating user specified privacy requirements.

3.2 Privacy Model Attacks.

Definition 1 ((k, A, dt)-Location Privacy Model). In order to accommodate personalized privacy requirements, each user can specify three parameters for protecting the location privacy

- k: It represents the anonymity level in the location k-anonymity model.

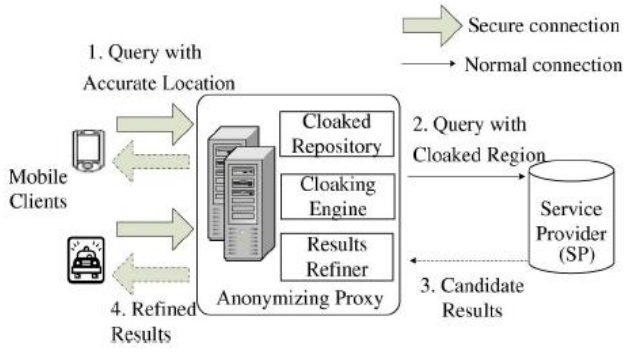


Fig. 3. System architecture.

- Amin: It specifies the minimum area that a cloaked region should have. This is to prevent the cloaked region from being too small for highly populated areas.
- dt: It is the maximum tolerable cloaking delay, which is a QoS
- **Definition 2 (Knowledge of the Attacker).**

Any party owning the following knowledge can be a potential attacker:

- a set of historical cloaked regions,
- the maximum moving speed of the user.

Definition 3 (Location-Dependent Attacks). Assume that

- Ru,ti and Ru,tj are user u's cloaked regions at times ti and tj, respectively,
- the maximum speed of user u is vu.The maximum movement boundary (MMBu,ti ,tj) of u at tj is a round rectangle that extends Ru,ti by a radius of uu _ (tj _ ti).

Denote by MMu,ti ,tj the intersection area between Ru,tj and
MMBu,ti ,tj :

$$MM_{u,t_i,t_j} = MMB_{u,t_i,t_j} \bigcap R_{u,t_j}.$$

Similarly, the maximum arrival boundary (MABu,tj,ti) of u at tj is a round rectangle that extends Ru,tj by a radius of uu _ (tj _ ti). Denote by MAu,tj,ti the intersection area between Ru,ti and MABu,tj,ti

$$MA_{u,t_j,t_i} = MAB_{u,t_j,t_i} \bigcap R_{u,t_i}.$$

If any inequality below holds:

- MMu,ti ,tj !=Ru,tj for any ti and tj, and
- MAu,tj,ti != Ru,ti for any ti and tj,

the location privacy of u might be compromised. This attack is termed as location-dependent attack.

In the previous example of Fig. 1a,MMA,ti,ti+1 of user A is the shaded area. Since MMA,ti,ti+1 !=A,ti+1 , A could suffer from location-dependent attacks.

3.3 Cloaking Set

Definition 4 (MaxMin Distance). Let Ri and Rj be two cloaked regions. The MaxMin distance from Ri to Rj is defined as:

MaxMinD(Ri,Rj)=maxmindistance(p,q).

MaxMinD(Ri,Rj)<=uu +(tj - ti).

(MaxMinD(Rj,Ri)<=uu .(tj _ ti)). Therefore, the definition of cloaking set as follows.

Definition 5 (Cloaking Set). Let CS be a user set and its
Minimum bounding rectangle (MBR) be Ru, ti at time ti.

Denote the previous cloaked region of each user u by Ru,ti_1 . CS is a cloaking set if and only if for any user u 2 CS,

1. MaxMinD(Ru,ti,Ru,ti_1) <= uu .(ti _ ti_1),

2. MaxMinD(Ru,ti_1,Ru,ti)<=uvu . (ti _ ti_1),
3. the privacy level ku <=CS,
4. the minimum area Aminu <=Area(MBR(CS)).

The first two conditions ensure that the cloaked region at any time is free of location-dependent attacks, the third condition is to protect the user identity by following the k-anonymity requirement, and the fourth condition ensures that the area of the cloaked region is not too small in a populated area. MBR RA,ti)1 is the cloaked region. The average area of the cloaked region for each query as a measure for anonymization cost.

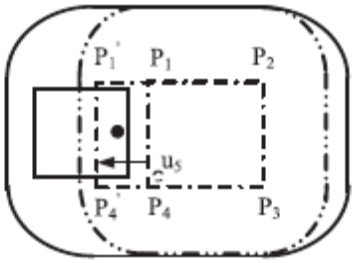


Illustration of extending cloaked region.

4 ICLIQUECLOAK ALGORITHM

4.1 Overview of ICliqueCloak

Intuitively, we want to use the MaxMin distance to find the safe cloaked region Ru;ti for a newly arrived request u at time ti. But on the other hand, we require Ru;ti as the input in computing the MaxMin distance. Therefore, the basic idea of our algorithm is to find a candidate cloaking set first, and
then extend some sides of the cloaked region until every user in the cloaking set is free of location-dependent attacks. The proposed ICliqueCloak algorithm involves four main steps. First, upon the arrival of a new request u, the existing requests that are in u's MMB and vice versa are detected and modeled in an undirected graph. Then, a cloaking set that satisfies location k-anonymity. MaxMin distance.
found from the undirected graph, and the MBR of the cloaking set is considered a candidate cloaked region. Next, the candidate cloaked region is checked whether it needs to be adjusted in order to prevent from location-dependent attacks. Finally, the graph will be updated accordingly if the cloaking is successful or some request(s) are found expired. Now, let's formally define the graph model.

Algorithm 1. Overview of ICliqueCloak
Input: a set of requests awaiting for anonymization, a new query request u
Output: a set of cloaked requests
1: Step 1: incrementally update the max-clique set for the
new request u (Section 4.2)
2: Step 2: find the cloaking set CSti satisfying location kanonymity from the max-clique set (Section 4.3)
3: Step 3: generate the cloaked region for CSti (Section 4.4)
4: Step 4: update the max-clique set upon request cloaking or expiration (Section 4.5)

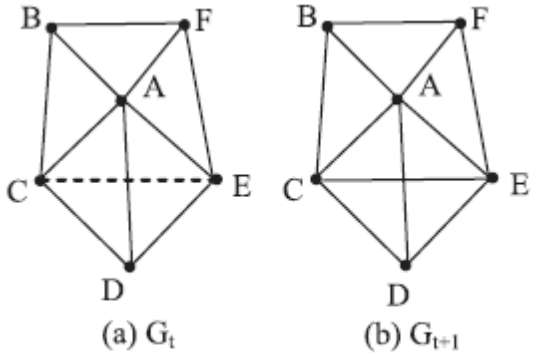
We use a simple example to illustrate the algorithm. Fig. 5a shows four requests u1, u2, u3, u4 from different users with their MMBui;ti_1;ti at time ti. Each MMBui ;ti_1;ti is extended from the previous cloaked region Rui;ti_1 .We can see that u1 is covered byMMBu2;ti_1;ti andMMBu3;ti_1;ti , and that u2 and u3 are both covered by MMBu1;ti_1;ti . Thus, an edge exists between u1 and u2, as well as between u1 and u3. Similarly, an edge exists between u1 and u4, as well as between u2 and u3, as shown in Fig. 5b. Now suppose that a new request u5 arrivesm, new edges between u1 and u5, u2 and u5, and u3 and u5 are added to the graph .As a result, a clique {u1, u2, u3, u5} can be found. Assume that ku1 { ku2 } ku3 {ku4 } 4 and {ku5}

3. The size of the clique is 4, which is no smaller than any k value. Meanwhile, assuming Area MBR {u1; u2; u3; u5}, max u2{u1;u2;u3;u5} Aminu , the MBR of {u1, u2, u3, u5} can be a candidate cloaked region CRti for this set of users.

Next, for each user in {u1, u2, u3, u5}, the algorithm checks whether the user's previously cloaked region at time ti_1 is covered by MAB of CRti . For simplicity, we take user u5 as an example. As shown in Fig. 6, the rectangle with solid lines is the cloaked region Ru5;ti_1 of u5 at time ti_1, and the rectangle with dotted-dashed lines (i.e., P1P2P3P4) is its cloaked region Ru5;ti at ti. The round rectangle with dotted dashed lines is the MABu5;ti;ti_1 of u5 from ti to ti_1. From the figure, we can see that MABu5;ti;ti_1 cannot fully cover Ru5;ti_1 . As such, u5 would suffer from location-dependent
attacks. Therefore, the cloaked region P1P2P3P4 is extended, where the edge of P1P4 moves to P01
P04 . As a result, Ru5;ti_1 can be fully covered by the new MAB (the round rectangle with solid lines). Further assuming that the new cloaked region P01 P2P3P04 is still in each user's MMB, it will then be returned as the cloaked region for {u1, u2, u3, u5}. Finally, u1, u2, u3, and u5 are removed from the graph.

4.2 Incremental Maximal Cliques

In this sections, the detail each step of Algorithm 1. To find a candidate cloaking set in the graph upon the arrival of a new request, the cloaking algorithm proposed in exhaustively searches the graph for cliques covering the new request. In the following, to present a new more efficient cloaking algorithm based on incremental maintenance of maximal cliques. For a graph without any edges, each node is a 1-node clique. Denote this set of maximal cliques by MCSet.



Example of adding a new edge.

The cliques in Ct can be partitioned into three classes:

- the cliques containing node u (Cu_t),
- the cliques containing node w (Cw_t),
- the cliques containing neither u nor w.

It is easy to see that adding edge euw to the graph can alter only the maximal cliques in Cu t or Cw t.

Algorithm 2. Incremental updating max-clique set
Input: max-clique set MCSet, a new request u
Output: updated max-clique set MCSet
1: add a new clique fug to MCSet
2: if the user of u had not issued any query before then
3: set u's last cloaked region to the whole service area
4: find u's neighbors according to Definition 7
5: push all edges connecting u and its neighbors to EdgeQueue
1512 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 8, AUGUST 2012
Fig. 7. Example of adding a new edge.
6: while EdgeQueue is not empty do
7: MCSet ;
8: pop up the first edge euw from EdgeQueue
9: find clique sets Cu t and Cw t in MCSet
10: compute C < Cu t \ Cw t
11: for each c 2 C do add c [fu;wg to MCSet0
12: C < MCSet.
13: for each ci 2 Cu t [Cw t do
14: for each ck 2 C do
15: if ci 6_ ckthen add ci to MCSet0
16: for each ci 2 MCSet _ Cu t _ Cw t do
17: add ci to MCSet.
18: MCSet <MCSet.

4.3 Finding the Cloaking Set

After incrementally updating the max-clique set, the cliques where the new request is involved might be candidates for the cloaking set. They can be classified into three classes: positive candidates, negative candidates, and non candidates.
Algorithm 3. Finding cloaking set in negative candidate clique
Input: negative candidate clique c
Output: candidate cloaking set CSti
1: sort the requests in c in descending order of their privacy level k
2: while jcj < max k and MBR(c) _ b Amin do

3: drop the request with the highest k from c
4: update jcj, max k, MBR(c), and b Amin
5: if jcj _ max k and MBR(c) _ b Amin then
6: CSti the set of current requests in c
7: else
8: CSti ;

4.4 Generating the Cloaked Region

Recall that a cloaking set should satisfy four conditions (see Definition 5). However, the cloaking set obtained from Algorithm 4 satisfies all the conditions except the second one. In order to resolve this, is need to extend the boundary of the cloaking set, such that for each request u 2 CSti , its previous cloaked region Ru,ti_1 is covered by the new MAB. The algorithm works as follows. After that get the cloaking set CSti , its MBR is computed as the candidate cloaked region CRti . For each request u 2 CSti , compute d = MaxMinD(Ru,ti_1, CRti). If d > vu _ (ti _ ti_1), _d = d _ vu _ (ti _ ti_1). Then, CRti is extended by _d on the direction where Ru,ti_1 locates w.r.t. CRti . Finally, the enlarged region is returned as the cloaked region Rti of the
cloaking set. The pseudocode is given in Algorithm 5.
Algorithm 4. Finding the cloaking set
Input: max-clique set MCSet, the new request u
Output: candidate cloaking set CSti
1: canCR = fc 2 MCSetju 2 cg
2: sort canCR in descending order of clique size
3: for each c in canCR do
4: find max k, min k, b Amin for c
5: if jcj _ max k and b Amin _ MBR(c) then
6: CSti = the set of requests in c, break
7: if jcj < max(ku, min k) or MBR(c) < b Amin then
8: not a candidate, continue
9: else
10: CSti = returned result of invoking Algorithm 3
11: if CSti = , then break

4.5 Update the Max-Clique Set for Leaving Requests

The requests will leave the system after they have been successfully cloaked or failed to be cloaked due to expiry of their tolerable cloaking delays. To efficiently identify expired requests, can employ a min-heap to keep track of the set of alive requests, where the key is the expiry

time. For both successfully cloaked requests and expired requests, they are seen as leaving requests. When they leave, the max-clique set should be updated accordingly. Specifically, for each leaving request, it should be removed from the maximal cliques where it is involved. Recall that a subset of a maximal clique is still a clique, but not guaranteed a maximal clique. Thus, need to check whether the updated cliques are still maximal. To do so, that can examine each of such cliques against other cliques remaining in the system. If it is a subset of any other clique, it is not a maximal clique and should be removed from the max clique set. For example, assume MCSet = {{A, B,C], {A,C,D}}. After the request D expires, it should be removed from {A,C,D}. {A,C,D} is updated to {A,C}, which is no longer a maximal clique since it is a subset of {A, B,C}. Hence, MCSet = {A, B,C} after removing D.

Updating max-clique set for leaving request
Input: max-clique set MCSet, leaving request u
Output: updated max-clique set MCSet
1: remove u from the min-heap
2: for each c in MCSet where u 2 c do
3: for each c0 in MCSet do
4: if c _ c0 then
5: remove c from MCSet, break

5 PERFORMANCE EVALUATION.

CONCLUSION:

The investigated cloaking algorithms that protect location privacy against location-dependent attacks. Its showed that most of the existing location cloaking algorithms cannot effectively defend against locationdependent attacks as they are concerned with snapshot user locations only. To address this problem,need to employ a graph model to formalize the problem and transformed it to the problem of finding k-node cliques in the graph. an incremental clique-based cloaking algorithm called CliqueCloak to generate cloaked regions. A series of experiments has been conducted to evaluate ICliqueCloak under various system settings. The experimental results show that the price paid for locationdependent attacks is small. The average processing time is only 5.7 ms and the cloaking success rate is about 97 percent for most cases, which validate the efficiency and effectiveness of the proposed ICliqueCloak algorithm.as well as managing the data sets and also providing security.

REFERENCES:

[1] B. Gedik and L. Liu, “Location Privacy in Mobile Systems: A Personalized Anonymization Model,” Proc. IEEE 25th Int’l Conf. Distributed Computing Systems (ICDCS ’05), pp. 620-629, 2005.

[2] K. Bharath, G. Ghinita, and P. Kalnis, “Privacy-Preserving Publication of User Locations in the Proximity of Sensitive Sites,” Proc. 20th Int’l Conf. Scientific and Statistical Database Management (SSDBM ’08), July 2008.

[3] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving User Location Privacy in Mobile Data Management Infrastructures," Proc. Privacy Enhancing Technology Workshop (PET '06), 2006.

[4] C. Chow and M.F. Mokbel, "Enabling Private Continuous Queries for Revealed User Locations," Proc. 10th Int'l Conf. Advances in Spatial and Temporal Databases (SSTD '07), 2007.

[5] Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002, Official J. European Communities, 2002.

[6] J. Du, J. Xu, X. Tang, and H. Hu, "iPDA: Enabling Privacy- Preserving Location-Based Services," Proc. Conf. Mobile Data Management (MDM), 2007.

[7] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Prive: Anonymous Location-Based Queries in Distributed Mobile Systems," Proc. 16th Int'l Conf. World Wide Web (WWW '07), 2007.