International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012

LPM: A DISTRIBUTED ARCHITECTURE AND ALOGORITHMS FOR LOCATION PRIVACY IN LBS

Muhamed Ilyas[1,*], Dr. R. Vijayakumar[2]

[1]Research Scholar, School of Computer Science, Mahatma Gandhi University
Kottayam, Kerala, India
Muhamed.ilyas@gmail.com
[2]School of Computer Science, Mahatma Gandhi University
Kottayam, Kerala, India
Kiran2k@bsnl.in

ABSTRACT

Recent advances in mobile communication and development of sophisticated equipments lead to the wide spread use of Location Based Services (LBS). A major concern for large-scale deployment of LBSs is the potential abuse of their client location data, which may imply sensitive personal information. Protecting location information of the mobile user is challenging because a location itself may reveal user identity. Several schemes have been proposed for location cloaking. In our paper, we propose a generic Enhanced Location Privacy Model (LPM), which describes the concept, the architecture, algorithms and the functionalities for location privacy in LBS. As per the architecture, the system ensures location privacy, without trusting anybody including the peers or LBS servers. The system is fully distributed and evaluation shows its efficiency and high level of privacy with QoS.

KEYWORDS

Location privacy, Location Based Services, Location Cloaking, Distributed Query Processing

1. INTRODUCTION

The last decade showed an accelerated development of mobile and Internet technologies. Internet technology with globally connected mobile networks introduces new business models and the development of service architecture. Location-Based Services (LBS) are such an example. Location based services (LBS) are Internet services that provide information or enable communication based on the location of users and/or resources at specific times. Service providers envision offering many new services based on a user's location as well as augmenting many existing services with location information [3]. At the same time, LBSs poses a new threat, i.e., privacy preservation. For example, someone wants to have dinner and is searching for a restaurant using the Internet. In order to get more accurate and useful research results, more terms such as the mobile user's location, the type of food, etc. should be included in his search criteria. Unfortunately, if the queries are not securely managed, it could be possible for a third party to retrieve the mobile user's personal sensitive information such as his location information, his habit, etc. In this case, even if an individual does not directly release personal information to the service provider, this provider may become aware of the sensitive information if it has to provide a service to such an individual [4].

Research in the field of privacy preservation in pervasive computing has mainly concentrated on techniques for anonymous communication [1], access control and obfuscation [6, 7], dummy requests [5], or on a combination of such techniques. Many of these techniques are based on a central server called Location anonymizer (LA). In this case, the mobile user has to submit his/her location identifier to the LA, and LA cloaks the location using different models

DOI : 10.5121/ijnsa.2012.4210

developed, like K-anonymity, before submitting the query to the LBS. Location Cloaking with a centralized architecture must trust the central third party server with their identities, locations and queries. However, there are a number of disadvantages for centralized approaches, such as a single point of failure, bottlenecks due to communication overhead, and privacy threats as these systems store all information in a single place. To overcome these problems, several decentralized approaches have bee proposed. [22].

We propose a distributed approach to protect user privacy in LBS that does not need a centralized server for location cloaking and does not trust any one including participating peers. Our approach is similar to the work proposed by [5], but with cluster based peer selection algorithm and an enhanced distributed peer cloaking method. Our approach uses the capabilities of current mobile systems to form ad-hoc Wireless Personal Area Networks (WPANs) using technologies like Bluetooth. As per the system, a user who needs a location services, called query initiator, initially forms a peer group of n individuals based on a cluster algorithm. Then it randomly selects a peer, called query requestor, to forward the query, on behalf of the query initiator, to the LBS server. A major challenge of this approach is the selection of the query requestor with uniform probability. It ensures that even if the LSP has access to the information that currently n devices form an ad-hoc network, the LSP is only able to identify the query initiator with a probability of $1/n$ [5].

In our approach, the user and the peers do not reveal their exact location to each other. Instead the actual positions are obfuscated with an imprecise location like circle. For maximal privacy protection this approach combines obfuscation with K-anonymity [5]. If a user requires a location service, our algorithm computes a minimum bounding circle (called Global Cloaking Area, GCA), that enclose obfuscated locations of all his peers. The GCA contains the obfuscated location of the user and the obfuscated locations of all other K-1 peers.

In summary, our contributions in this paper are as follows:

We propose a heuristic algorithm to compute the obfuscated location, called Self Cloaked Area (SCA), of the user and all its participating peers. Self Cloaking is done individually by the query initiator and all participating peers.

We develop a Greedy algorithm for generating a user's K-anonymous obfuscated location from available n SCAs. In each iteration the algorithm checks the K-anonymity and continues until K-anonymity level is met. Unlike in [5], where the selection of peers to meet the K-anonymity is done by the query initiator, our work distributes this process among peers. Each peer calculates, whether it's Self Cloaked Area is within the GCA and is eligible to participate in the obfuscation process to meet the K-anonymity level.

We present a near-uniform random selection algorithm to select a query requestor without revealing their identities. In paper [5] they presented a decentralized approach to protect user privacy during the access of LBSs using wireless ad-hoc networks. Users do not need to trust any involved party, including their peers, the LSP or the infrastructure. We extend this work by further decentralizing location obfuscation among peers with less computational overheads, and also introducing simple greedy algorithm for a near-uniform random selection for any type of peer distribution.

2. RELATED WORKS

Location anonymity and privacy awareness in location-based services has been extensively studied as a solution to protect user privacy in recent literatures. The objective is to allow the mobile user to request services without compromising his/her privacy, especially location privacy [8]. Several privacy protection techniques have already been proposed. These techniques are broadly classified into TTP-based methods and (ii) TTP-free methods. Based on

the underlying methodologies, these techniques can further be divided into three categories: pseudonym, Cloaking, Transformation.

2.1 Simple and TTP Schemes

In the simplest form of communication between an LBS user (U) and an LBS provider (P), the former sends a simple query (Q) containing an ID, his location (L) and a request for information (I) that he wants to retrieve from P as shown in figure 1. Thus, a simple query sent from U to P can be Q = {ID, L, I} = {ID, x_U, y_U, "Where is the closest bus station?"} By sending their current locations to P, LBS users assume that P manages their data honestly and refrains from any misuse. However, LBS providers cannot always be trusted and more complex communication schemes are needed. [23].

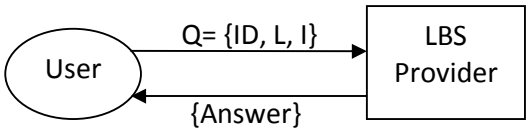

Figure 1. Simple communication scheme with an LBS user and an LBS provider

Most of the initial solutions for location privacy were based on Trusted Third Parties (TTP) as shown in Figure 2. In the simple scheme described above, users send their location information and queries directly to the LBS provider. In TTP scheme, instead of sending the query directly to LBS server, it is submitted to a TTP, where the location and identification of the user is obfuscated using either transformation or pseudonym. TTP act as an intermediate entity between user and LBS server. So, LBS providers are no longer aware of the real locations and identities of the users. The problem is that the user has to trust the third party intermediate entity instead of LBS server as in the case of simple LBS service, and whatever location privacy LBS users can get depends on the honest behaviour of the TTP.

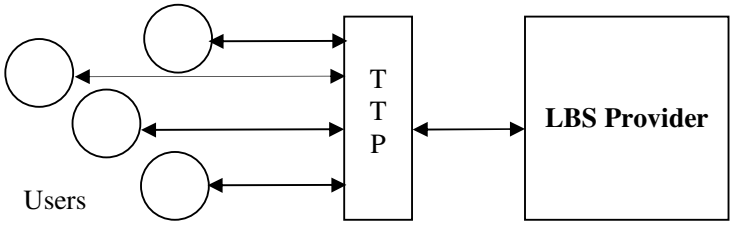

Figure 2. A TTP based scheme

TTP-based schemes are very common because they are easy to understand/develop, and because, in general, they offer a reasonable trade-off between efficiency, accuracy and privacy. Moreover, some of the ideas used in these schemes arose in more mature fields like e-commerce [23]. A TTP can act as a Pseudonymiser or as an Anonymiser.

Pseudonymiser is the simplest form of a TTP scheme. It receives queries from users and, prior to forwarding to them to the LBS server; it replaces the real ID of the user with pseudonym. In this way the real user is hidden from the LBS server. However, the real ID is kept with the Pseudonymiser in order to forward the answer from the LBS server to the user. The problem is that it is vulnerable to attack as both real IDs and their corresponding

pseudonyms are stored at the same place. Moreover, users must completely trust Pseudonymiser, because the latter see all the location information of the former.

Anonymiser is the most sophisticated form of TTP scheme. In this scheme, the intermediate entity which act as a TTP, hide the real location of the user by using the k-anonymity property. k-Anonymity is an interesting approach to hide a user location by cloaking (spatial cloaking) the location of another k-1 users. Spatial cloaking with k-anonymity was first suggested by Samarati and Sweeney [16, 17, and 18]. The location of a user is k-anonymous if it is indistinguishable from the location of another $k - 1$ users. So, the fundamental idea behind k-anonymisers is to replace the real location of the user by cloaking areas (spatial cloaking) in which at least k users are located. Spatial cloaking is being applied in both TTP based scheme as well as TTP free schemes. While the former employs a centralized method, the latter suggests different distributed methods for maximum location privacy.

2.2 TTP Based Spatial Cloaking

Spatial Cloaking is the most widely used privacy preserving technique for users accessing LBS. The main idea behind cloaking approaches is to blur a user's exact location in a larger cloaked region and to make him/her indistinguishable among the set of other (real or dummy) users located in the cloaked region

Many existing approaches in spatial cloaking are based on a centralized architecture. These approaches rely on the existence of a trusted Intermediary server called location anonymizer which protects a user's private location and identity information from an untrusted location server (e.g., Mokbel et al., 2006; Gruteser and Grunwald, 2003; Gedik and Liu, 2005a, 2005b; Du et al., 2007) [9 -14]. The main idea in centralized cloaking is to put an anonymiser between the users and the location server to prevent the server from learning users' precise location information and identities. Every location-based query is first sent to the anonymizer, which transforms the user's exact location to a cloaked area (i.e., rectangle or circle) and forwards the query to the LBS server for that cloaked area. While different cloaking algorithms are proposed for cloaking a user's location, the common objective is to blur a user's location in an area of size at least A_{min} and/or among a set of at least $k - 1$ other users. Depending on the approach, these parameters can be specified by each user independently, or are chosen as system parameters. During the second phase, the privacy-aware location server, which is modified to process a cloaked region query, generates a candidate list which is guaranteed to include the nearest neighbor of any point inside the cloaked region. This list is then transferred to the client side for further refinement to obtain the final result set [15]. The blurred spatial area can be based either on the k-anonymity concept [Samarati 2001; Sweeney 2002a, 2002b] [16, 17,18] (i.e., the area should contain at least k users) or on a graph model that represents a road network [Duckham and Kulik 2005]. [19]

2.2 TTP Free Spatial Cloaking

Centralized approaches discussed above, however the centralized approach has several disadvantages. This approach requires an anonymiser, as sophisticated as the location server itself, to act as a proxy between users and the server per query. There are chances for single point of failure/attack and bottleneck due to communication overhead. Another important drawback is that, in many scenarios cloaking users' location information in a larger region or among $k - 1$ other user does not protect user's location information. This is due to the fact that based on user distributions in the space and the value of k (or similarly size of the cloaked region), precise user location can be derived using several techniques. To overcome these limitations, decentralized approaches have been proposed that construct cloaked region. The approaches proposed by Chow et al. (2006) [20] and Ghinita et al. (2007b, 2007c) [20, 21] assume users communicate with each other to collaboratively form a cloaked region. Ghinita et al. (2007b) propose a hierarchical overlay network resembling a distributed B+ tree for constructing the cloaked region that overcomes the above drawback. However, it suffers from

very slow response time. Ghinita et al. (2007c)[21] propose methods which provide stronger privacy than Chow et al. (2006) for various distributions and do not suffer from slow response time of Ghinita et al. (2007b). The authors propose a distributed method to find a random set of k adjacent users based on their 1-D Hilbert ordering. Finally, Duckham and Kulik (2005) [19] propose a graph model to represent possible user's locations and denote the cloaked region by a set of vertices in the graph. The client progressively gives more information about her precise location until the query result set reaches her desired accuracy.

Tanzima Hashem, Lars Kulik [5] have developed a decentralized approach to protect user privacy during the access of LBSs using wireless ad-hoc networks. Users do not need to trust any involved party, including their peers, the LSP or the infrastructure provider. It exploits the wireless advantage that all users in communication range can overhear a message to anonymize the communication among users.

Wireless Ad-hoc networks are adaptive and self-organizing, and a consequence securing such networks is non-trivial. Several efficient protocols have been developed for securing medium access, routing, resource management, quality of service and security in mobile ad-hoc networks. In particular, protocols like AODV, R-AODV (Reliant Ad-hoc On-demand Distance Vector Routing) and Trust Based Scheme for QoS Assurance in Mobile Ad-Hoc Networks [25] propose security enhancements by ensuring that data does not go through malicious nodes that have been known to misbehave. However, in our paper, we are not going into the details of ad-hoc network security.

Our work is similar to [5] but with less communication overhead and with a distributed cloaking method. In [5], every user has to maintain the list of their peers within the communication range all the time and it is maintained even if the user is not intended to submit any request to the LBS server. This makes heavy processing and communication overhead for all users in a mobile environment, and peers may change dynamically. Secondly, calculating the cloaked area and selecting K-1 peers within the cloaked area are done at the query requestor, which incurs processing overhead at the query requestor. In our method, selecting K-1 peers within the cloaked area is done dynamically, only when the user wants to get some service from the LBS server. Selection of K-1 peers within the cloaked area is done by the peers, thus eliminating the process overhead at the query requestor. The second Simulation results show that it has less communication overhead and high quality of service.

2.3 Query processing

Several efficient algorithms has been developed for finding the nearest POI with respect a rectangular or a circular area. In new Casper [10], Mokbel et al. have developed an algorithm that returns a range of POIs including the nearest POIs for every point of a rectangle. In [12], algorithms have been proposed to evaluate m-nearest POIs for every point of a circle. All of these algorithms may need to return a large set of answers and thus incur high processing and communication overheads. In our previous paper [24], this issue has been addressed by using a decentralized architecture for processing POI and range queries.

3. SYSTEM ARCHITECTURE AND LOCATION CLOAKING

We present a decentralized system that employs the power of ad-hoc networking for obfuscating the user location from third party LBS servers. User and each participating peer, cloaks their location as a circle, where the user location may be anywhere within the circle, and thus the location cannot be identified by an adversary. The user, who wants to access the LBS service, first determines K-1 number of participating peers for cloaking the location. After that it determines the size of the circle, which contains all K-1 peers that participate in the cloaking process, (Called Global Cloaked Area) in terms of its radius. The cloaking process proceeds in

two stages. Initially, the query initiator determines the radius of GCA and value of K. and it calculates its Self Cloaked Area (SCA). Then it sends a broadcast message to all its peers, which are normally one hope away from the user. The message contains three parameters; the pseudo IP address of the query initiator, the parameters of its SC, and the parameters of the initial GCA. On receiving the message, each peer calculates its SCA. Then it performs a spatial '*within*' operation to identify that its SCA lies completely within the boundary of GCA. The result is true if the SCA is fully within GCA, else the result is false. The peer returns its SCA along with Boolean result of the spatial operation. After receiving the results from its peers, the query initiator checks for the K-anonymity. If the desired anonymity is met, it proceeds to the next step, otherwise continues the process with hope distances > 1 until the K-anonymity level is met. The GCA will be a minimum radii circle which encloses all K-1 SCAs. Then our Random Selection Algorithm selects a query requestor to forward the query along with GCA to LBS Server.

In our proposed method, the peers are responsible for identifying themselves, whether to participate in the cloaking process or not. This is a distributed approach, thus eliminating the overhead at the query initiator for calculating the Globally Cloaked Area (GCA) with K-1 peers.

3.1 Generating SCA

In our approach, we generate a circular cloaked area which contains the peer's real location anywhere in the circle. We have developed a heuristic algorithm to generate the cloaked circle. Let (x, y) be the real location of the mobile user. (The location might be received from GPS or any other means). In order to obtain maximum anonymity, we cloak the point (x, y) with a surrounding circle. But if we generate such a circle, adversary can easily identify the location of the user, as it may be centre of the circle. So we translate the real location of the user to a point (x_0, y_0), and generate a pseudo circle with centre (x_0, y_0) as shown in figure 3. The radius of the pseudo circle is chosen in such a way that the real location (x, y) of the mobile user must be anywhere within the circle. Let R be the radius of the Self cloaked circle, decided by each user. In order to find a random point (x_0, y_0), we randomly choose a distance value r, where r <= R, and an angle θ, where 0<= θ <= 2Π.

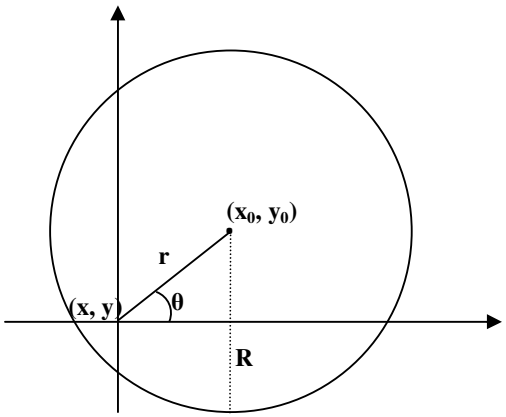

Figure 3. Self Cloaking

The real location (x, y) of the user is transformed into another location by using simple polar to rectangular coordinate conversion equation $x = r \cos \theta$, $y = r \sin \theta$

Then, the transformed location (x_0, y_0) is obtained by $x_0 = x + r \text{ Cos } \theta$

$y_0 = y + r \text{ Sin } \theta$, where r <= R

The cloaked area, with radius R, then can be calculated. This ensures that the original location of the mobile user is within or on the boundary of the newly created obfuscated circle with centre (x_0, y_0). The algorithm for generating a self cloaked area by a user is given below.

Algorithm 1: ComputeSCA

Input:

x_0 , y_0 : Centre of the GCA circle
RGCA : Radius of GCA
Ref : x_i , y_i , R
Ref : within (Boolean)

Output:

x_i : x-coordinate of the obfuscated user position
y_i : y-coordinate of the obfuscated user position
R : Radius of SCA (Obfuscated circle)

1. x ← x-coordinate of the user position from GPS
2. y ← y-coordinate of the user position from GPS
3. Let r (Random value where $0< r \leq R$
4. θ (Random value where $0 \leq \theta \leq 2\Pi$
5. R (Required radius of the obfuscated circle (SCA)
6. $x_i = x + r \cos \theta$
7. $y_i = y + r \sin \theta$
8. within ← Return 'True' if the spatial '*within*' operation of SCA with GCA (RGCA as radius), else return 'False'
9. return x_i, y_i, R, within as reference
10. Return true if spatial within operation is true

3.2 Generating GCA

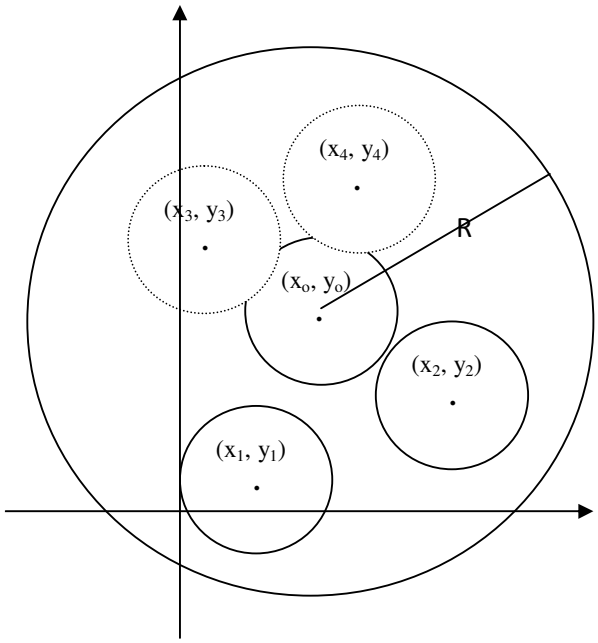

Figure 4. Globally Cloaked Area

Let K_l and K_h be lowest and highest anonymity level of the query initiator j. Also let R_l and R_h be the lowest and highest radius of the circular area which the query initiator wants to obfuscate, which we call the Globally Cloaked Area (GCA) as shown in figure 4. R is selected in such a way that it balances the K-anonymity and an optimal area which includes all other K-1 self cloaked peers. At the first step, the Query initiator sends a message to all its 1-hop peers requesting its pseudonym, the obfuscated origin (x_{pi}, y_{Pi}) of the Self Cloaked Area (SCA) and the radius of SCA; if SCA of the peer Pi is within the Globally Cloaked Area (GCA).

Initially the message is sent down to all 1-hop peers with a value R. If the query initiator fails to find sufficient number of SCAs within the limits of GCA, query initiator either decrement the value of K or it increments the value of R. This process continues until the value of K >= K_l and the value of R reaches R_h.

3.3 The Greedy Algorithm

We present an algorithm to compute the Globally Cloaked Area of the query initiator j. This is Greedy algorithm which executes until the desired level of anonymity is obtained. Assume that K is the desired anonymity level of j and R_l and R_h be the lowest and highest radius of the Globally Cloaked Area. i.e., the area of Globally Cloaked Area is $\pi R_l^2 \leq GCA \leq \pi R_h^2$. To compute the GCA, we have to find the smallest circle r that encloses a K-subset (including j's SCA) from the n SCAs. The SCA of a user i is described by its obfuscated centre (x_i, y_i) and the radius r_i. If the initial GCA with radius R_l and query initiators obfuscated centre (x_0,y_0), is not able to contain all K-1 SCAs then the value of R (Radius of GCA) is incremented in each step until R_l reaches the value R_h. We have developed a greedy-based algorithm (GBA) with a time complexity of O(h), where h is the hop count. Initially the message is broadcasted to all 1-hop peers. All peers receiving this message, computes its own SCA, perform a spatial 'within' operation with GCA (The 'within' predicate in spatial geometry returns t (TRUE) if the first geometry is completely inside the second geometry). If the result of the 'Within' operation is true, then function returns the obfuscated centre of the SCA, and the radius of the SCA. The numbers of SCAs are counted after each hop iteration, if the number of SCAs are below the K-anonymity level, then we increment the hop and continue the process until the system desired parameter h_{max} is reached.

Algorithm 2: Compute GCA

Input:

- h_{max} : The maximum hop count
- K : The anonymity level
- R_l : The minimum radius of the GCA
- R_h : The maximum radius of the GCA
- x_0, y_0 : Obfuscated coordinates of Query initiator

Output: A, The Globally Cloaked Area of the Query initiator that covers K-SCAs

1. A ← α
2. Let S(Pseudo ID, x_0, y_0, Boolean Result) be the set of SCA
3. S ← α
4. Compute SCA of Query initiator
5. x_0 ← X-coordinate of the centre of SCA of Query initiator
6. y_0 ← Y coordinate of the centre of SCA of Query initiator
7. AGCA ← Initial Area of GCA with circle R_l
8. for R = R_l to R_h step 1 do
9. for h = 1 to h_{max} step 1 do

i. if Compute SCA = true then add SCA to S
ii. Count ← Total count of Peers after Computing their SCAs and within the geometrical boundary of AGCA
b. if Count < K then continue else stop
10. if count < K the continue to step 8 else stop
11. Stop

Once the Optimal GCA has been computed, the query initiator randomly finds a query requestor from the set of available peers within the GCA. The query initiator performs a random selection operation over the set of *n* SCAs to select a query requestor to forward the query on behalf of query initiator to the LBS server. The peers are addressed with their pseudonyms that are sent back from peers, during the process of computing SCAs. The query is then forwarded to the query requestor, and it is submitted to the LBS server with GCA. Since the GCA is an obfuscated area, adversary may find it difficult to locate the query initiator. Moreover all IPs are encrypted with their pseudonyms. The LBS server returns a list of results applicable for this GCA. From this set of result the query initiator filter the values that he is interested in.

4. EXPERIMENTAL EVALUATION

We evaluate our system with Greedy based GCA computation in [5]. We adopted the selection metric from two classes: processing time at the query initiator (finding the GCA and the peers participating in the process) and peer response time for the query from the query initiator. This can be decomposed as the sum of communications delay (C_D) and Query initiator's processing time (Q_P). Peer response time (T_R) is the sum of the time required to calculate its cloaked area (T_C) and the communication overhead (C_D). i.e., ($T_R = T_C + C_D$). As both of the systems assume the same communication delay, this may be approximated as $T_R \approx T_C$. To compare the performance, we define average peer response time metric as shown in table 1.

n	No. of peers
t_i	Time that Query initiator requested the peer i
t_j	Time that Query initiator received the result from peer i
Avg_{RT}	Average response time, computed as $\sum(t_i - t_j) / n$

Table 1: Definition of average peer response time

As our system employs a simple transformation method to obfuscate peer location, the average response time comparatively less that other methods as shown in figure 5. The processing time Q_P is less compared to the methods in [5], because, the greedy algorithm that forms the GCA to meet K-anonymity, is run at the query initiator, whereas in our method, this process has been distributed among individual peers. The time complexity of the processing time for the greedy algorithm of GGC in [5] is O(n log n) where as our system the time complexity is O(h) where h is the hop of the peers that are included in the GCA. We set our simulation for experiments in this article on a Pentium 2.8 GHz and 1 GB RAM with varying K and R for GCA. For all K the system has shown remarkable performance compared to GGC in [5], because all SCA computation and the selection of optimal GCA are done at peers simultaneously, instead of computing at the system of query initiator. Figure 5 shows the average response time for GGC and for our system. We assume that the distributions of objects are normal. Figure 6 shows the computational cost for generating GCA for a set of 50 users. Since, in our system, the SCAs are

calculated at the peers, the value of the anonymity level K or the numbers of peer does not affect

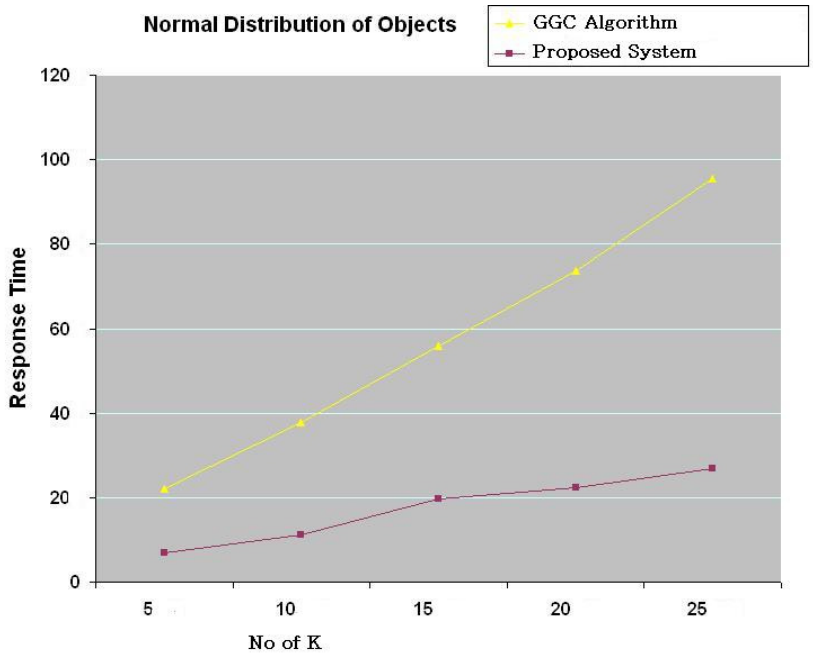

Figure 5. Response time for K subset of 25 SCAs

the total computational cost. All the computations are distributed and are executed in our system simultaneously. Where as in the case GGC the GCA is calculated by the query initiator and the computational cost of GCA is directly proportional to the value of K and the number of peers n.

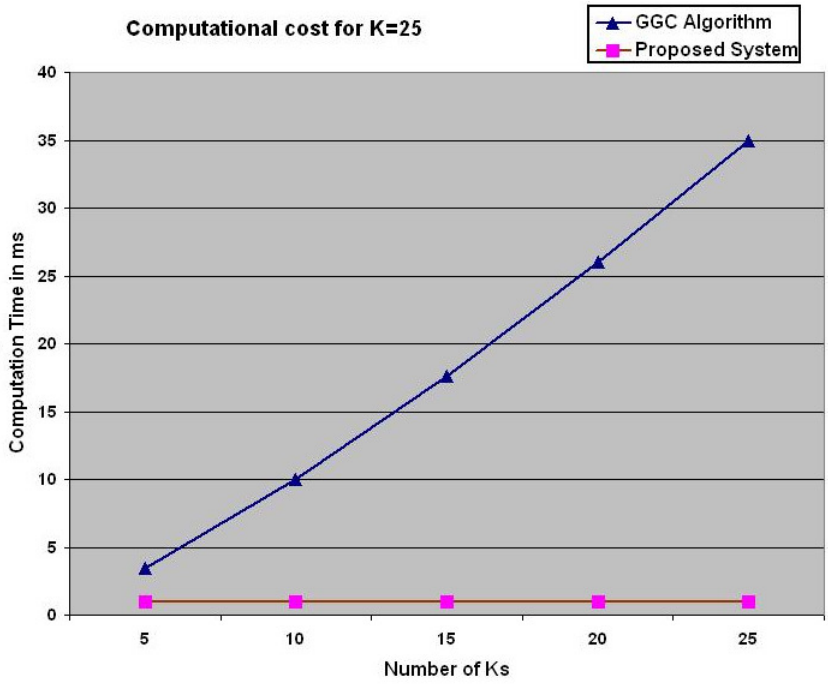

Figure 6. Computational cost for a subset of 25 SCAs

5. CONCLUSION

In this paper, we have presented a distributed location obfuscation method, to protect location privacy for Location Based services (LBS). In this case, the query initiator does not want to trust anyone including the peers and any third party service providers. Even peers do not reveal their exact locations, instead presents only their Self cloaked Areas. The system we presented fully distributes all computations, and even GCA calculation is done at peers. We have also presented algorithms which needs less computation time at the query initiator, compared to our previous works, due to the fully distributed approach. We also evaluated our system with different size of K-anonymity level and shows good accuracy and optimal results.

We plan to extend our work, with a network assisted approach, where the numbers of participating peers are less than the anonymity level K. We are also investigating the possibility of moving peers and dynamic Self Cloaked Areas (SCA).

REFERENCES

[1] J. Al-Muhtadi, R. H. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi. "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", In Proc. of ICDCS02, pages 74-83. IEEE Computer Society, 2002.

[2] C. Bettini, S. Mascetti, and X. S. Wang. Privacy protection through anonymity in location-based services. To appear in Handbook of Database Security: Applications and Trends, Springer, 2007.

[3] Yan Sun, Thomas F. La Porta, and Parviz Kermani, "A Flexible Privacy-Enhanced Location-Based Services System Framework and Practice," IEEE Transactions On Mobile Computing, Vol. 8, No.3, Mar. 2009, pp. 304-321.

[4] Lin Yao1, Chi Lin1, Xiangwei Kong, Feng Xia1, Guowei Wu1, "A Clustering-based Location Privacy Protection Scheme for Pervasive Computing", 2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing.

[5] Tanzima Hashem□, Lars Kulik, Don't trust anyone'': Privacy protection for location-based services, Pervasive and Mobile Computing 7 (2011) 44–59

[6] F. Gandon and N. M. Sadeh. A Semantic E-Wallet to Reconcile Privacy and Context Awareness. In Proc. Of ISWC03, volume 2870 of LNCS, pages 385{401. Springer, 2003.

[7] R. Wishart, K. Henricksen, and J. Indulska. Context obfuscation for privacy via ontological descriptions. In Proc. of First Int. Workshop on Location- and Context-Awareness (LoCA), volume 3479 of LNCS, pages 276-288. Springer, 2005.

[8] 2PASS: Bandwidth-Optimized Location Cloaking for Anonymous Location-Based Services Haibo Hu, and Jianliang Xu, Senior Member, IEEE

[9] M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in: MobiSys'03: Proc. of the 1st Int. Conf. on Mobile Systems, Applications and Services, 2003, pp. 31–42.

[10] M.F. Mokbel, C.-Y. Chow, W.G. Aref, The new casper: query processing for location services without compromising privacy, in: VLDB'06: Proc. of the 32nd Int. Conf. on Very Large Data Bases, 2006, pp. 763–774.

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

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

[13] B. Gedik, L. Liu, Location privacy in mobile systems: a personalized anonymization model, in: ICDCS'05: Proc. of the 25th IEEE Int. Conf. on Distributed Computing Systems, 2005, pp. 620–629.

[14] C. Bettini, X. Wang, S. Jajodia, Protecting privacy against location-based personal identification, in: SDM'05, Proc. of the 2nd VLDB Workshop on Secure Data Management, 2005, pp. 185–199.

[15] A taxonomy of approaches to preserve location privacy in location-based services Ali Khoshgozaran and Cyrus Shahabi

[16] SAMARATI, P. 2001. Protecting respondents' identities in microdata release. IEEE Trans. Knowl Data Engin. 13, 6, 1010–1027.

[17] SWEENEY, L. 2002a. Achieving k-anonymity privacy protection using generalization and suppression. Inter. J. Uncert. Fuzz. Knowl.-Based Syst. 10, 5, 571–588.

[18] SWEENEY, L. 2002b. k-anonymity: A model for protecting privacy. Inter. J. Uncert. Fuzz. Knowl.-Based Syst. 10, 5, 557–570.

[19] DUCKHAM, M. AND KULIK, L. 2005. A formal model of obfuscation and negotiation for location privacy. In Proceedings of the International Conference on Pervasive Computing.

[20] Chow, C., Mokbel, M. and Liu, X. (2006) 'A peer-to-peer spatial cloaking algorithm for anonymous location-based service', GIS, pp.171–178.

[21] Ghinita, G., Kalnis, P. and Skiadopoulos S. (2007b) 'PRIVE: anonymous location-based queries in distributed mobile systems', WWW, pp.371–380.

[22] Ghinita, G., Kalnis, P. and Skiadopoulos, S. (2007c) 'MobiHide: a mobile peer-to-peer system for anonymous location-based queries', SSTD, pp.221–238.

[23] Agusti Solanas, Josep Domingo-Ferrer, and Antoni Mart´ınez-Ballest (2008) 'Location Privacy in Location-Based Services: Beyond TTP-based Schemes'

[24] Muhamed Ilyas & R. Vijayakumar (2010) 'A Proxy based Framework for Efficient Range Query Processing in a Cellular Network', I.J. Information Tech. and Comp. Sc., 2, 1-8.

[25] Sridhar Subramanian and Baskaran Ramachandran (2012), 'Trust Based Scheme for QoS Assurance in Mobile Ad-Hoc Networks', International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.1

Authors

Muhamed Ilyas received his MCA degree from University of Kerala, Trivandrum, India. He is currently pursuing his PhD degree in the School of Computer Science, Mahatma Gandhi University, Kottayam, Kerala, India, under the guidance of Dr. R. Vijayakumar. His research interests include next-generation wireless system architectures, design and evaluation of location and service management schemes in mobile computing environments, and mobile database systems. He is a member of the ACM

Dr. R. Vijayakumar, working as Professor in the School of Computer Science, Mahatma Gandhi University, Kottayam, Kerala, India had completed 25 years of teaching career. His graduation is in B.Sc (Electrical Engineering) from College of Engineering, Trivandrum in 1984; and his M.Tech Degree in Computer Science & Engineering is from IIT Bombay in 1992. He was awarded the first Ph. D in Computer Science and Engineering in the faculty of Engineering from University of Kerala in 2000. Published 47 papers in National levels, International levels and journals in various fields of Parallel Distributed Processing and applications He is a consultant of Co-operative bank/Govt. institutions computerization and banking software/Hardware. He was member of Board of studies for Under Graduate and Post graduate studies in Kannur University and Mahatma Gandhi University, Kerala. Being the Senate Member of Calicut University, Kerala he is Chairman of Board of examiners in Computer Science and related subjects in Universities in South India. Completed sponsored projects of Govt. of Kerala and institutions in Kerala in the fields of Computer Network and applications. Apart from these he is research guide in Mahatma Gandhi University, Kerala and Anna University, Coimbatore, India.