How to cite this article:

Saravanan, P. S., & Balasundaram, S. R. (2020). Enhanced privacy protection against location-dependent attacks in location based services using spatial cloaking. *Journal of Information and Communication Technology,* 19*(1)*, 45-63. https://doi.org/10.32890/jict2020.19.1.3

ENHANCED PRIVACY PROTECTION FROM LOCATION-DEPENDENT ATTACKS IN LOCATION BASED SERVICES USING SPATIAL CLOAKING

Perumal Shanthi Saravanan & Sadhu Ramakrishnan Balasundaram
Department of Computer Applications,
National Institute of Technology Tiruchirappalli, India
shanthisaravanan09@gmail.com; blsundar@nitt.edu

ABSTRACT

Use of Internet enabled mobile devices has facilitated the rapid development of location-based services (LBS). LBS allow users to access useful information such as the nearest ATM, temple, and so on. Although users enjoy the convenience of LBS, they are being exposed to the risk of location disclosures which could lead to potential abuse of location data. Hence, location privacy protection has recently received considerable attention in LBS. There are numerous techniques presented by various researchers to protect the location-context of users. Location cloaking is an often used technique to protect location-contexts. Most of the existing location cloaking algorithms are only concerned with snapshot user locations and cannot effectively prevent users from location-dependent attacks when user location-contexts are continuously updated. This paper presents a solution to protect users from location-dependent attacks by improving the existing clique based cloaking algorithm. The main idea is to maintain maximum sized cliques required for location cloaking in an undirected graph. Thus, a qualified clique can be quickly identified and used to generate a cloaked region when a new request arrives. In addition, dummy queries are generated to protect users from unusual situations. Through maximum sized cliques and dummy query generation, more user queries get

Received: 4/11/2018 **Revised:** 4/4/2019 **Accepted:** 24/04/2019 **Published:** 23/12/2019

cloaked within a reasonable amount of time, thereby providing better privacy protection when using LBS applications. The experimental results showed that the proposed cloaking algorithm outperformed existing algorithms such as IClique, OptClique and MMBClique in terms of its cloaking success rate and processing time.

Keywords: Location based services, location-dependent attacks, privacy preservation, spatial cloaking

INTRODUCTION

Advances in information and communication technologies (ICTs) have revolutionized the way in which people perform their activities and obtain benefits from automated services. Especially mobile technologies have paved the way for getting details of services in less time; wherever the requester may be, and whenever the need arises. Location Based Services (LBSs) are a growing category of mobile applications that enable information services to be accessible with the help of mobile devices through the mobile network and to make use of the location of the mobile device. Though mobile users get benefits from the LBS applications, they have to expose their location-context to service providers (Bamba, Liu, Pesti, & Wang, 2008; Du, Xu, Tang, & Hu, 2007; Xu, Teo, Tan, & Agarwal, 2009). Malicious service providers through location-context, can determine users' life style, personal details and in extreme cases, track individuals (Cheng, Zhang, Bertino, & Prabhakar, 2006; Ghinita, Kalnis, Khoshgozaran, Shahabi, & Tan, 2008; Chow & Mokbel, 2009; Lin, Zakariah, & Mohamed, 2010; Pan & Meng, 2013; Shanthi & Balasundaram, 2015). This violates the privacy of users. Hence the location-context must be protected from adversaries, including malicious service providers (Kalnis, Ghinita, Mouratidis, & Papadias, 2007).

An efficient way to protect users is to blur the actual location-context into a cloaked region (CR) which preserves location k-anonymity property (Mokbel, Chow, & Aref, 2006; Um, Kim, & Chang, 2010; Pan, Xu, & Meng, 2012). Gruteser and Grunwald (2003) have incorporated the k-anonymity mechanism of relational databases to protect the user's location privacy. In the relational database, k-anonymity (Sweeney, 2002) used in the context of privacy preservation means that for each tuple, there is at least k-1 similar tuples. Whereas in location privacy, k-anonymity (Xu & Cai, 2007) means that for each user, there is at least k-1 users of the same location. That is to say,

a minimum of k numbers of users are sharing the same location-context while issuing the location-based query to service providers. Thus, the cloaked region introduces uncertainty in finding the exact location-context of users; thereby protecting the location privacy of the users.

Most of the existing cloaking algorithms (Gedik & Liu, 2005; Mokbel et al., 2006; Um et al., 2010; Shanthi & Balasundaram, 2015; Kuang et al., 2017; Biswas & Sairam, 2017; Nguyen, 2017) have not considered the effect of continuous location updates of users. The continuous location updates may result in serious privacy breaches when different one-shot queries are frequently issued by mobile users. Towards this end, this paper proposes MClique-Dynamic cloaking algorithm with the goal of protecting users while updating their location-contexts continuously.

BACKGROUND AND RELATED STUDIES

The scenario for location-dependent attacks is given in Figure 1. Assuming that at time t_1 user Shan sends a query to 'Find the nearest ATM' and is cloaked into the region R_1. Later, at time t_2 she issues another query to 'Find shopping mall within a km' and is cloaked into the region R_2 (users in R_1 and R_2 are shown in Figure 1). Suppose an attacker knows the historical region of Shan (i.e.) R_1 and R_2 and her speed limit, then it is easy to pinpoint the current location of Shan. The reason is that the user must be limited to the maximum movement boundary (MMB) computed at time, t_1. The MMB is computed by extending the previous CR by a radius of 'r' and its computation is given in Equation 1.

$$r = (t_2 - t_1) \times \text{speed of the vehicle} \quad (1)$$

r = radius by which the CR generated at time t_1 is extended to form the MMB
t_1 and t_2=query sent time by the user

From this inference, an attacker can deduce the user, Shan, who must be located in the overlapped area of MMB of the user at time t_1 and R_2. The overlapped area may also be a single location point. In this case, the exact user location is disclosed with strong evidence. Similarly, the previous location of the user can also be deduced by an attacker. If an attacker knows the maximum arrival boundary (MAB) of the user, current and previous cloaked regions then the previous location of the user is limited to the intersection area of the MAB and previous cloaked region (Figure 1 - right side). The MAB is computed by extending the current cloaked region by a radius of 'r' as specified.

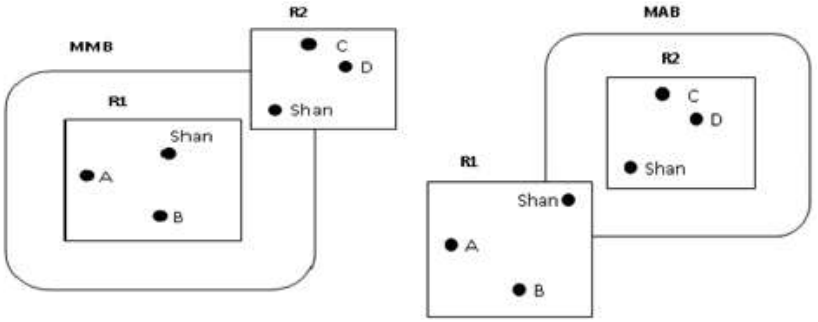

Figure 1. Location-dependent attacks

That is to say in continuous location updates, the mobile user's current and previous location can be predicted by LBS providers through historical information such as current and previous cloaked regions and the mobility pattern. This is known as location-dependent attacks (Pan et al., 2012; Shanthi & Balasundaram, 2014) and can be described in Equations 2 and 3:

Current location (user) = Area (MMB ∩ Current CR)

$$\text{Current location (user)} = \text{Area (MMB} \cap \text{Current CR)} \quad (2)$$

Previous location (user) = Area (MAB ∩ Previous CR)

$$\text{Previous location (user)} = \text{Area (MAB} \cap \text{Previous CR)} \quad (3)$$

Location-dependent attacks have been discussed in some existing work (Ghinita, Damiani, Silvestri, & Bertino, 2009; Xu, Tang, Hu, & Du, 2010). All these work only considered cloaking granularity as their privacy metric. The cloaking granularity alone preserves location privacy, but fails to protect user identity in case there is only one user in the cloaked region. To resolve this problem, Pan et al. (2012) proposed IncrementalCliqueCloak (ICliqueCloak) algorithm. This algorithm adopts both cloaking granularity as well as location k-anonymity as its privacy metrics and finds out the cloaked region for any user at time t_{i+1} within his/her MMB at time t_i. However, this algorithm suffers from 'out-of-time' query; that is to say, some queries cannot be cloaked within the time period. Therefore, the queries cannot be answered. To minimize the 'out-of-time' queries as well as to protect users from location-dependent attacks, an efficient cloaking algorithm called MClique-Dynamic is proposed in this study.

Algorithms such as CliqueCloak (Gedik & Liu, 2008), MMBClique, IClique and OptClique (Pan et al., 2012) locate the cloaked region for a user 'u' based on a privacy value (k). During the computation of the cloaked region, the algorithms group only neighbors whose privacy value is less than or equal to that of the request of the user, 'u'. So that, for queries with high privacy value, the waiting time (which plays a major role in computation processing time) of the queries getting cloaked is increased; also in some cases, the queries cannot be cloaked. Thus, this leads to a reduced success rate and increased processing time. These issues are effectively handled in the proposed MClique-Dynamic by defining the privacy value (minimum privacy value guaranteed is 2 and maximum privacy value guaranteed is the number of users' issued queries at time 't') dynamically and generating duplicate queries. Thus, MClique-Dynamic guarantees the improved success rate of queries getting cloaked by minimizing 'out-of-time' queries and reducing processing time.

In view of protecting users from location-dependent attacks and ensuring the improved success rate of queries getting cloaked and to reduce processing time of cloaked region generation, in the proposed MClique-Dynamic, maximum sized clique is determined using greedy heuristic within the MMB of users. At the same time, this algorithm incorporates duplicate query generation in addition to location k-anonymity, reciprocity (the cloaked region contains u and at least k-1 additional users, also every user in a cloaked set generates the same cloaked set for the given k-value) and cloaking granularity (the area of the cloaked region which is larger than the user-specified threshold value) properties. We conducted a series of experiments to evaluate the performance of the proposed MClique-Dynamic algorithm using a synthetic dataset generated with the help of the mntg traffic generator (Mokbel et al., 2013).

OVERVIEW OF MCLIQUE-DYNAMIC ALGORITHM

The proposed MClique-Dynamic algorithm finds out the MMB first for each user upon the arrival of location-based queries from various users at a given time period. Then it computes the neighbors. Both users and their neighbors are modeled in an undirected graph. Next, the algorithm checks whether a clique set that satisfies location k-anonymity is formed in the modeled undirected graph. If any such clique set is found, then the minimum bounding rectangle (MBR) of the clique set is compared with A_{min} and A_{max} (minimum and maximum area covered by the cloaked region) values. The MBR (which will be given as a cloaked region) should be greater than or equal to A_{min} in order to prevent location disclosures.

Consider a case in which the clique set satisfies the location k-anonymity but fails to satisfy cloaking granularity. In this case, the MBR of the clique set is extended in such a way that it satisfies the cloaking granularity. Similarly, if the cloaked region exceeds A_{max} then it will be reduced to be equal to A_{max} in order to avoid the larger cloaked region. The above mentioned criteria are stated as follows:

Let MS = $\{u_1, u_2 \ldots, u_n\}$ be a user set that forms the clique, MBR is $R_{u,ti}$ at time, t_i and S_u is the speed of a user. The previous cloaked region of each user u is denoted by $R_{u,ti-1}$. The set MS is a cloaking set if and only if it satisfies the following conditions for any user u in MS:

1. $\text{Distance}(R_{u,ti}, R_{u,ti-1}) \leq S_u \cdot (t_i - t_{i-1})$
2. $\text{Distance}(R_{u,ti-1}, R_{u,ti}) \leq S_u \cdot (t_i - t_{i-1})$
3. $|MS| \leq 2$
4. $\text{Area}(MBR(MS)) \geq A_{min}$
5. $\text{Area}(MBR(MS)) \geq A_{max}$

In order to defend against location-dependent attacks, the users in the cloaked region must satisfy the first two conditions. The third condition states the location k-anonymity requirement to protect user identity. Finally, the fourth and fifth conditions ensure that the area of the cloaked region is not too small (populated area) and not too big, respectively. Here, distance (R_i, R_j) is the Euclidean distance between a point in R_i and its closest point in R_j.

There is a possibility for a situation in which the location k-anonymity property may not be satisfied. In this situation, the anonymizer generates duplicate queries with random users whose location-context is bounded within the MMB of the actual query issuer. The results returned in the duplicate queries are cached by the anonymizer for future use. By this random duplicate generation, the anonymizer maintains k-anonymity value as a minimum of 2. In the following section, a detailed discussion on steps associated with cloaked region construction is described to effectively and protectively answer location-based queries.

PRIVACY PROTECTION

Procedure for protecting the location privacy of users by generating k-anonymous cloaked regions using MClique-Dynamic is given in the following subsections.

Location Anonymization in Dynamic Context

In movement (continuous location updating), instead of one single query with snapshot location, a list of queries with different snapshot locations are sent to the LSP. In this scenario, if a cloaked region is constructed for each snapshot location using existing clique based cloaking algorithm, a malicious LSP can easily uncover the real location. For instance, assume a user uses the LBS app at 9.00 a.m. for obtaining a route suggestion for a particular destination. Later at 9.20 a.m., he/she issued the query about the POI (fuel station) during mobility. In order to protect the user, existing CliqueCloak generates two separate cloaked regions (without considering any query dependency) CR_1 and CR_2 at time 9.00 a.m. and 9.20 a.m., respectively. From the cloaked regions, the LSP can easily conclude the traveling route as well as the current location of the user.

Algorithm 1: MClique-Dynamic

INPUT : Set of location context of the users- FP
OUTPUT : Cloaked Region- CR
Step 1: **begin**
Step 2: **for** i←1 to | FP | **do**
Step 3: construct MMB
Step 4: **end for**
Step 5: **for** i←1 to | FP | **do**
Step 6: compute neighbor
Step 7: **end for**
Step 8: construct graph G ← (V, E)
Step 9: **for** i←1 to | V | **do**
Step 10: find clique existence with its neighbors
Step 11: **if** (clique exist) **then**
Step 12: **if** Amin ≤ MBR(clique) ≤ Amax **then**
Step 13: CR← MBR(clique)
Step 14: V←V ¯ {vertices associated with the clique }
Step 15: E← E ¯{ edges associated with the clique }
Step 16: send CR as the location-context of users in the clique to the LSP
Step 17: **else if** (MBR(clique) < A_{min}) **then**
Step 18: extend MBR(clique) in such a way that Area(MBR(clique)) ←A_{min}
Step 19: **else**

(continued)

Step 20:	reduce MBR(clique) in such a way that Area(MBR(clique) ← A_{max}
Step 21:	**end if**
Step 22:	**else**
Step 23:	Generate duplicate queries from dummy locations that lie within the MMB and returns to the area comprising the actual query issuer and duplicate queries as the cloaked region
Step 24:	**end if**
Step 25:	**end for**
Step 26:	**end**

The reason is that the current location of the user must be limited to the MMB of the CR_1 (previous cloaked region). As per the CliqueCloak, the CR_2 (current cloaked region) may or may not be completely residing inside the MMB. If the CR_2 is not completely residing inside the MMB, then the user must be in the intersection area of the MMB and CR_2. Hence, by correlating the MMB, previous and current cloaked regions, the service provider can easily pinpoint where the user is currently located. Similarly, the previous location of the user can also be determined. To prevent users from this type of attack, CliqueCloak has been modified in order to accommodate continuous location updating.

When a user (while in movement) wants to make a request to LBS, the anonymizer hides the location-context using clique based cloaked region construction algorithm in such a way that the current cloaked region always completely resides inside the MMB. For that, whenever a query is issued by various users, the anonymizer computes the MMB (as mentioned above) and then determines the neighbors. For a user 'u', the neighbors are the ones where the MMB is contained in each other. Next to neighbor computation, graph construction, clique computation and cloaked region construction are performed. The pseudo code for the proposed MClique-Dynamic cloaking is given in Algorithm 1.

With the computation of the clique, the size of the problem is reduced by 1 every time the clique is not formed. This is expressed by the recurrence Equation 4 as follows:

$$T(n) = T(n-1) + f(n) \quad (4)$$

n = number of mobile users issued the queries

T = time to compute the clique

The function f(n) accounts for the time needed to reduce an instance to a smaller one and to extend the solution of the smaller instance to that of

a solution of the larger instance. Applying backward substitution, the above equation are peformed by Equations 5 and 6.

$$T(n) = T(n-1) + f(n) \quad (5)$$

$$= T(n-2) + f(n-1) + f(n)$$

$$\vdots$$

$$= T(n-k) + f\left(n-(k-1)\right) + \cdots + f(n) \quad (6)$$

In the worst case, the problem size should be reduced to two; hence, substitute in the above equation 6, yield Equations 7 and 8.

$$T(n) = T(2) + f(-3) + \cdots + f(n) \quad (7)$$

$$= T(2) + \sum_{i=0}^{n} f(i) + \sum_{i=-3}^{-1} f(i) \quad (8)$$

In order to verify the pairwise connection between the users, the algorithms have two for-loops each with n iterations. Hence, f(i) takes n^2 time as in Equations 9.

$$\sum_{i=0}^{n} n^2 = n^3 \quad (9)$$

Thus, in the worst case scenario, clique computation takes $O(n^3)$ time and in the best case scenario, clique computation takes only $O(n^2)$.

Dummy Location-Contexts Generation

For the generation of duplicate queries, two important pieces of information are needed. One is service specific information (SSI) and another one is the location-context from where the query is posed to the LSP. Both the information can be generated at random. The random generation of SSI may not affect users' location privacy. However, the random generation of location-contexts may reveal users' location privacy. The reason is the distribution of the location-contexts. For example, if k-1 location-contexts are chosen randomly within the MMB, there may be a possibility for these k-1 positions to be very close to the user's position in which the privacy region is generated. This can have negative consequences such as (a) malicious LSP will probably ignore outliers and may deduce the user's location-context with high certainty and (b) the privacy area that is the cloaked region generated may be a smaller

region and therefore the malicious LSP can easily pinpoint the user's location-context.

To counter these problems, some control is needed over the distribution of dummy location-contexts while generating duplicate queries. For this purpose, modified grid based dummy generation algorithm called DLC (Dummy Location-Context) is proposed and is described as follows.

A uniform square grid is created with an area (A) equal to the MMB of the user (to whom the CR is generated), and is composed of n vertices. Among the n vertices, one is the user position p, the other n-1 vertices will be used as dummy locations for duplicate query generation. The pseudo code of the DLC is given in Algorithm 2, and steps involved in the algorithm are stated as follows.

Step 1: Calculate the number of vertices, n as the square root of G, where G is a square number (the value of G is taken by the anonymizer).
Step 2: Attach the user position, p to one of the vertices, by randomly generating x, y indices.
Step 3: Determine the side length of the grid cells, L_s by dividing the area of the grid (A) by the square root of G.
Step 4: Calculate the position for each grid vertex in row-major layout.
Step 5: Select k-1 number of vertices and enter its position into the dummy location-context array K_d.
Step 6: Return K_d.

Algorithm 2 : DLC

INPUT: p-user location-context, k-privacy value, A-MMB of the user
OUPPUT: K_d- array of k numbers of location-contexts
Step 1: **begin**
Step 2: initialize $K_d \leftarrow$ {p};count $\leftarrow$1
Step 3: $n \leftarrow \sqrt{G}$
Step 4: $x \leftarrow rand(1,n); y \leftarrow rand\ (1,n)$
Step 5: $L_s \leftarrow \sqrt{A}/n$
Step 6: **for** $i \leftarrow 1$ *to n* **do**
Step 7: **for** $j \leftarrow 1$ *to n* **do**
Step 8: $x_1 \leftarrow$(i –x)· L_s +p_x
Step 9: $y_1 \leftarrow$(j –y)· L_s +$p.y$
Step 10: $K_d \leftarrow K_d \cup (x_1, y_1)$
Step 11: *count* $\leftarrow$*count* +1
Step 12: **if** *count* $\leq k$ **then**
Step 13: continue

(continued)

Step 14: **end if**
Step 15: **end for**
Step 16: **end for**
Step 17: **return** K_d
Step 18: **end**

PERFORMANCE EVALUATION

We have implemented the proposed MClique-Dynamic cloaking algorithm using Java, executed on Intel(R) Core (TM) 2 Duo 2.00GHz machine with 3.00 GB of RAM and Windows 7 Operating System. For experimental purposes, approximately 2.5 MB of data was generated with the help of the mntg traffic generator (http:// mntg.cs.umn.edu). The input parameters of the generator are shown in Table 1, and the simulated traffic pattern is shown in Figure 2.

Table 1

Input parameters of mntg traffic generator

Search Area	Tamilnadu
Traffic model	Brinkhoff
Starting vehicles	100
Simulation time	20
Additional vehicles each time unit	10

Traffic Results

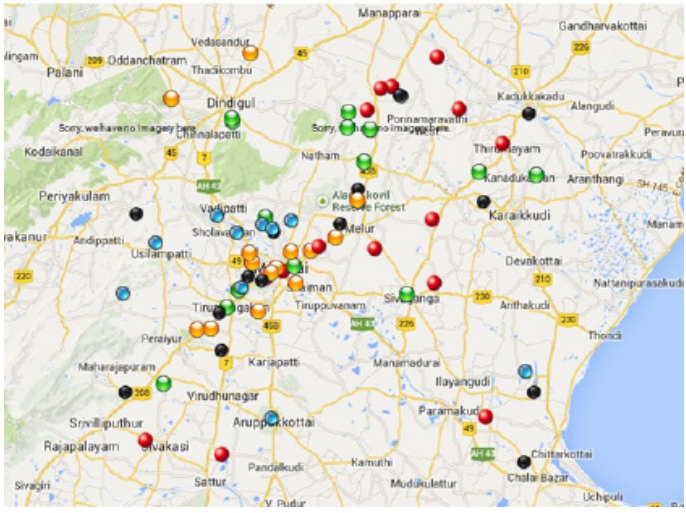

Figure 2. Traffic pattern simulation.

Results and Discussion

The proposed MClique-Dynamic algorithm is compared with three algorithms, namely MMBClique, OptClique, and IClique. MMBClique refers to the revised version of the cloaking algorithm proposed by Gedik and Liu (2005). In this algorithm, the tolerable maximum cloaked region is replaced with the MMB to prevent from location-dependent attacks; so that it generates larger cloaked regions as users' privacy area. Thus MMBClique increases cloaking time. In order to reduce cloaking time, Pan et al. (2012) has proposed IClique and OptClique.

IClique adopts both cloaking granularity as well as location k-anonymity as its privacy metrics and finds out the cloaked region for any user at time t_{i+1} within his/her MMB at time t_i. Hence compared to MMBClique, IClique can generate a smaller cloaked region to serve as users' privacy area. Thus, IClique reduces cloaking time. The optimized version of IClique is OptClique algorithm which further reduces the size of a cloaked region. In this algorithm, the MMB is set to infinity. Even though this algorithm does not protect users from location-dependent attacks, it is included for comparison to show the cost required for defending against location-dependent attacks. However, all the above mentioned algorithms (MMBClique, IClique, and OptClique) suffer from 'out-of-time' queries. Therefore, the success rate of the queries getting cloaked is reduced. This problem is effectively handled by the proposed MClique-Dynamic cloaking algorithm and is shown in the following experimental results.

The experimental parameters are shown in Table 2, and for evaluation purposes, metrics such as success rate, time to generate dummy location-contexts (DLC) and cloaking time are used. As shown in Figure 3, the success rate of IClique and OptClique decrease with much more diversified privacy levels. They reach 89% and 87%, respectively when the privacy level reaches 25. The reason is that the attempts by both IClique and OptClique to find out cloaked regions by incrementally maintaining maximal clique takes more time to cloak users' requests for larger k values. Therefore, some queries would have expired before they are cloaked successfully, and this is reflected in the reduced success rate.

In addition, the success rate of MMBClique decreases significantly with increasing k. Its success rate drops to about 66% when the privacy level reaches 25. The main reason is that MMBClique finds the cloaking set only from neighbors whose privacy levels are less than that of the new request. Thus, a request with a higher privacy level is difficult to be cloaked successfully in the MMBClique. In contrast, MClique-Dynamic has the best performance. Its success rate is about 99%, even with increasing k values. This is mainly

because MClique-Dynamic employs duplicate query generation to find a cloaking set with a larger k-value and in situations where sufficient requests are not available for cloaking. As a result, all the requests are cloaked before they expire. Therefore, this indicates a significant difference in the cloaking algorithm of the proposed MClique-Dynamic.

Table 2

Experimental Parameters

A_{min}	$500m^2$
A_{max}	$4\text{-}6km^2$
Speed	50km/h
Minimum no. of duplicate queries generated	2

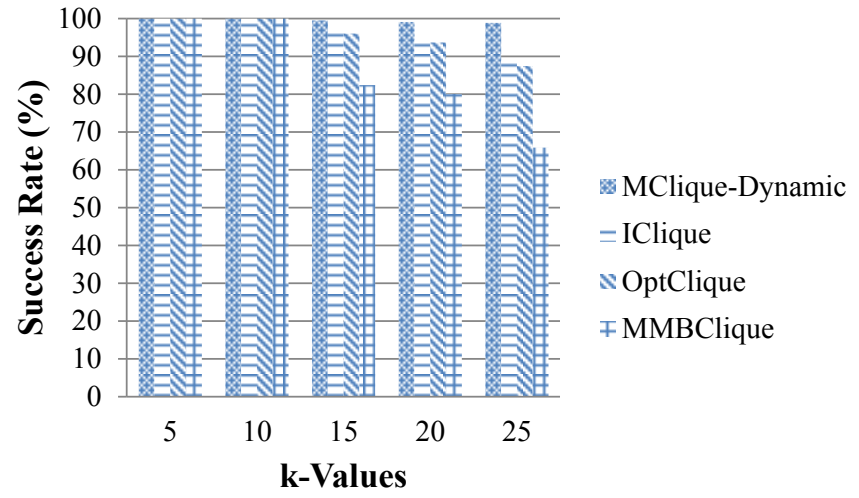

Figure 3. Success rate.

Figure 4 shows the cloaking time of all algorithms. In most cases, waiting time dominates the overall cloaking time of cloaked region generation, and cloaking time increases with increasing k values. In particular, MMBClique cannot scale up to a larger privacy level and its processing time gets worse dramatically. In contrast, IClique and OptClique require a much shorter processing time than MMBClique. The reason is that both these algorithms can quickly find the cloaking set from the set of incrementally maintained maximal cliques.

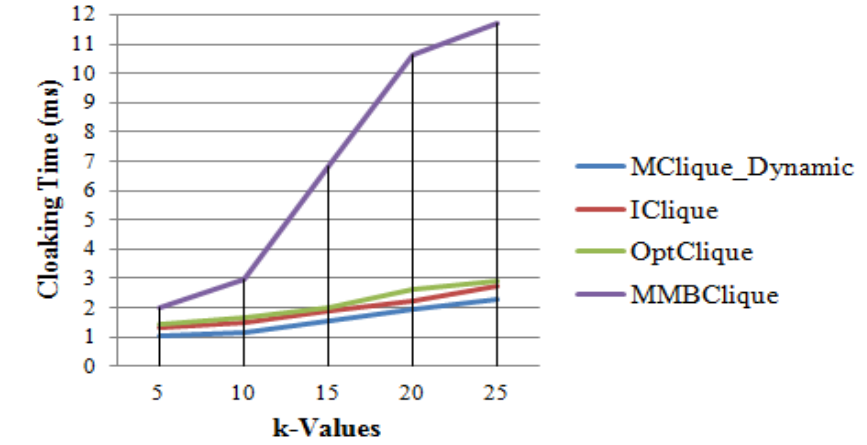

Figure 4. Cloaking Time.

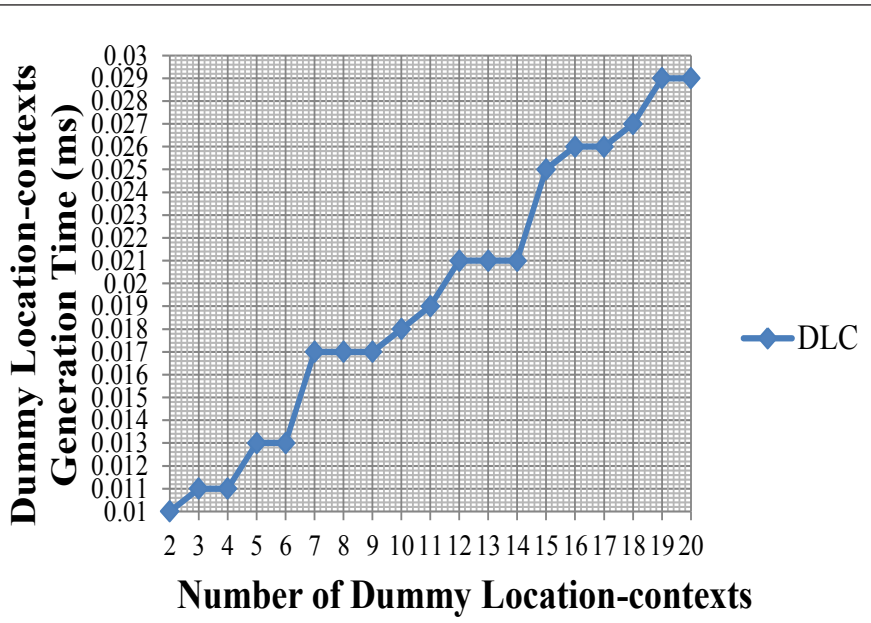

Figure 5. Dummy Location Generation Time.

Although the proposed MClique-Dynamic algorithm employs DLC to protect users from extreme situations, the cloaking time of the proposed MClique-Dynamic can be further reduced. The reason is that the duplicate queries are generated in a reasonable amount of time (in milliseconds). Figure

5 show that the DLC takes only 0.029ms when the privacy value reaches 25. This signifies that the proposed location cloaking algorithm does not delay cloaking time. Besides, the proposed algorithm speeds up the process of finding the cloaking set only from neighboring nodes, thereby eliminating time taken to incrementally maintain maximal cliques.

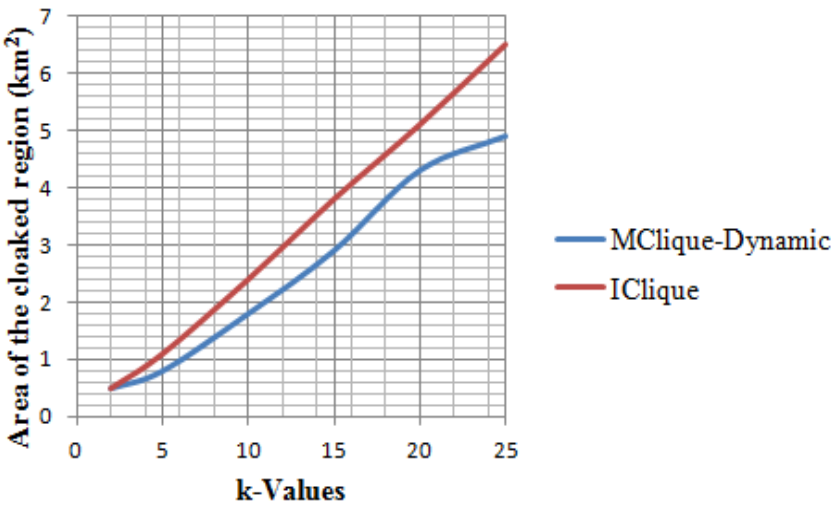

Figure 6. Spatial Resolution.

Figure 6 compares the size of the cloaking areas of IClique and MClique-Dynamic. As can be seen, cloaking areas are less than 2km^2 for lower privacy values (k=5), less than 4.5 km^2 for medium privacy values (k=10 or k=15) and less than 5km^2 for higher privacy values (k=20 or k=25). That is MClique-Dynamic generates a smaller (better) region than IClique when privacy values increase. This is mainly because MClique-Dynamic imposes two threshold values (A_{min} and A_{max}) while seeking out the cloaked region, but IClique employs only one threshold value (A_{min}).

Figure 7 shows that the cloaked region computation time varies for varying A_{max} (1-3km^2, 4-6km^2, and 7-9km^2) values. It is observed that the higher the A_{max} value, it results in higher cloaked region computation time. The reason is that cloaked region computation time includes the time to compute neighbors, construct MBR and check for clique existence. Hence, when the A_{max} value is increased, there may be a chance of getting more numbers of users inside the MBR, thereby increasing the time taken for checking clique existence. This is reflected in an increase in cloaked region computation time.

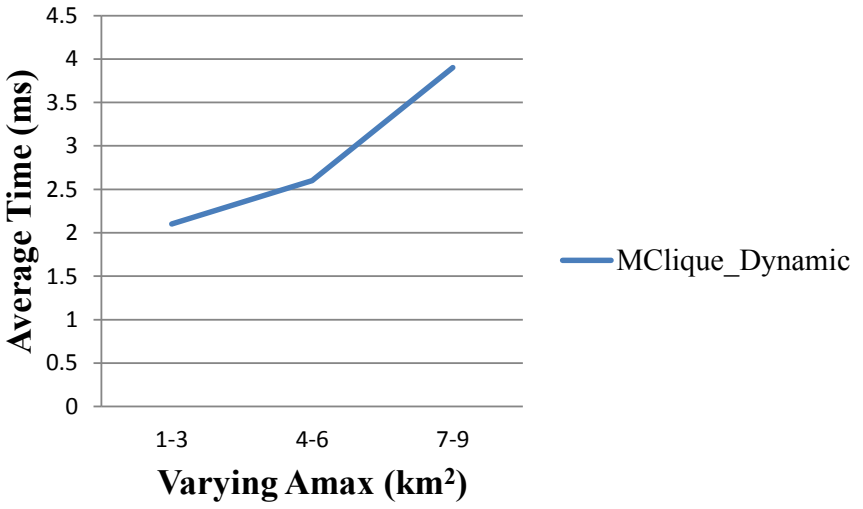

Figure 7. Different Amax Values.

In addition, threshold values influence the size of the CR. Since the size of the CR is restricted by the threshold values, spatial resolution is affected by A_{min} and A_{max} values. A larger A_{min} and A_{max} generate a larger region as users' privacy area. Similarly, a smaller A_{min} and A_{max} lead to a smaller CR. Therefore care must be taken while choosing threshold values. In the experiment, better cloaking is achieved when selecting A_{max} at 4-6km^2.

The positive side of the proposed MClique-Dynamic is a) minimization of 'out-of-time' queries and b) generation of CR that lies between two threshold values A_{min} and A_{max} (i.e. $A_{min} \leq$ cloaked area $\leq A_{max}$). Through minimizing 'out-of-time' queries, most of the queries are successfully cloaked and by using threshold values the anonymizer prevents the generation of cloaked regions which are neither too small nor too large. Intuitively, a smaller CR brings better performance in terms of storage cost and computation, while a larger CR makes it better from the privacy point of view.

Despite this, regions which are too small improve the level of certainty of locating users and therefore the location privacy of users is violated. On the other hand, regions which are too large may increase processing cost of the service provider which in turn leads to a reduction in the quality of query services. Thus, the proposed MClique-Dynamic algorithm balances both the privacy of users and quality of query services in terms of generating reasonably-sized CRs. In addition, it preserves the two main ingredients of cloaking namely a) location privacy—unable to locate user and b) query anonymity—unable to identify person who sent query.

CONCLUSION

In this paper, we have presented the MClique-Dynamic, cloaking algorithm which safeguards the location privacy of users from location-dependent attacks. To obtain an improved success rate, duplicate queries are generated using DLC algorithm while seeking out the cloaking set. In addition, two threshold values are used to avoid the generation of CRs which are either too small or too large. As these would reduce the certainty level of locating users in populated areas, and also maintain the quality of query services. Through generating duplicate queries and reasonably-sized CRs, the proposed MClique-Dynamic is capable of protecting users' location privacy better than existing cloaking algorithms such as IClique, OptClique, and MMBClique. The improvements are shown in the experimental results using synthetic dataset generated by the mntg traffic generator.

Existing solutions to cloak users' location-contexts may degrade the usefulness of LBS applications when more numbers of requests emerge from people in different places. The reason is when query density that is user density is increased, the graph constructed by the anonymizer may become a dense graph and processing the dense graph by an anonymizer will take more time. One way to speed up the anonymization of location-context of various users is partitioning the user graph into separate units of smaller sub-graphs which can be mapped onto parallel processors of anonymity servers hosted on trusted cluster computing environment bases. In accord with this, we propose a graph partitioning based spatial cloaking algorithm in our future work.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for profit sectors.

REFERENCES

Bamba, B., Liu, L., Pesti, P., & Wang, T. (2008). Supporting anonymous location queries in mobile environments with privacy grid. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 237-246). ACM.

Biswas, P., & Sairam, A. S. (2017). Modeling and Performance Comparison of Privacy Approaches for Location Based Services. arXiv preprint arXiv:1711.04974.

Cheng, R., Zhang, Y., Bertino, E., & Prabhakar, S. (2006). Preserving user location privacy in mobile data management infrastructures. *Privacy Enhancing Technologies* (pp. 393–412). Springer: Berlin Heidelberg.

Chow, C. Y., & Mokbel, M. F. (2009). Privacy in location-based services: A system architecture perspective. *Sigspatial Special, 1*(2), 23–27.

Du, J., Xu, J., Tang, X., & Hu, H. (2007). iPDA: Supporting privacy-preserving location-based mobile services. In 2007 International Conference on Mobile Data Management (pp. 212–214). IEEE.

Gedik, B., & Liu, L. (2005). *Location privacy in mobile systems: A personalized anonymization model.* In Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS 2005) (pp. 620–629), IEEE.

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

Ghinita, G., Damiani, M. L., Silvestri, C., & Bertino, E. (2009). Preventing velocity-based linkage attacks in location-aware applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 246–255). ACM.

Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., & Tan, K. L. (2008). Private queries in location based services: Anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 121–132). ACM.

Gruteser, M., & Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services* (pp. 31–42). ACM.

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

Kuang, L., Wang, Y., Ma, P., Yu, L., Li, C., Huang, L., & Zhu, M. (2017). An Improved Privacy-Preserving Framework for Location-Based Services Based on Double Cloaking Regions with Supplementary Information Constraints. Security and Communication Networks.

Lin, Y. M., Zakariah, M. I., & Mohamed, A. (2010). Data leakage in ICT outsourcing: Risks and countermeasures. *Journal of ICT, 9*, 87–109.

Mokbel, M. F., Chow, C. Y., & Aref, W. G. (2006). The new Casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases* (pp. 763–774). VLDB Endowment.

Mokbel, M. F., L. Alarabi, J. Bao, A. Eldawy, A. Magdy, M. Sarwat, ... and S. Yackel. (2013). MNTG: An extensible web-based traffic generator. In *International Symposium on Spatial and Temporal Databases* (pp. 38–55), Springer: Berlin, Heidelberg.

Nguyen, H. H. (2017). *MeshCloak: A map-based approach for personalized location privacy.* arXiv preprint arXiv:1709.03642.

Pan, X., & Meng, X. (2013). Preserving location privacy without exact locations in mobile services. *Frontiers of Computer Science, 7*(3), 317–340.

Pan, X., Xu, J., & Meng, X. (2012) Protecting location privacy against location-dependent attacks in mobile services. *IEEE Transactions on Knowledge and Data Engineering, 24*(8), 1506–1519.

Shanthi, P., & Balasundaram, S. R. (2014, November). An Efficient Clique Cloak Algorithm for Defending Location-Dependent Attacks in Location Based Services. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies* (p. 22). ACM.

Shanthi, P., & Balasundaram, S. R. (2015). A graph-based cloak algorithm to preserve location privacy in location-based services. *International Journal of Information Privacy, Security and Integrity, 2*(2), 138–158.

Sweeney, L. (2002). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10*(05), 571–588.

Um, J. H., Kim, H. D., & Chang, J. W. (2010). An advanced cloaking algorithm using Hilbert curves for anonymous location based service. In *2010 IEEE Second International Conference on* Social Computing (SocialCom), (pp. 1093–1098). IEEE.

Xu, H., Teo, H. H., Tan, B. C., & Agarwal, R. (2009). The role of push-pull technology in privacy calculus: The case of location-based services. *Journal of Management Information Systems, 26*(3), 135–174.

Xu, J., Tang, X., Hu, H., & Du, J. (2010). Privacy-conscious location-based queries in mobile environments. *IEEE Transactions on Parallel and Distributed Systems, 21*(3), 313–326.

Xu, T., & Cai, Y. (2007, November). Location anonymity in continuous location-based services. *In Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems* (p. 39). ACM.