PRIVACY PRESERVATION AND SERVICE PROVISION

IN

GEO-SPATIAL-TEMPORAL SYSTEMS

By

PRADEEP BEJGAOM

Bachelor of Technology in Computer Science

Jawaharlal Nehru Technological University

Hyderabad, Andhra Pradesh, India

2010 - 2014

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2016

PRIVACY PRESERVATION AND SERVICE PROVISION

IN

GEO-SPATIAL-TEMPORAL SYSTEMS

Thesis Approved:

Dr. Johnson P Thomas

Thesis Adviser

Dr. Christopher Crick

Dr. David Cline

Name: PRADEEP BEJGAOM

Date of Degree: December, 2016

Title of Study: PRIVACY PRESERVATION AND SERVICE PROVISION IN GEO-SPATIAL-TEMPORAL SYSTEMS

Major Field: COMPUTER SCIENCE

Abstract: The main aim of Location Based Services (LBS) is to ensure user privacy while providing the best LBS for users. There are very few works focusing on the tradeoff between location privacy preservation and location query information gathering which are used to improve the LBS in future. Privacy preserving Location Query (PLQ) is one such technique, however its quality of service is very low and its inability to identify the unreachable areas in the user obfuscation area can cause problems to user privacy. In addition to this, the untrusted Location Based Server (LBSr) can malfunction and perform other attacks like maximum movement boundary attack, shrink region attack, user query privacy attack to identify or track the user.

In this thesis, we propose the Privacy preserving Geometric Aware Location Query (PGALQ) algorithm which collects the aggregate information of user queries and also provides geometric aware obfuscation by eliminating unreachable areas. We also extend PGALQ with Pseudonym and Multiple Query Confusion Algorithm (PMQCA) to prevent LBSr to launch the above mentioned attacks. The results of the proposed algorithm is compared with the algorithms mentioned in the literature which shows that our algorithm performs efficiently.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 MOTIVATION

Users can issue location based queries according to their needs, such as i) where is the nearest gas station?, ii) what is the journey time to New York?, iii) Best Indian Restaurant near me? etc. These queries are answered according to their location where they issued the query. But these Location Based Services (LBS) can cause problems to user privacy, if they are not used with utmost care. The main aim of LBS is to ensure user privacy while providing the best LBS for users.

Spatial Obfuscation [7] or Cloaking Granularity is one such technique, which hides the user's original position from the Location Based Server (LBSr) while at the same time providing service to the user query. The idea is to expand the area where the user may be located before sending the query, in order to hide the user's real position. The larger the area, the more difficult it becomes to pinpoint user location.

Spatial Obfuscation alone can protect user spatial information [9][10] but it cannot protect user-identity. To protect user-identity many researchers have come with different solutions of which K-Anonymity [7] is notable.

K-Anonymity is a well-known general privacy concept not restricted to location query. It ensures identity protection by combining the original user who requested the LBS query with queries of other k-1 users in the same area before sending the query to LBSr. Once the LBSr gets the query, it has k users and therefore it cannot differentiate the original user who requested the service. The combination of these 2 techniques will become a secure privacy measure in Location Based Services and many techniques have been proposed based on this.

In addition to providing privacy, queries to LBS can help to improve the quality of service. Providing the aggregate search data of users to LBSr will help to improve services in future. For example, consider a location where there are many user queries for McDonald's which is not present in that location. That means, people in that location are showing much interest in McDonald's and a new branch is needed there. If such aggregate information can be provided for LBSr or Service providers, they can improve the services in the future based on user interests instead of uninformed estimations.

In the literature, many techniques have been proposed for privacy of users but very little work has been done to provide search data to service providers in order to improve services. PLQ (Privacy Preserving Location Query) [1] is one such technique which does privacy preserving and also collects aggregate information using a map-hierarchy. PLQ is presented in section 2.1.

The B^{ob}- tree is a geographic aware obfuscation algorithm [2] which is proposed to eliminate the problems caused by geometric obfuscation. Geometric obfuscation techniques deal with only the geometry of the obfuscated region and expands the area in any geometric shape without considering what is included in the geometric area. Hence attackers can eliminate the unreachable portions from the obfuscated area thereby reducing the area and hence increasing

the probability of inferring the user's exact position within the obfuscated area. The B^{ob}- tree is proposed to deal with such problems and is presented in section 2.2.

In Multiple position attacks, the attacker tracks and correlates several position updates or queries of the same user to decrease his privacy. The Maximum Movement Boundary attack (MMB) [5] is one such attack. The attacker calculates the maximum movement boundary of the user from all the extreme positions, where the user could have moved between two succeeding position updates or queries. And immediately when the next request is issued by the user, then he intersects the obfuscated area of the new request with the calculated maximum Movement boundary area. The intersection area is the area where the user can be found at the time the new request is issued by the user. This is presented in section 1.3.

In this thesis, we propose a Privacy preserving Geometric Aware Location Query (PGALQ) which is based on PLQ to solve the flaws of PLQ by incorporating geographic aware obfuscation. Just like PLQ, we divide a particular area to fixed sizes (here, into number of counties) to count the number of user queries generated in the partitioned regions. Service providers generally want to know the trends in larger areas (cities, counties) than smaller areas (streets) to invest money and improve services. But, unlike PLQ we don't obfuscate the entire region, instead we incorporate geographic aware obfuscation to eliminate unreachable portions from the obfuscated area to obtain the user desired level of privacy. PLQ also doesn't answer for multiple position attacks like Identity matching attack [7] and MMB attack [5][6]. Hence, we extend PGALQ to propose a new algorithm called Pseudonym and Multiple Query Confusion Algorithm to deal with such attacks, since the existing solutions cannot be incorporated into PGALQ.

1.2 Obfuscation and Problems

Spatial Obfuscation tries to preserve location privacy by hiding the exact location of the user before sending the user's query to the Location Server. This is achieved by expanding the area where a user may be located. The larger the area, the more difficult it becomes to pinpoint the location of the user. Traditional obfuscation techniques including geometric obfuscations are more vulnerable to map-matching attacks, where an adversary can eliminate all un-reachable points from the obfuscated area by using his external map knowledge as shown below in figure 1. Furthermore, by using the semantic information provided by the map, such as points of interest or type of building, the obfuscation area can be reduced, which helps in finding the user accurately.

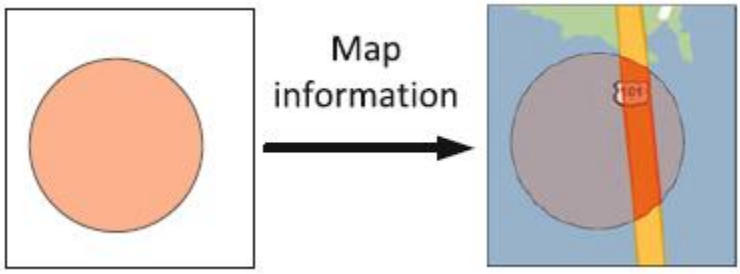

Figure 1: Map-Matching Attack

Another big disadvantage of spatial obfuscation is that, it alone cannot provide privacy for users. For example, consider a case where Alice obfuscates her area and sends a query to find the nearest hospital to that area. The service provider may not know the exact position of Alice but he can infer that, Alice must have a health problem since the query is for the nearest hospital. Such information is sensitive and no user may want to disclose it. To solve this problem, the obfuscation technique is combined with K-anonymity where combine the original user's query

with the queries of other k-1 users present in that obfuscation area and then send the queries. Now, the service provider cannot know who among those k-users has queried and who has health problems. To do this, we need a Trusted Third Party (TTP) to add the other k-1 users who are present in that obfuscated area before sending the query to an un-trusted service provider. In addition to obfuscation, we have to use TTP to produce k-anonymity. TTP finds the other k-1 real users and forms a area containing those k-1 users along with original user and forms an obfuscation area.

1.3 Max-Movement Boundary Attack

When users request multiple queries to get Location Based Services, then the attacker performs a Maximum Movement Boundary Attack [5] to deduce sensitive information like location of the user by using knowledge about the user. This may include maximum speed etc. and the attacker can reduce the obfuscated area as shown in the below figure.

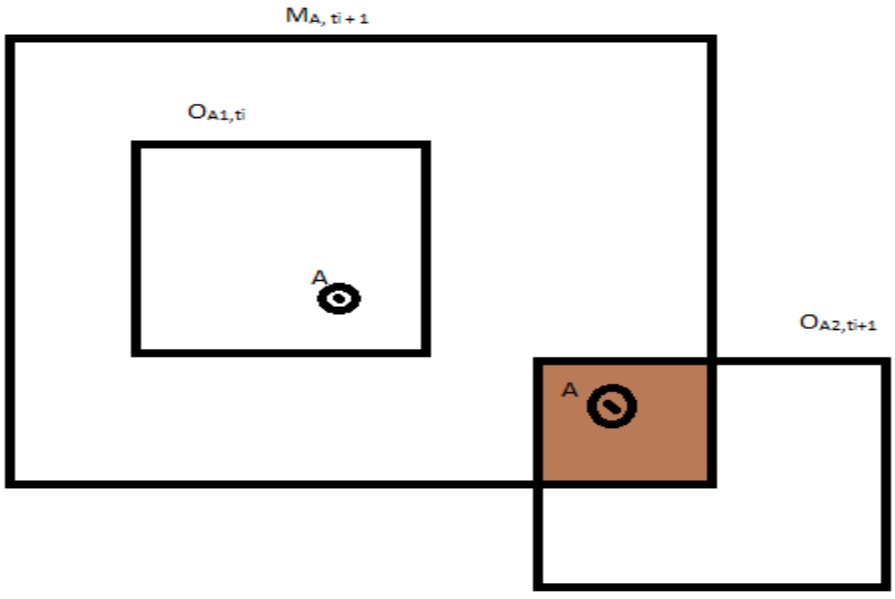

Figure 2: Maximum Movement Boundary Attack

- Let, 'A' be the original user who sent the query to the Location Based Server (LBSr). $O_{A1,ti}$ is the obfuscated region of the user at time 'ti' and $O_{A2,ti+1}$ is the obfuscated region of the user at time 'ti+1'.When the LBSr gets request $O_{A2,}$ then he calculates time Δt , i.e. the time between the new query and the previous query $\Delta t = (ti+1) - ti$. He calculates the maximum movement Boundary (MMB) based on map information and maximum speed information. Let MMB be $M_{A,ti+1}$ and this is intersected with $O_{A2,ti+1.}$ The intersecting area is the area where the user is found.
- To deal with such attacks, [5] proposed two alternatives.1) Request Deferral and 2) Postdating. But these approaches only considered the cloaking granularity as the privacy metric and they fail to protect user identity in the case there is only one user in the cloaked region.

We study the other alternatives proposed in the literature to deal with this problem in section 2. However, these approaches cannot be used in the proposed idea, which is explained in section 3.2.1.

1.4 PROPOSED APPROACH

The proposed approach collects the aggregate information of queries (as explained in section 1.1) requested by users which may be used for the betterment of location based services and it also provides geometric aware obfuscation by eliminating unreachable regions in the obfuscated area. The proposed architecture is based on [1], a map-hierarchy algorithm which helps service providers to collect information about the location based query. But in our approach, we divide the region into fixed size partitions i.e. New York City into counties. This is because service providers show interest in collecting information to improve services based on a larger feasible area instead of street area which is very small.

In the proposed algorithm, we use a geographic aware obfuscation technique [2] where the obfuscated region does not contain unreachable areas and the obfuscated region completely belongs to a single partition. This is realized by a Trusted Third Party (TTP) Anonymizer, which acts as a bridge between users and service providers. We also propose a new Pseudonym and Multiple Query Confusion (PMQCA) algorithm that extends PGALQ, to deal with Multiple Sample attacks and location dependent attacks.

1.5 OUTLINE OF THE THESIS

The rest of this thesis proposal is organized as follows: Chapter 2 provides a literature review of PLQ, geographic-aware obfuscation and privacy protection method on Multiple Sample query attacks. Chapter 3 presents our proposed approach (PGALQ) to collect information and how geographic aware obfuscation is used to provide privacy and propose a new algorithm (PMQCA) to deal with Multiple Sample query attacks and location dependent attacks. Chapter 4 presents simulation results. Finally, Chapter 5 is the conclusion.

CHAPTER II

LITERATURE REVIEW

2.1 PRIVACY PRESERVING LOCATION QUERY (PLQ)

The issue of Privacy preserving location queries has attracted much research. However, there are few works focused on the trade-off between location privacy preservation and location query information collection. To address this, [1] proposed Privacy Preserving Location Query (PLQ), an efficient Privacy Preserving Location Query processing framework which also supports Location Query Information Collection.

Consider the below diagram.

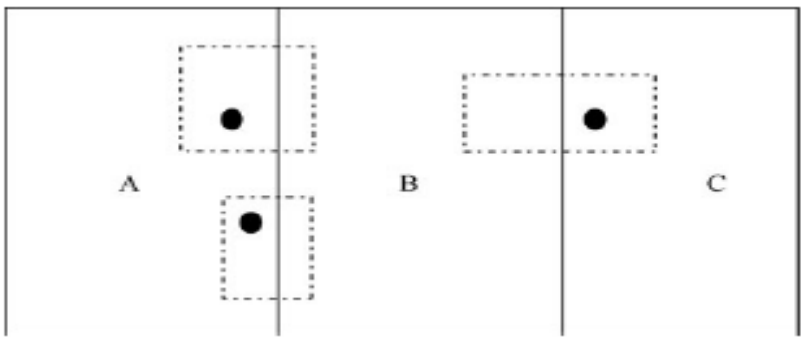

Figure 3: An Example of Cloaked Area using the traditional approach

Here, black dots represent the exact location information of the user and dashed rectangles represent the obfuscated area of the user. The LBS providers want to know which region issues the most queries among regions A, B, and C from fig.3

By observing the three queries, LBSr may incorrectly view that region B has issued all the queries and hence will think that region B requires more services which is not correct. To overcome this, they proposed a map-hierarchy method where the location of the two users in region A is obfuscated to the whole region A and the location of the user in region C is obfuscated to the whole region of C. So, now LBSr knows the collective requests of users from the respective regions and can provide or improve services efficiently without compromising the privacy of the user.

This paper assumes that a map-hierarchy is offered by LBS providers in advance. The content and the structure of the map-hierarchy depends on what information the LBS provider needs, such as, city A of a map should be divided into two or three parts. The requirements of an LBS provider decide how to divide the map. In general, the more detailed the map-hierarchy is, the better query information and services it can provide.

2.2 GEOGRAPHIC - AWARE OBFUSCATION

Obfuscation is the most popular technique in Location-Based Services that aims at protecting the privacy of personal location information. All the proposed techniques [11] before this paper are geometry based and they cannot assure location privacy when the adversary has knowledge about the geography of the obfuscated region. This problem is addressed by proposing the B^{ob} Tree [2], which is based on the B^{dual} Tree and contains geographic aware obfuscation on its nodes.

For example, consider the below diagram,

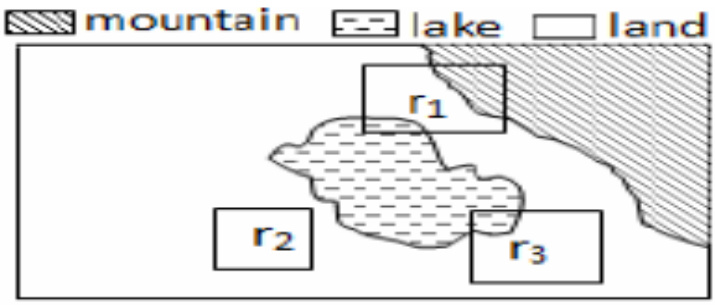

Figure 4: Example of unapproachable region

Here, r1, r2 and r3 are obfuscated regions of users. In r2 there is no unreachable regions (region where there is no possibility for a user to exist), r3 has an unreachable region lake and r2 has 2 unreachable regions because of the lake and mountain. If an adversary has map knowledge, then he can eliminate the unreachable areas and can now locate the user with more precision. Hence in [2], they proposed an approach where we identify the area lost by unreachable parts and add that much amount of reachable area before sending the query to the LBSr and protect the user with the same amount of obfuscation area as requested by the user.

2.3 ICliqueCloak Algorithm

Most of the existing privacy aware algorithms, which comply with location k-anonymity model are concerned with snapshot user locations only. They have not considered the effect of continuous location updates or Multiple Position Attacks. This may result in serious privacy breaches when different one-shot queries are frequently issued by the mobile user.

Location-dependent attacks have been studied in some previous works [5]. However, the prior solution only considered the cloaking granularity as the privacy metric and they may fail to protect user identity in the case there is only one user in the cloaked region. For example,

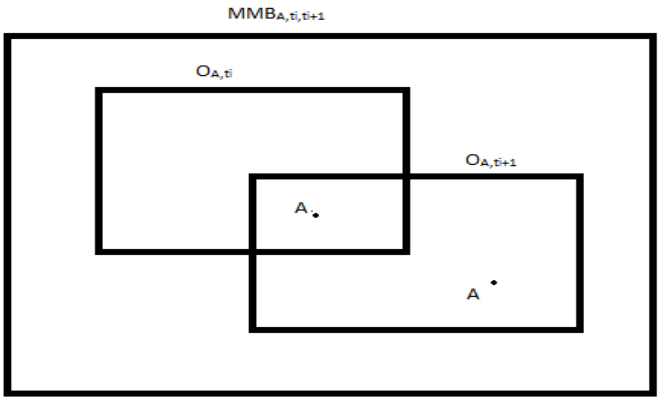

Figure 5: Example of Location-dependent Attacks

user 'A' has queried for the nearest hospital with obfuscated region $O_{A,ti}$ at time 'ti', and again he has queried for another super-specialty hospital at time 'ti+1'. Here, the cloaked region $O_{A,ti+1}$ is not subjected to the maximum movement boundary attack, but the adversary or location server can know that user 'A' has some health problem. This type of attack exposes the user's personal information which is sensitive information.

In [3], they adopted both the cloaking granularity and location k-anonymity as privacy metrics and proposed a new cloaking algorithm called ICliqueCloak to incorporate the effect of continuous location updates in the process of location cloaking. Consider the below diagram to understand this algorithm.

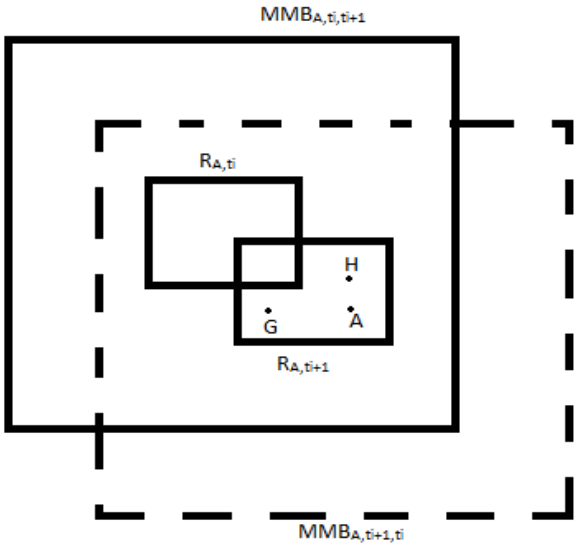

Figure 6: ICliqueCloak Algorithm

In this, the cloaking algorithm is aware of obfuscation area, $R_{A,ti}$ and $MMB_{A,ti,ti+1}$ and attempts to find the cloaked region for 'A' within $MMB_{A,ti,ti+1}$. Supposing that G and H are found in $MMB_{A,ti,ti+1}$ at t_{i+1} then A,G and H can form a cloaking set and generate a obfuscation area $R_{A,ti+1}$. Thus even if the attacker knows each user's speed limit, he/she still cannot tell the exact location of A in $R_{A,ti}$ and $R_{A,ti+1}$. Also the adversary cannot identify the person who has issued a query and identity of the user will not be at risk. They have used a graph model to formulate this problem.

2.4 N-DCM Model

Location based services should protect users information and the typical method is to protect the user's location information. Every user has specific privacy requirements like k-anonymity and minimum obfuscated area (Δs). For different privacy requirements of users, the model can set the

set up personal degree of anonymity, if the user requirements cannot be satisfied. In practice [4], the application of this model makes up for the shortcomings in the quality of service, unlike other approaches where they delay the service until the requirements are met. Consider the diagram below,

Figure 7: The Rectangle of the User's Rectangle

Let the users information set M={m1,m2,m3,m4,m5,m6} and the corresponding anonymity k={3,2,3,13,15,14}. ‘M’ is the set that represents 6 users who have queried for the service and ‘k’ is the set that holds corresponding k-anonymity requirement values of these 6 users. As shown in the above figure, the user’s information set (because m5 = 15 and there are only 6 users) cannot form any anonymous set according to the clique anonymity rule because there are only 6 users and 'd' requires the k-anonymity as 15 users. Because of this problem, in [4] the model uses k-means method to divide the set M into 2 groups as shown in the figure below.

Figure 8: Handle user's queries with k-means method

There are 2 sets, M_1= {m1 ,m2 ,m3} and set M_2= {m4, m5, m6}. ‘m5’ k-anonymity requirement is 15 but there are only 3 users in set M_2. The model randomly selects ((15/3)-1), 4 dummy queries from the query content engine and adds them along with original query to the request. Now there are 3 users in the group and 5 queries, so the probability of finding the right user and right query is 1/15. There are 3 * 5 = 15 combinations in the set. When the set is received by the attacker, the attacker finds that there are 3 persons and 5 queries and it is hard for the attacker to distinguish which person needs the location-based services. The success rate of the attacker is 1/15. The result reaches with the same privacy effect as the user requires.

2.5 Summary

Section 2.1 has explained the PLQ algorithm, which divides the area into regions to collect the information and send to LBSr, to improve services in the future. In section 2.2, we showed how geographic aware obfuscation is used to remove unreachable areas from the obfuscated area. The solution for maximum movement boundary attack is explained in section 2.3. Section 2.4 shows how adding of dummy questions increases the quality of services while providing the same user requested privacy level.

CHAPTER III

PROPOSED ARCHITECTURE

3.1 PRELIMINARY:

A user when requesting for a service, sends two privacy requirements, k-anonymity and minimum obfuscation area(Δs). k-Anonymity is the minimum number of users to be present in the request to protect the identity. The Minimum Obfuscated Area (Δs) is the minimum amount of area that has to be obfuscated before sending the query.

The TTP Anonymizer checks if these conditions are met before sending the query to the LBSr. Any request that cannot meet the requirements is discarded. If an attacker or LBSr, reduces the obfuscated area to less than Δs, then privacy is said to be compromised. The user, depending on the trust he has on a particular LBSr, varies these values. The higher the k-anonymity value and Δs, the less the trust on a particular LBSr.

In PLQ, the Map-hierarchy is offered by LBS Providers in advance. In the same way in our algorithm, the map is divided into regions by the LBS Provider in advance and this information is published to TTP Anonymizers. Hence, when there is a request from a user, the TTP Anonymizer knows from which region the request is issued based on the user's co-ordinates. The TTP anonymizer fulfils the user requirements and then sends the request to the LBS Provider.

3.1.1 PROBLEM SCENARIO

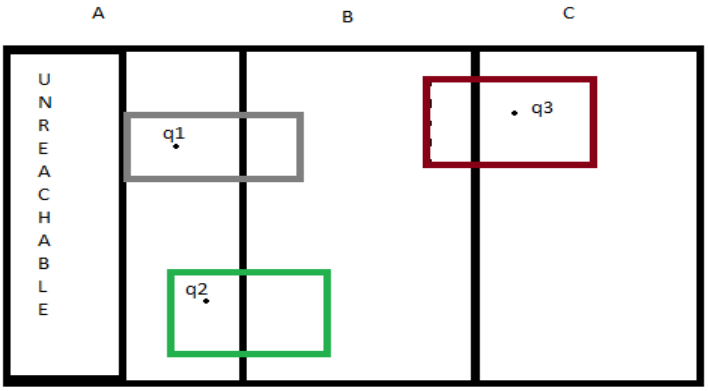

Figure 9: Problems of PLQ & B^{ob}- Tree

1) Consider the above figure, where q1, q2, q3 are 3 users and A, B, C are three regions that are published by the LBS Provider in advance as in PLQ. A has some unreachable area. If we use PLQ, when q1, q2 issue a request then the entire A region is added into their obfuscated area before sending the user query to LBSr. Similarly, the entire C region is added into obfuscated area for q3 when q3 requests a LBS. But if the adversary or un-trusted LBSr has map knowledge then he can eliminate the un-reachable areas from the obfuscated area. So, the area where user can be located is reduced and if it is less than the user requested obfuscated area (Δs) then the user is said to be under attack. On the other hand, let user q3 request a service with Δs much smaller than the entire area of region C ($\Delta s << Area(C)$). In this instance also, the obfuscated

area of q3 is expanded to the entire C region in PLQ. Because of this, the quality of LBS services has decreased since, the higher the obfuscated area, the lesser the quality of service.

2) If we use Geographic aware obfuscation [2] directly, then the obfuscated area lost due to unapproachable parts is added randomly and might be expanded as in figure 9. Hence, the LBSr gets incorrect information, because LBSr may view region B as the most frequently queried region because all the three cloaked areas cover a part of region B.

To deal with the above cited problems, we propose a new algorithm called Privacy Preserving Geometric Aware Location Query (PGALQ) which is explained below.

Algorithm:

Privacy Preserving Geometric Aware Location Query (PGALQ)

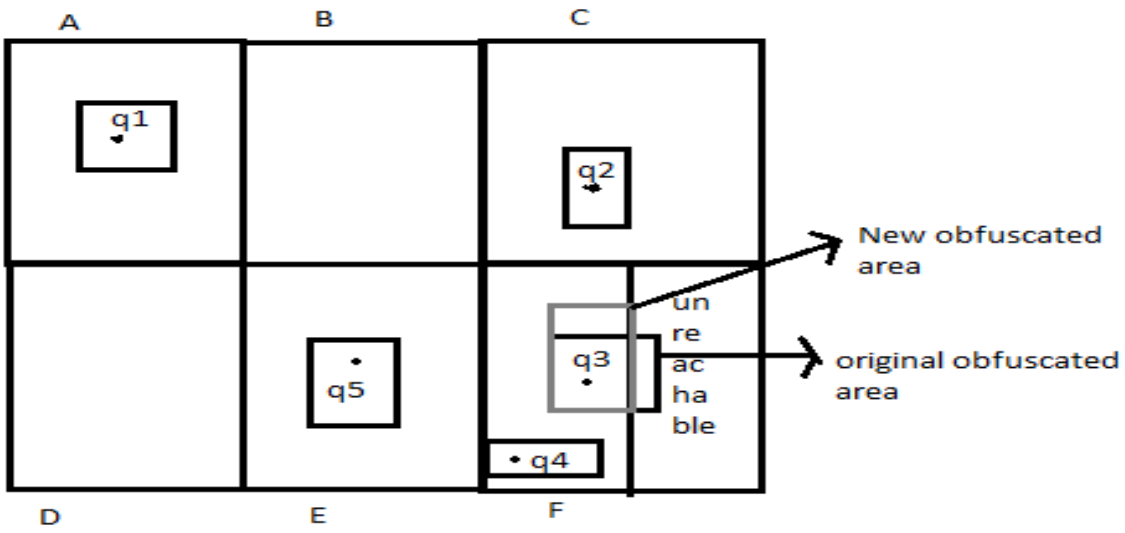

Figure 10: Diagrammatic Explanation of PGALQ

Consider there are 6 regions where region F has some unreachable area as shown in figure 10. Let 'Δs' be the minimum obfuscation area required by the user. It varies depending on the user's

belief on the particular LBSr. Queries q1 and q2 don't have any unreachable areas and the corresponding count of query q1 requested by the user in region A is incremented and the same applies to query q2 in region c. For q3, some area of Δs is lost because of unreachable parts present in region F. Hence we have to add a new reachable area to the obfuscated area in place of the unreachable part, in such a way that, the obfuscated region lies completely inside region F and increment the count of the corresponding query from that region, before sending it to the LBSr. Each time a user query is requested from a particular region its corresponding count is incremented and stored at the TTP. The aggregate information is sent to the LBSr at regular intervals to prevent the tracing of users when more than one query is requested by the same user requesting the same service.

3.1.2 System Architecture

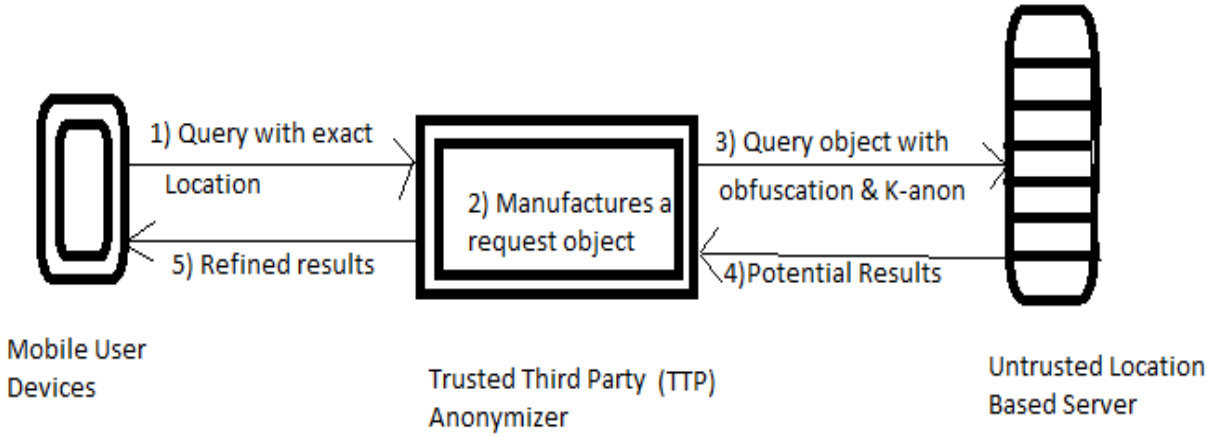

Figure 11: System Architecture

We adopt a 3-tier model comprised of mobile device (users), Trusted Third Party (TTP) Anonymizer and LBS provider. Since, the user does not trust LBSr it contacts TTP Anonymizer for user privacy.

The TTP Anonymizer does the following tasks:

1) Receives the user original position and then gives a unique number to the request object and then obfuscates randomly in the same region.

2) Checks the un-reachable points of the obfuscated area and substitutes new reachable points from the same region. The TTP Anonymizer has map information and it will check for reachable co-ordinates and unreachable co-ordinates. In our simulation, we created unreachable points in some regions randomly and stored in the program. We check every point that is obfuscated and checks if it belongs to the generated unreachable area. If not then that point is reachable or else it is unreachable. In this way it adds all the reachable points to the request and removes the unreachable points to obtain the same amount of obfuscation area. If we use B^{ob} tree then obfuscation would be done directly based on the information like reachable and unreachable stored on the nodes of the tree. But here we selects the random direction among the four directions and if the reachable area is not satisfied then we select the random direction in one of the other three directions followed by any one direction among the other two directions and then the last direction. Even after this, if the reachable area is not greater than user desired area then the same thing happens as explained above. After the entire region is visited and our algorithm is not able to form an obfuscated regions that satisfies user requirements then that is considered as the worst case scenario.

3) Add k-1 users to obtain k-anonymity with pseudonyms to the request object.

4) Add d-1 dummy questions with original question to the request.

5) Increment the query count with respect to the originating region and TTP sends the request object, which meets the user requirements. The object also has dummy questions in it.

6) Sends the aggregate query count to the LBSr at regular intervals to hide the original query. For example, a user has requested for the nearest gas station, according to our algorithm some dummy questions will be added to the request. If the query count is sent after every request, then LBSr can know the original query i.e. nearest gas station, in the previous request because its value is incremented.

7) TTP obtains response objects from LBSr which is a set of suitable responses and eliminates unnecessary responses because of obfuscation, k-anonymity and dummy questions. Then it sends the refined results to the user who requested for service.

Algorithm for PGALQ:

1) Mobile user requests for a service with the following parameters.

   (MU_i, POS_i, K_i, ΔS_i, Q_i).

   Where, MU_i = pseudonym of the user i.

   POS_i = Spatial co-ordinates of the user i.

   K_i = k-anonymity requirement of the user i.

   ΔS_i = minimum obfuscation area requirement of the user i.

   Q_i = It is the original query of the user i.

2) TTP acts as a mediator between the users and the Location based Service Provider (LBSr). It obtains the request from the user and does the following tasks.

-TTP -> req_i , assigns a new unique request id.

-TTP[region], TTP identifies the region from which the query is raised and also identifies the minimum area that has other 'k-1' users.

-TTP -> $req_i(O_i)$, generates the random obfuscated area covering those 'k-1' users.

-While true {

-TTP[$req_i (O_i)$] , TTP checks if the obfuscated area has any unreachable area and store the reachable area.

-TTP[$req_i(O_i) > \Delta S_i$] , TTP checks if the reachable area is greater than user desired area.

-If not then it adds the new reachable area.

-TTP[$req_i(OA_i) > \Delta S_i$], TTP checks if the new obfuscated area has reachable area greater than user desired area.

- If the above step is true, then breaks the loop.

}

-TTP[reg(count)] , TTP increments the corresponding query count in that region.

-TTP[req] -> LBsr. , sends the request object to LBSr, if it is not expired.

-TTP[agr_info] -> LBSr, sends the aggregate query information to LBSr after a specific time interval.

3) LBSr obtains the request object RO_i from the TTP Anonymizer and serves the request by sending the valid responses.

$RES_i(RO_i, R_1, R_2, R_3,.....)$

4) TTP Anonymizer obtains the response object from the LBSr, filters it and removes all the unwanted responses from it and then sends to the MU_i the information for the service it has requested.

3.2. Pseudonym and Multiple Query Confusion Algorithm (PMQCA)

3.2.1 Deficiencies of existing algorithms for Maximum Movement Boundary Attack

Algorithms like ICliqueCloak [3], URALP [6] which were developed to prevent maximum Movement Boundary Attack cannot be used in our above proposed PGALQ. This is because in both techniques, we have to find the Minimum Bounding Rectangle (MBR) for k-users based on their previous cloaked region and maximum movement boundary. In our proposed PGALQ, the obfuscated region must belong to a single region. Hence, the MBR also has to belong to a single region which depends on the previous obfuscated area and other external factors like speed, and it may not possible every time for the MBR to belong to a single region. We cannot obfuscate an area that belongs to 2 regions which are called miscounts, because it affects the aggregate count of the queries that has to be sent to LBSr for the improvement of LB services. Additionally, URALP and ICliqueCloak assume that a user is always assigned the same pseudonym id for all the queries issued. Because of this, the attacker can relate multiple query samples of a user and wait for the new request that has the same pseudonym id to perform a Maximum Movement Boundary (MMB) attack. This cannot be done using our scheme. This is explained in the next section and our algorithm deals with multiple sample query attacks as described below.

3.2.2 Pseudonym and Multiple Query Confusion Algorithm (PMQCA):

In figure 12, the MBR for q1, q2 belongs to both the regions as shown. Hence according to ICliqueCloak and URALP the obfuscated area has to be $MBR_{q1,q2}$ or the extension of $MBR_{q1,q2}$. That means, the obfuscated region has to belong to both the regions which is not supported by our PGALQ. Because of this, we cannot use MBR in our algorithm. Hence, we propose a new algorithm PMQCA to prevent multiple query attacks in our PGALQ algorithm.

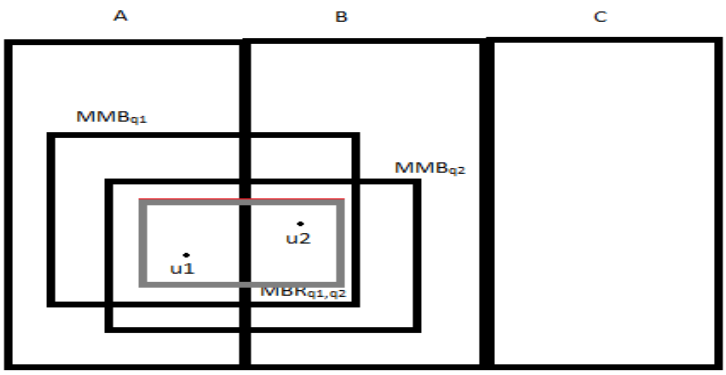

Figure 12: Problem with MBR in our Idea

In PMQCA, we propose a different approach that does not depend on MBR. Consider, a user u1 from region A who submits a query q1 with minimum cloaked area 'Δs' for a service. Now, the TTP gets the original location of the user and it generates a new record object with unique key to remember and identify this request uniquely. Then it finds the minimum possible obfuscated area that is greater than user requested area 'Δs', with 'k-1' other users to maintain k-anonymity and generates the random pseudonyms[8] for all the k users in the request and adds to the corresponding request object. Then as in [4], TTP randomly selects 'd-1' (d is based on user or

TTP) dummy queries and adds these queries along the original query to the request object and sends this request object to the LBSr. This is explained in the below diagram.

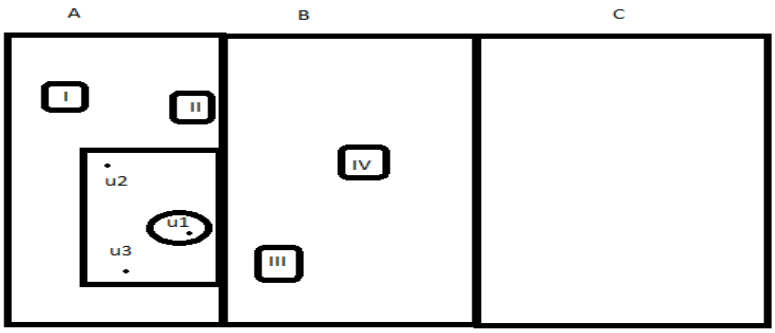

Figure 13: Solution with PMQCA

Let u1 be the original user who has issued a query q1 for service and 'd' be the number of dummy questions the user has requested. The TTP which acts as a mediator gets the request and does all the necessary processing before sending the request object to the service provider or LBSr. It adds u2, u3 who are other 2 users, to the request to obtain k-anonymity (k = 3). It also adds 3 dummy queries q2,q3,q4 to the request along with the original query q1, since d = 4 is considered. Let I, II, III, IV be the responses from the LBSr for queries q1, q2, q3, q4 respectively.

In this case, the success rate of the attacker is 1/12 with k = 3 and d = 4. In addition to this, the adversary does not know the original question and when the new request is issued from the user, he cannot link this request to perform a Shrink Region Attack [7]. An example for the Shrink Region Attack is explained using figure 14A with three users A, B, and C located at different positions. User A issues two different queries to the same client. The simple k-anonymity

approach used by A generates the k-anonymity set (A, B) for the first query and the anonymity set (A, C) for the second query. If the client correlates both queries, the client can infer that A originally issued the query. But in our approach as shown in figure 14B, we use a stateless pseudonym technique and also dummy questions which makes the attacker to view the 2 requests as independent requests. That means that when A issues a query for the first time, our approach assigns a random pseudonym for the user as well as other elements of k-anonymity and forms a k-anonymity set (a_1, a_2) where a_1 = A, and also generates question set (q_1, q_2) where q_1 is the original query. Now, the second time when A issues the query q_1 to the same client then our approach generates a set (a_2, a_3) where a_2 = A, and also generates question set (q_1, q_2). If the attacker tries to correlate both the queries, then because of the different pseudonyms he cannot come to a conclusion that there is a common user in both the queries. In addition to this, the un-trusted LBSr cannot know whether the original question is q_1 or not, but it sends the response both times.

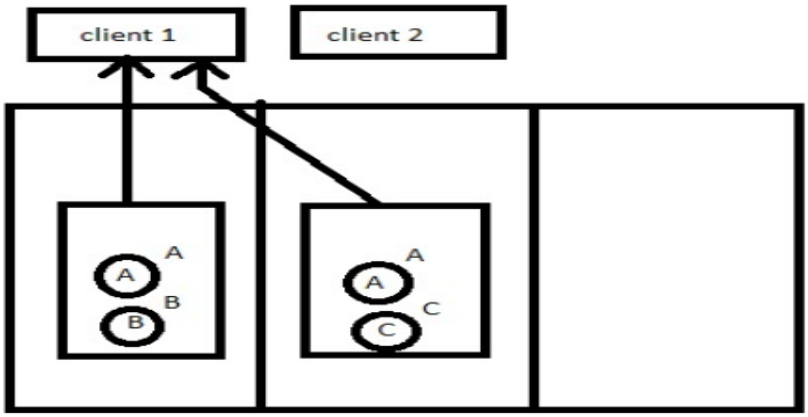

Figure 14A: Shrink Region Attack

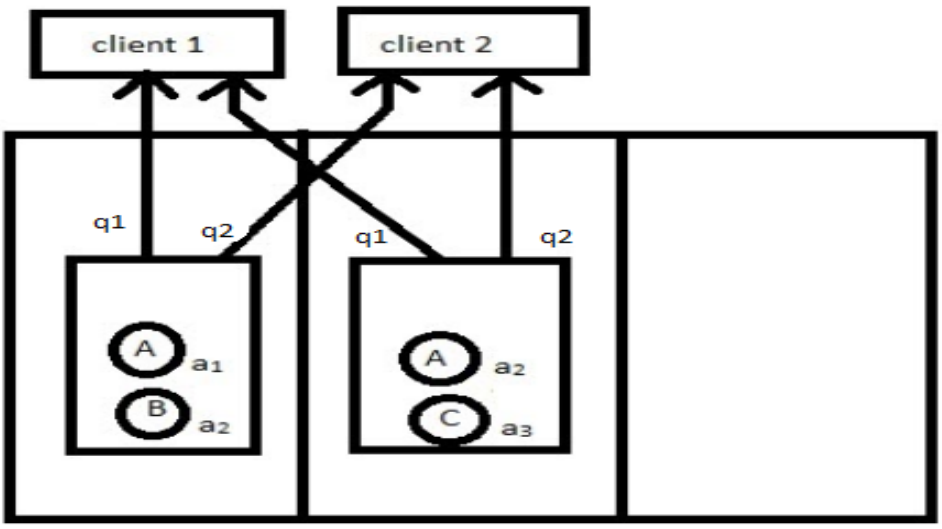

Figure 14B: Solution for Shrink Region Attack using PMQCA

To perform the MMB attack, the attacker has to track 2 or more queries issued from the same user. In our proposed approach, because of random pseudonyms generated for each query, the attacker cannot identify the requests from the same user. In addition, there will be dummy questions along with the original query associated with every request. Because of this, the untrusted LBSr cannot identify the original query. Even if the same user requests 2 different queries for the same service from the same service provider, he cannot conclude that the requests are from the same user because of k-anonymity and also because of his inability to perform a shrink region attack, as explained in figure 14B. The attacker cannot even identify that the 2 requests are for the same service because there are other dummy questions which also have equal probability. Hence, the attacker gets confused and cannot launch a MMB attack because he doesn't know in which direction the user is heading since one among the 4 responses (d = 4) is the answer to his original query.

Also consider an example that is taken in URALP[6], where an attacker Bob who wants to keep track of his teenage daughter Alice. One day on the way to a bar, Alice would like to find the nearest gas station to fuel her car. However, she does not want to disclose her location. So she reports a cloaked region to LBS instead of her exact coordinate. In the bar, she makes another query to see if any of her friends nearby are interested in joining her while hiding her coordinates because she also does not want her father to know that she is in a bar after school. By accessing the service provider's data somehow, Bob may obtain the cloaked location information of Alice's two LBS queries. From the first query, he can infer with high probability that Alice is currently driving in the city, so her speed cannot exceed 30mph. Then he determines the area where Alice can reach using maximum movement boundary (MMB) based on the time of the second query by using Alice's maximum speed. Therefore, knowing the area map and the MMB, Bob can limit the obfuscation area to certain locations (i.e., the bar) by removing all the unreachable regions based on the map and the MMB. Since we cannot use URALP's MBR solution for this problem, which is explained in section 3.2.1, We propose a new approach to deal with this problem.

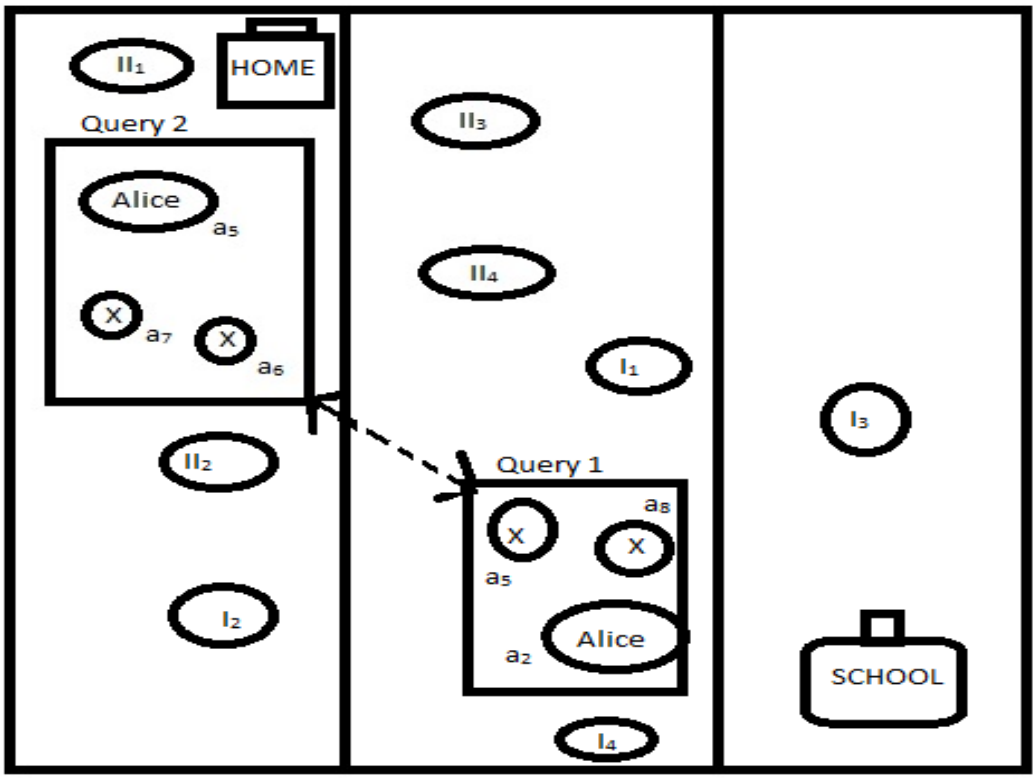

Figure 15: PMQCA solution for MMB attack**.**

Suppose Alice has requested for 2 services as above. Consider Figure 15, where a_2, a_5, a_8 are random pseudonyms of the first query and a_5, a_6, a_7 are random pseudonyms generated for the second query. In our approach, we are using the random pseudonym technique that can hide the identity of Alice and thereby tries to protect her from a MMB attack. When Alice first queried for the nearest gas station, then the request has 3 users including Alice and 4 questions including the question of 'nearest gas station'. In the bar, when Alice requests for 'nearby friends', its pseudonym is changed and it is added to the query with k = 3 and d = 4. When Bob tries to obtain location information of Alice, he cannot identify which queries belongs to Alice because

of change in pseudonyms for every query. Even if he guesses the first query belonged to Alice, then 1/12 is the original direction Alice is heading to and he may calculate MMB for that request and checks for Alice's next queries to perform a MMB attack. This might result in many intersections because there might be other users who are requesting for service since each request has many responses which are of equal probability.

Consider the above example of URALP to understand user query privacy. If Bob can manage to get spatial information then he can also get the queries requested by Alice. So, let's suppose Alice requested for 'nearest gas station' in her first query and 'nearest pub' in her second query. Bob can know the timing of these requests issued and he can link it with Alice school timings. That means even if k-anonymity is used, because of the timing the query is issued, Bob can consider that Alice might have issued the query and afterwards when the second query for the pub is issued then he can obtain the spatial information of the user and since it is in the direction of their home, Bob will come to a conclusion that it is Alice who issued the query. In addition to this, he also knows the query requested by Alice for the second time and he confirms that Alice has gone to a pub while returning from school. But if we use PMQCA, Bob cannot come to a conclusion about the place visited by Alice, as shown in figure 16. Suppose a query is requested by a user from Alice's School. This request has additional dummy queries along with the original query (nearest gas station). When Alice queries for second time requesting 'nearest pub', it is again mixed with dummy questions like 'nearest McDonalds' or 'nearest municipal office' etc. Hence, Bob cannot know what the original query is from the second request, this is shown in figure 16. Unlike the other approaches where there is only a single query in the request, PMQCA offers user query privacy also.

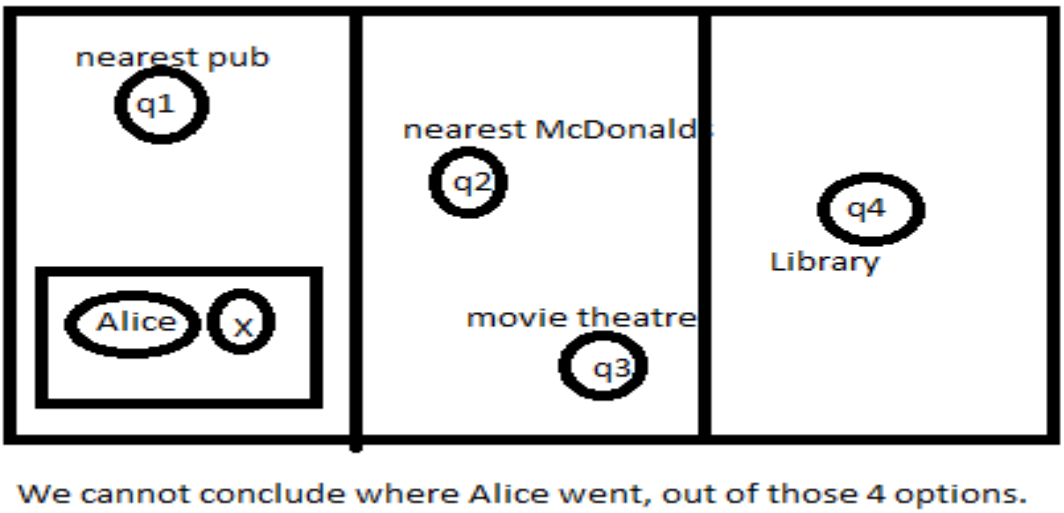

Figure 16: Example for User Query Privacy.

Since, there is only one query in the other approaches, the user who has requested for a gas station will stop at a gas station. Since, Bob knows the spatial information of the second query, he calculates the time it takes to reach that gas station by the first query user and timing of the second query. If both of them match then he concludes both the queries were requested by a single user and since both the queries spatial direction is towards the home, it is Alice who has requested for that service. In PMQCA, since multiple queries are there in a single request, when Alice requests for 'nearest gas station', then other dummy questions like 'football match timing', 'new movie time', 'nearest McDonalds' etc. are added to the request. The response for these queries will be in different directions. So, Bob knows that one among those responses is the original direction of the user. Next time, when Alice requests for 'nearest pub', it is again added with different types of queries. When Bob tries to match the timing of the second query and the time taken by the user of first query to travel near the spatial area of the second query, he will be confused because there may be other users who have requested for some new service which

matches the time of the first query. Hence, he does not know the original direction of the user and Bob cannot conclude that it is Alice who has requested the service both times or any other time. Hence, PMQCA also provides user's identity privacy.

After obtaining the response, the TTP refines the response and forwards the valid response to the user, using unique request id that is generated for each query in the beginning.

Algorithm for PGALQ with PMQCA:

Most of the algorithm remains the same except for the TTP Anonymizer where additional steps are added for implementing PMQCA. We look at TTP Anonymizer steps that differs with PGALQ.

i) Generate request object with unique key RO_i that can be identified uniquely within TTP anonymizer.

ii) Identify the region to which the user belongs to, using spatial co-ordinates POS_i.

iii) find 'k_i - 1' mobile users near MU_i and store the minimum possible boundary co-ordinates such that all 'k_i - 1' mobile users and MU_i are inside those boundary co-ordinates and calculate the area under these boundary points and store it as OA_i.

iv) generate random pseudonyms for these mobile users including MU_i, which are inside the boundary co-ordinates and add them randomly to RO_i.

v) Check if OA_i is greater than ΔS_i. If yes, go to step vii.

vi) Expand the boundary co-ordinates randomly in any direction and update OA_i. This is done until OA_i is greater than or equal to ΔS_i.

vii) Check if OA_i has any unreachable co-ordinate points and store the count of those points(UP_i).

viii) If $UP_i = 0$ then go to step ix, else expand the boundary co-ordinates randomly in any direction and update OA_i. This is done until $OA_i - UP_i > \Delta S_i$.

ix) Increment the count of corresponding query in that region.

x) Add 'd - 1' dummy questions and original question to the request object.

xi) Send the request to the LBSR. The request format is shown below.

RO_i (boundary co-ordinates, randomly added MU_i and other 'k_i - 1' users with randomly generated pseudonyms , Q_i and other 'd - 1' dummy queries added randomly).

xii) Send the aggregate information of queries from each region to LBSr, if the time interval to send the aggregate count of queries from TTP to LBSr is crossed.

CHAPTER-IV

SIMULATION RESULTS

4.1 Data Model

In all the experiments except query success rate (section 4.7), we assume the k-anonymity requirement of a user has been met. To implement PGALQ, we need a map hierarchy which is published by the LBSr. So, let us assume LBSr has divided the region between (35,-98) and (44,-89) into 9 regions. Then each region is again divided into 9 other regions. That means, there are a total of 54 regions where the leaf node region has the size of 1 unit of latitude and 1 unit of longitude, as shown in the figure.

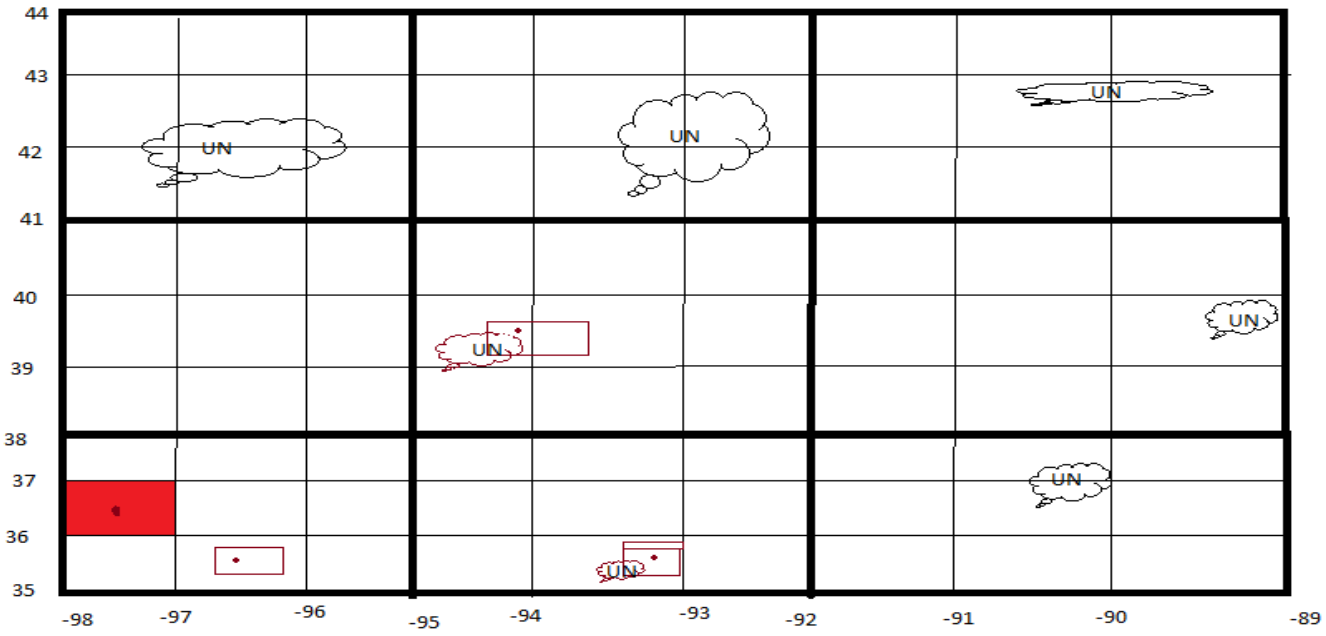

Figure 17: Data Model

Some unreachable area is generated randomly in some regions as shown in the figure and stored in the program. The experiment uses the Oldenburg urban traffic network [12] simulator for simulation of user data. In the above diagram, the clouds represent unreachable areas that are randomly generated and stored in the program with label 'UN'. The red rectangle represents the obfuscation area that is generated by PLQ as it obfuscates the area into a complete region. The black dot represents the original position of the user and the rectangle represents the obfuscation area.

Example of a point generated by the simulator.

Ex: 0 0 newpoint 38.758353 -96.853652.

Where first value is pseudonym, second value is the timestamp, third value says whether the user is a new user or an old user and next two values indicate the co-ordinates of the user.

The points generated by the simulator are slightly modified according to our requirements using a Generator program. Example of a point modified by the generator.

Ex: 38.758353, -96.853652, 20, 20, 2,q1

the first 2 values indicate the co-ordinate points and the next 2 values indicate the minimum length and breadth of the obfuscation area and the next value indicates the user option in the worst case where PGALQ cannot create an obfuscated area and the final value is the user query.

We implemented our algorithms in Java and also implemented PLQ and modified version of CliqueCloak for maximum movement boundary and used them for comparison with our algorithms.

4.2 PGALQ vs PLQ in Quality of Service

The higher the obfuscation area, the lower will be the quality of service. The region size depends on the LBSr and the LBSr tries to collect information in a larger areas like counties instead of streets because the LBSr will try to invest and improve the services in an optimal area rather than in a small area like streets. In PLQ, the obfuscation area depends on the area of the region.

Hence, if the region size is higher, then the obfuscation area is also higher, which means lower the quality of service. But in PGALQ, the obfuscation area will be equal to the user required area. Hence PGALQ offers high quality of service when compared to PLQ. To understand this, consider the below diagram.

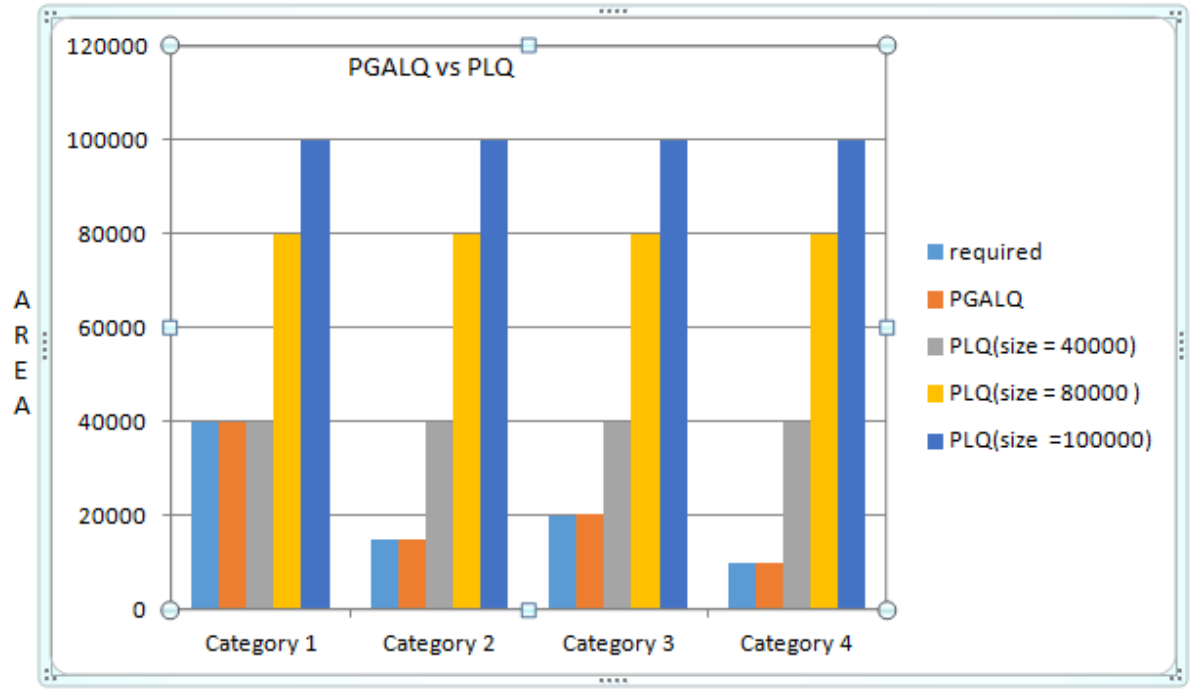

Figure 18 : PGALQ vs PLQ

In category 1, the user required area is 40000 and if the region size is also equal to 40000 then both PLQ and PGALQ obfuscates into same amount of area. In category 2, the required area is much smaller than the region size where region size is 40000 and required area is slightly less than 20000.In this case PLQ expands to the complete region and thus reduces the quality of service. But, PGALQ produces the obfuscated area that is equal to the required area. In all the categories, if the region size increases then the obfuscated area of PLQ also increases which implies in decrease in quality of service. However, PGALQ's obfuscated area is not affected by the region size as long as it is greater than the required area. Hence, PGALQ offers high quality of service when compared to PLQ.

4.3 PGALQ vs PLQ in worst case scenario

The worst case scenario is when the user required area is equal to the region size and the region has some unreachable area or the user required area is greater than the size of the region.

In the first case, where the region size including unreachable area is equal to the required area, PLQ obfuscates the area into the complete region and sends the request to LBSr. This means LBSr can map the obfuscated area and remove the unreachable area which decreases the obfuscated area to below 'Δs'. Thus user privacy is compromised.

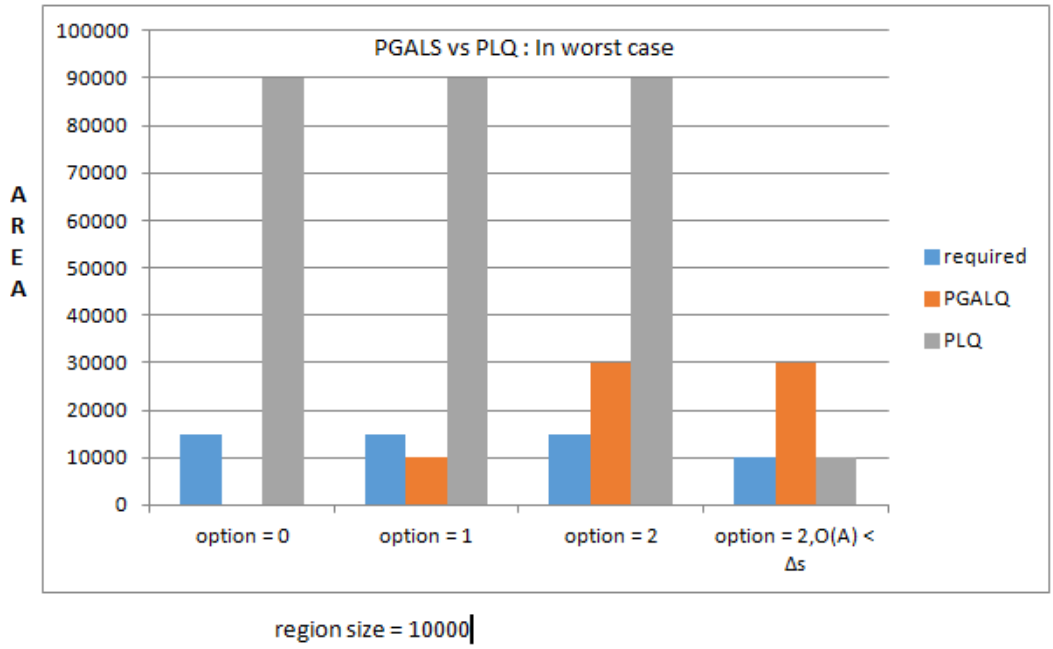

Figure 19 : PGALQ vs PLQ : In Worst Case

In PGALQ, you will deal with problems such as the obfuscation area going below Δs based on a user option value. This value input by the user is used to deal with worst–case scenarios. If user value for option = 0, the user does not want to compromise on privacy and also quality of service. Hence, PGALQ doesn't obfuscate the area and it simply discards the user request. The obfuscated area is therefore zero for PGALQ which is shown in the figure.

Therefore the success rate decreases but the algorithm will not violate user requirements. If option = 1, then PGALQ behaves like PLQ and it obfuscates the complete region and sends the request to LBSr. If option = 2, then PGALQ expands the obfuscated area into its adjacent region (sibling in map-hierarchy). Here, the quality of service is decreased but user privacy is maintained.

In figure 19, observe the 4^{th} category where Δs is greater than the size of the region. Here, both PLQ and PGALQ do not satisfy user requirements. Hence, PLQ expands the obfuscation area into its parent node i.e. area between (35,-98) and (38,-95) if the point belongs to one among those 9 regions, because it cannot form the obfuscated area in the current region. If PGALQ cannot form an obfuscated area in the present node, it expands the obfuscation into its siblings or adjacent nodes i.e. if the point belongs to any of the 3 regions between (35,-98) and (36,-95) then the obfuscated area will be area between (35,-98) and (36,-95), instead of the parent node. Hence the obfuscated area generated by PGALQ is lesser than PLQ even in this worst-case scenario.

4.4 PGALQ vs B^{0b} obfuscation in number of miscounts:

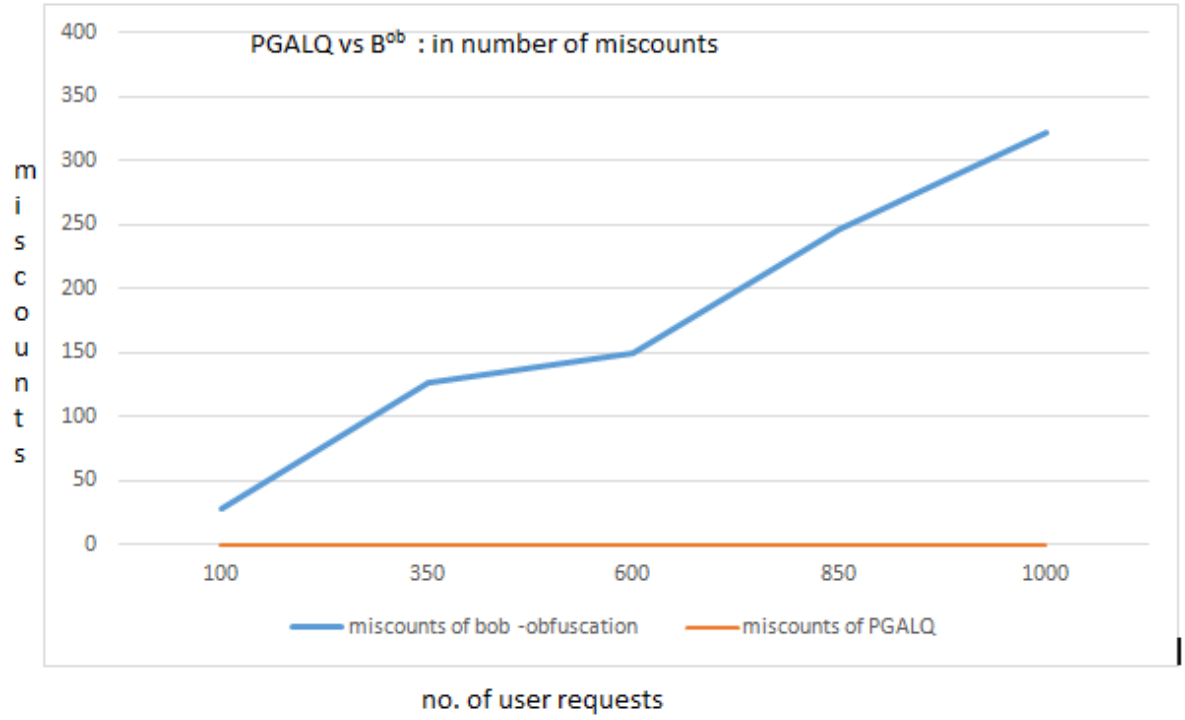

Figure 20 : PGALQ vs B^{ob} : in number of miscounts

X-axis: no. of user requests.

Y-axis: no. of miscounts.

Miscounts means the obfuscated area is expanded into 2 regions. Miscounts occur because user requests at the border of the regions is randomly expanded into other region using B^{ob} obfuscation. By using PGALQ, we expand the area in that particular region. Therefore miscounts don't occur.

In the above graph, as the number of requests increase, the number of miscounts also increase. This is because of the increase in number of requests that are in the border of the regions. However, the number of miscounts of PGALQ is always zero.

4.5 PGALQ vs PLQ in eliminating unreachable points

PLQ doesn't checks for any unreachable points in the obfuscated area. For example, consider a region 'r1' that has an area of 100 square units, of which 50 square units is an unreachable area. Let user 'u' from that region r1 has requested for LBS with minimum obfuscated area as 75 square units. PLQ expands the obfuscated area as a complete region 'r1' without considering the amount of unreachable area in that region. Hence user privacy is said to be compromised. PGALQ handles such problems depending on the user input. That means, PGALQ either discards or expands the obfuscated region into the whole area of its adjacent region . The below graph shows the amount of unreachable points included in PLQ.

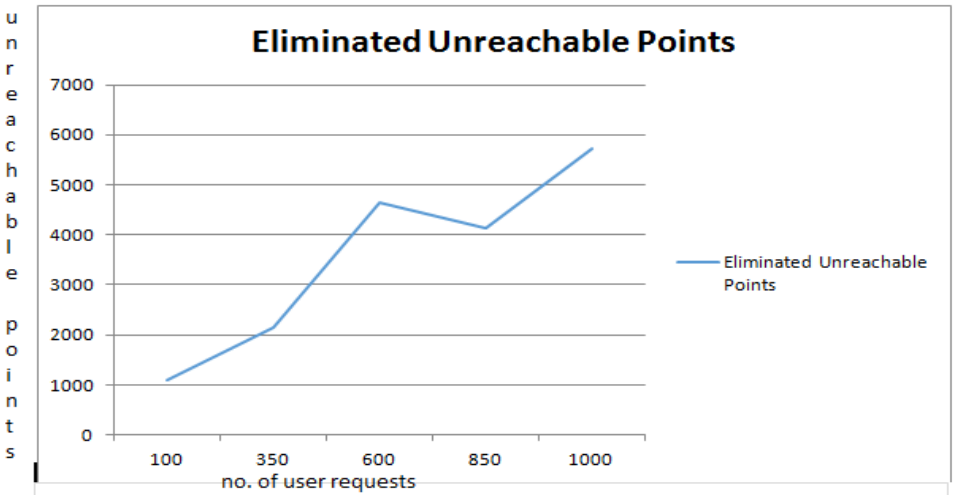

Figure 21 : Eliminating Unreachable Points

X-axis: no. of user requests

Y-axis: no. of unreachable points recovered by PGALQ

The amount of unreachable points included in the obfuscation area of the users increases with the number of user requests. That means, PGALQ substitutes at least this much amount of area or discards the requests that contain this much area depending on the user option.

4.6 User Query Privacy: PMQCA vs other approaches

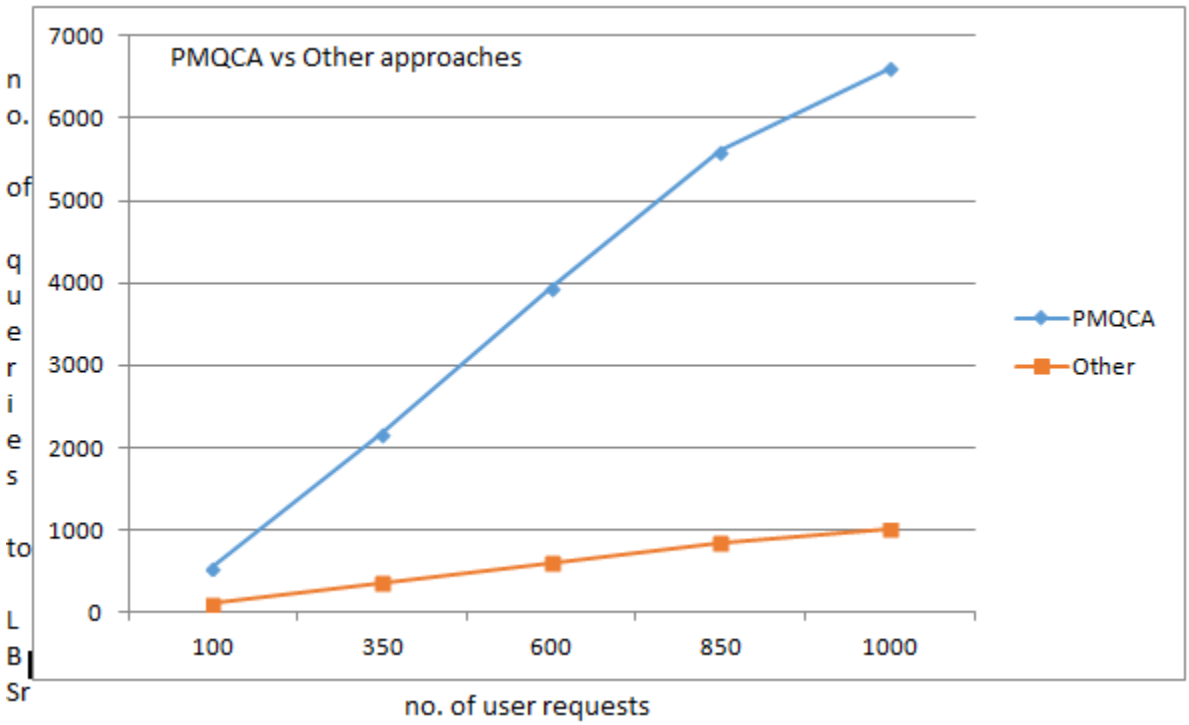

Figure 22 : User Query Privacy

X-axis: no. of user requests

Y-axis: total no. of queries that went to LBSr

PMQCA unlike other algorithms adds dummy questions to prevent user requests from a Shrink region attack. Adding of dummy questions to the request also protects user query privacy. For example, a user has requested for the nearest pub. Then LBSr can recognize that the user will go to the pub. But by adding the dummy questions, going to pub will be a possibility and there are other possibilities which can happen with equal probability.

The above graph shows how PMQCA offers query privacy when compared to other approaches like CliqueCloak. Here TTP selects some random number of dummy questions which it adds to the request along with the original query. We have selected the range of dummy questions between 3 and 10. From the above graph, we conclude that out of 7000 odd queries which went to LBSr only 1000 are original user queries. As the number of requests increases then the

average number of dummy questions for an original question also increases. Therefore, user query privacy also increases.

4.7 Query Success rate: PMQCA vs CliqueCloak

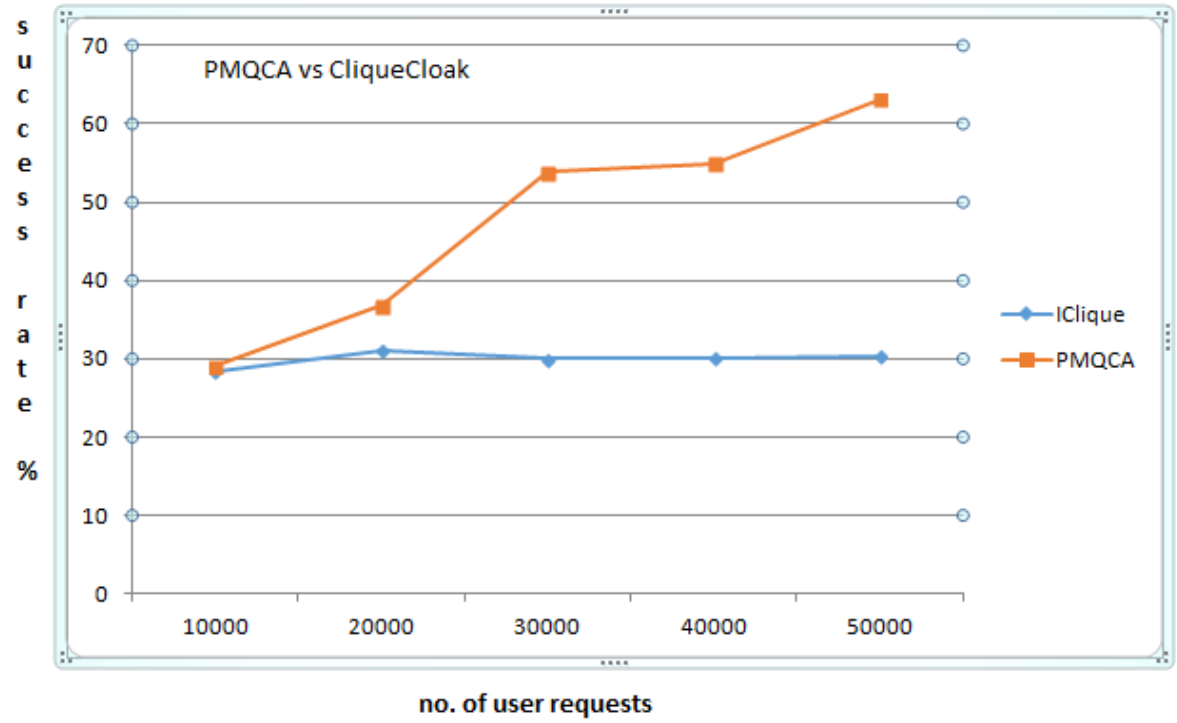

Figure 23 : Query Success Rate

X-axis: no. of user requests

Y-axis: query obfuscating success rate %.

The CliqueCloak algorithm checks if there are ‘k-1’ other users in the Maximum Movement Boundary of that user. Then it checks whether the original user is present in each and every ‘k-1‘ users maximum movement boundary. If that condition holds, then it finds the minimum bounding rectangle of those ‘k’ users in the request and expands the obfuscation area until obfus(Area) > Δs**.** If any of the above conditions are not met, then the request is discarded by TTP**.**

In PMQCA, random pseudonyms make the 2 requests of the same user as independent of each other. PMQCA calculates the random obfuscation area for that request and checks whether there are ‘k-1’ users present in that region. If not, then it expands the area to find the ‘k-1’ users in the obfuscated area. If the algorithm cannot find the obfuscated area greater than user desired area and with ‘k-1’ other users within some time limit, then TTP discards the request. Hence, the success rate of PMQCA will be much higher than the CliqueCloak.

In the graph, initially the user density is low, therefore the success rate of both the algorithms is low. The number of requests increases because user density is increased. Hence, PMQCA finds ‘k-1’ users or can expand the area until it finds ‘k-1’ users and sends the query to LBSr. Hence the success rate is increased. But for CliqueCloak, it has to find the ‘k-1’ users within the maximum movement boundary and this cannot be increased because it depends on external factors like speed. Otherwise the user request is discarded.

CHAPTER-V

CONCLUSIONS

The basic idea of this thesis comes from PLQ, user privacy is protected while collecting aggregate information of user queries to the LBSr to improve services. We proposed a new algorithm PGALQ by adapting geometric aware obfuscation, which performs better than PLQ in providing Quality of Service. Our algorithm PGALQ also protects user privacy by eliminating unreachable parts and discards the request or expands the obfuscated area depending on the user input. We have also proposed PMQCA, an extension of PGALQ to deal with the shrink region attack, Maximum movement boundary attack and query privacy attack. Simulation results showed that the proposed architecture performs better than algorithms in the literature.

If the region size is small, then both PGALQ and PLQ might generate the obfuscated area much larger than the user desired area. Hence, the way LBSr should divide the region size has to be studied more. The proposed algorithm tries to protect user privacy for users who have requested for the service on the go. It doesn't provide any privacy if the attacker knows the daily trajectory of the user. PMQCA protects the users from Maximum Movement Boundary attack without considering the recent MultiTarget Tracking methods because of advancements in technology. The combination of PMQCA and CliqueCloak may be a good solution for MultiTarget Tracking which has to be studied. We have to check whether other algorithms like position dummies, mix zones can be adapted into PGALQ.

REFERENCES

[1] Liu Yubao, Chen Xiuwei, Li Zhan, *et al*. "An efficient method for privacy preserving location queries". *Front Computer Science*, 2012.

[2] To, Q.C., Dang, T.K., Küng, J.: "B^{ob}-Tree: An Efficient B^{+}-Tree Based Index Structure for Geographic-Aware Obfuscation". In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.), *Proceedings Third Asian Conference on Intelligent Information and Database Systems*, (ACIIDS), 2011.

[3] X. Pan, J. Xu and X. Meng, "Protecting Location Privacy Against Location-Dependent Attack in Mobile Services", *Proc. 17th ACM Conf. Information and Knowledge Management*, ACM, 2008.

[4] Hu Wenling, Wang Yongli, Zhang Gongxuan, "A Privacy Protection Method for Continuous Query Based On Location Services", *Wuhan University Journal of Natural Sciences*, Springer-Verlag Berlin Heidelberg 2013.

[5] G. Ghinita, M.L. Damiani and C. Silvestri, "Preventing Velocity-Based Linkage Attacks in Location-Aware Applications", *Proc. 17th ACM SIGSPATIAL Int'l Conf Advances in Geographic Information Systems,2009*.

[6] Nha Nguyen, Seungchul Han, and Minho Shin,"URALP: Unreachable Region Aware Location Privacy against Maximum Movement Boundary Attack", *International Journal of Distributed Sensor Networks*, Vol. 2015, no.2 January 2015.

[7] M. Wernke, P. Skvortsov, F. Dürr and K. Rothermel, "A classification of location privacy attacks and approaches", *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 163-1175, 2014

[8] J. Petit, F. Schaub, M. Feiri and F. Kargl, "Pseudonym schemes in vehicular networks: A survey", *IEEE Commun. Surveys Tutorials.*, vol. 17, no. 1, pp. 228-255, 2015

[9] Matt Duckham , Lars Kulik, "A formal model of obfuscation and negotiation for location privacy", *Proceedings of the Third international conference on Pervasive Computing*, May 08-13, 2005.

[10] Gutscher A (2006) "Coordinate transformation—a solution for the privacy problem of location based services?", *Proceedings of the 20th international conference on parallel and distributed processing* (IPDPS '06), 2006.

[11] M. Damiani, E. Bertino, and C. Silvestri. Protecting Location Privacy through Semantics-aware Obfuscation Techniques. *Proc. of the International Fedreation for Information Processing* (IFIPTM), pages 231–245, 2008.

[12] Thomas Brinkhoff: Network-based Generator of Moving Objects, https://iapg.jadehs.de/personen/brinkhoff/generator, last accessed on 12/05/2016.

VITA

PRADEEP BEJGAOM

Candidate for the Degree of

Master of Science

Thesis: PRIVACY PRESERVATION AND SERVICE PROVISION IN GEO-SPATIAL-TEMPORAL SYSTEMS

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December, 2016.

Completed the requirements for the Bachelor of Technology in Computer Science and Engineering at Jawaharlal Nehru Technological University, Hyderabad, Andhra Pradesh, India in 2014.