# An Appraoch for Preserving Loaction Privacy using Location Based Services in Mobile Cloud Computing

Fizza Abbas, Rasheed Hussain, Junggab Son and Heekuck Oh
Dept. of Computer Science and Engineering, Hanyang University, South Korea
Fizza_alvi85@yahoo.com, hkok@hanyang.ac.kr

*Abstract*- ***Mobile Cloud Computing is today's emerging technology. Customers enjoy the services and application from this combination of mobile technology and cloud computing. Beside all these benefits it also increases the concerns regarding privacy of users, while interacting with this new paradigm One of the services is Location based services, but to get their required services user has to give his/her current location to the LBS provider that is violation of location privacy of mobile client. Many approaches are in literature for preserve location privacy but some has computation restriction and some suffer from lack of privacy. In this paper we proposed a novel idea that not only efficient in its protocol but also completely preserves the user's privacy. The result shows that by sharing just service name and a large enough geographic area (e.g. a city) user gets required information from the server by doing little client side processing We perform experiments at client side by developing and testing an android based mobile client application to support our argument***.

*Keywords*- ***Location privacy, Location based service and Mobile cloud computing.***

## I. INTRODUCTION

Mobile cloud computing (MCC) is an integration of cloud computing and mobile technology. Cloud computing is one of the latest trends in IT world. It is internet-based computing, whereby shared resources, software and information are provided to computers and other devices On-demand like the electric grid. This technology has the capacity to admittance a common collection of resources on request. It is proving extremely striking to cash-strapped IT departments that are wanted to deliver better services under pressure [1]. So the Mobile cloud computing is an infrastructure where both the data storage and the data processing happen outside of the mobile device. MCC provides many advantages like extending battery life time, improving data storage capacity and processing power, reliability, dynamic provision scalability, multi-tenancy and ease of integration. MCC has many applications areas including Mobile commerce, mobile learning, Mobile healthcare, mobile gaming and many more. The ABI Research predicts that the number of mobile cloud computing subscribers is expected to grow from 42.8 million (1.1% of total mobile users) in 2008 to 998 million (19% of total mobile users) in 2014 [2][*]. Many services are offered to facilitate users in MCC. One of them is Location based services (LBS), that uses the geographical position of a user to provide services such as health services like finding nearby hospitals, business services like locations of banks, similarly nearby restaurants, cinemas and the list just grows. There are various devices and techniques that can be used to detect the location of the user in the system. Some of the examples are Global positioning system (GPS) and RFID [2]. LBS provide tools to find the user destination (for example MapQuest), there are applications to link with users' Facebook and other applications and help them find nearby event. There are friend-finder applications and those related to social networking like Google Buzz. These and many more services have shown that LBS have opened an exciting new way of utilizing a mobile phone. With all these benefits of location-based services, we have the threat of compromising a user's privacy. LBS collect information of user location to provide him/her services increases the threat of violating user's location privacy. When a user uses location based services and send query to LBS provider, he/she sends his/her current location. A hacker can trace his/her activities from location information and can harass user or perform any kind of malicious activity with this information. LBS provider can also link user's visited locations and can deduce private information. This is one key issue because anyone can trace a user's activities if he/she finds the present or past

[*] This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education, Science and Technology (No. 2012-R1A2A2A01046986).

[*] This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-R1A1A2009152).

locations of a user [1]. Protecting user privacy and application secrecy from adversary is a key to establish and maintain consumer's trust in the mobile platform especially in MCC. In this paper we provide a novel idea on location privacy of user. With our novel approach we show that a user can still get services from LBS provider without ever giving any information about his her location.

The remaining paper is organized as follow. Section II highlights the related work and their limitations. Section III provides proposed scheme, implementation of scheme, results and analysis. Section IV gives conclusion and future work of the proposed scheme.

## II. Related Work

Many approaches have been put forward in this regard that have their benefits and also have their limitations. The various techniques include Cloaking, Private Information Retrieval, Generation of Dummies and Homomorphic encryption based techniques. Some of these approaches suffer from computational issues and other face lack of privacy.

Spatial Cloaking [3,4], also called as K-anonymity, is based on hiding user's location among the locations of few of his/her neighbours which are sent to a trusted third party (TTP) called anonymizer which hides user's position among those other users and sends sends service query for all of them to LBS provider. In order to do that, TTP (also called annonymizer) needs to be up to dated with a pool of users. This scheme suffers from various drawbacks. First, the use of a TTP means after all trusting on a third party. Also if TTP compromises then all other users' location comprimse. Another drawback is that, if a user asks for a service again then he/she can be identified among other k users. Cloaking or k-anonymity is effiecient in processing due to involevemnt of all calculation on TTP, but have high probaility of identifying user's location.

Generation of dummies [5] technique hide user's location and trajectory by sending several queries instead of only one. The drawback is slow server's response due to number of requests sent out by a user grows. Not only this but the LBS may suspect that it is under an attack and thus the requests may be ignored. Again the location privacy depend on number of queries send by mobile user. Also if this location information is exposed to the adversary, he/she can extract the true user's information. In this approach dummies must be selected intelligently otherwise it can easily reveal information about actual user.

Cryptrographic appraoches [6] guaratenees strong privacy at the cost of processsing. The techniques based on homorphic encryption are efficient in preserving the user's privacy because the LBS gets no knowledge from the encrypted query and sends back the required result to the client. But the high computational and communication complexity makes this approach impractical.

Private Information Retrievel [7] enable a user to query request to the database without revealing the request. This provide some how perfect privacy to the user, but suffer from computational cost or some assumptions about limitations of server's computational power. PIR protocols are expensive and require significant amount of server resources.

By looking at the various issues in above mentioned already existing approaches, we conclude that a robust yet simple approach is needed, which require moderate computational resources, normal secure communication between client and the server, avoids a TTP (trusted third party) and also avoids unnecessary computations (like searching for services by server for dummies or K-1 users).

Keeping in the mind all these aspects we present a novel scheme for using LBS services anonymously.

## III. Proposed Scheme

Our scheme is based on the fact that services in a Location-based services database are recognized by their longitude and latitude values (hereafter we will call them x and y for simplicity). In our proposed scheme a client's mobile first send a query to the server containing the required service and a region. A service can be a hospital and a region can be as much bigger as a metropolitan city. Our result later will help us determine this region. LBS server receives this request and searches it database for all records with that service name in that particular region and write their x, y (longitude and latitude) values in a text file and returns that file to the mobile client. At mobile client an application calculates the distance between client's location (client's longitude and latitude) with each of x, y in the text file and finds the nearest x, y value. After that this value is send back to the server. Upon receiving this value LBS server again matches this x, y with its database, finds the exact object and then returns all the details, i-e name and address to client. Fig 1. shows the overall working of proposed scheme.

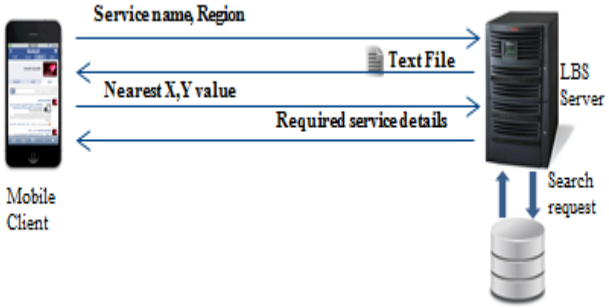


Fig 1. Working of proposed scheme

During interaction with each other neither server will ever know about client's current location nor will client ever know anything about server's database. We assume that this client server communication is secured using normal symmetric key cryptography. The data exchanged between client and server will be encrypted by one side and will be decrypt by other side. Moreover, client and server exchange shared keys in start of communication by Diffie-Hellman key exchange. Following is the algorithm for the scheme.

### *A. Proposed Algorithm*

- Client sends service name and region to LBS provider
- LBS Server searches all the records in the database of given service name under given region.
- Server writes all coordinates (longitude, latitude) values (x, y) in two columns in a text file and sends this back to client.
- Client receives the coordinates' text file.
- Client side application finds and sort distance between client's current locations coordinates with each of coordinates in text file sent by LBS provider and finds the nearest coordinate to client's location.
- Client sends these coordinates to server.
- Server receives coordinates and matches with its database and finds the exact record details.
- Server returns the service details to client.

### *B. Impelemntation*

One of the most important parameter to be checked in our scheme is the time taken by the client side application to calculate the distances from all the coordinates sent by LBS server in the text file and then to sort them to find the nearest service coordinates which will then be sent to the LBS server. This client side application was designed using Java extension for Android, in Android enabled Eclipse environment named as Android Developer Tools. The pseudo code for client application is given below.

```
READ currentX, currentY from GPS device
READ x_array[], y_array[] from "server_coordinates_ file.txt"
i=0, x = 0, y = 0, current = 0, previous = 1000, targetX = 0,
targetY = 0, dist[]
REPEAT
 x = x_array[i]
 y = y_array[i]
 dist[i] = SQRT((x-currentX)*(x-currentX) + (y-currentY)*(y-currentY))
 current = dist[i]
     IF current < previous    //comparison for sorting
         targetX = x
         targetY = y
         previous = current
     ENDIF
i++
UNTIL EOF
WRITE "Nearest coordinates=",targetX ,targetY
```

### *C. Results*

Our test bench was comprises of a Galaxy Note 2 mobile. Specifications are as under:

Processor: Quad-core 1.6 GHz Cortex-A9

RAM: 2 GB

Operating System: Android OS, v4.1.x (Jelly Bean)

After installing the Android Application Package File (APK) on the mobile we repeatedly executed it for many times and calculated application execution times by providing text files with different number of x and y values ranging from 50 in a file to 1000 in a file. Our results shows that client side application finds nearest coordinates among 500 coordinates in just 15 seconds and among 1000 records in just 29 seconds. As we performed our experiments on various text files with 50 to 1000 records; the time consumption for searching and finding the nearest coordinates from each of these text files.

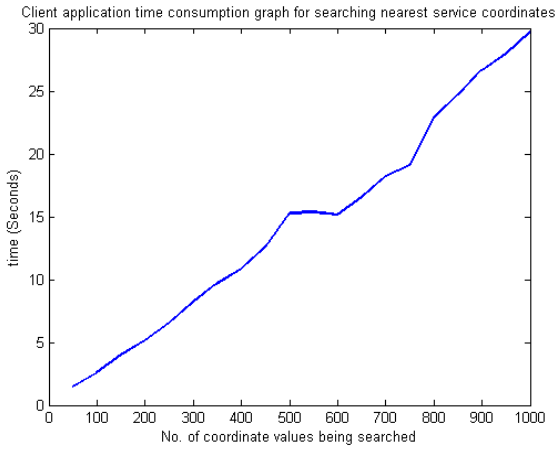


Fig 2. Time consumption for searching nearest coordinate on Client side.

The results in Fig 2. show that a large number of category can be searched in seconds on today's smart phone. It is worth mentioning here that a particular service in a limited geographic region (e.g. a city) is fairly limited in number. For example there are around 20 hospitals [8], around more than bank branches [9] and almost 130 restaurants [10] in Seoul, South Korea. With our application we can search the nearest service in seoul in just a couple of seconds as shown in the graph. The linear graph also indicate that the region can be extend by analyzing what is the tolerable execution time limit a user can wait to ensure his/her privacy is conserved while still getting the required serice anonymously. We prove through the pratical implemnataion in Java on andriod phone, that todays mobile can easily calulates these number of records in seconds and client get required information in seconds.

*D. Analysis of scheme*

In this section we analyze the performance of our novel scheme.

*1) Guarantee of anonymity*

In our scheme a server never get to know about the location of the client at all. This is one remarkable achievement of this scheme. Client only sends service name and region, but not his/her location.

*2) No use of TTP or Anonymizer*

By not utilizing any trusted third party or anonymizer we not only save time, but also we are free from having to trust someone after all. Not only is this but also there is not threat of being exposed in case of the compromise of TTP or anonymizer.

*3) No computations for dummies*

As our scheme is not sending extra or fake client's location coordinates, hence the server will not have to do computations for fake location data. That will save precious server resources and time. The server will make a normal search to its database to find out the matched records and will simply write them to a text file for sending it to client.

*4) Practical client side application execution time*

The time execution graph for our client side application clearly shows that this much processing can be easily handled by a commonly available smart phone. As we have discussed various cases for number of services in a large city, our application performs smoothly for such large enough region.

*5) A simple scheme for implementation*

Our scheme does not require any special hardware or software for implementation on not only at client side but also at server side. With normal search queries the server retrieves the records from database and a simple text file is exchanged between server and client.

*6) No threat of communication being caught*

Another remarkable feature of this scheme is that if this client-server communication is ever compromised, the adversary will only get a location coordinates text file sent from server to client or a location sent to server by client. In any case client's exact location is perfectly secure as well as server's database.

## IV. CONCLUSION AND FUTURE WORK

We have presented a robust scheme with privacy preservation for Location-Based Services. We have made and tested a client side application which executes fairly quickly to get the desired results. In our scheme LBS provider cannot know about client's actual location, similarly, client application cannot know anything about server's database. LBS provider can only know the nearest service to client, but it cannot reveal the exact position of a client. In future we aim to enhance our client side application and we try to optimize our code to calculate the results more quickly. We suggest that we can enhance our application more if our application send the nearest coordinate as well as randomly chosen coordinate from text file so that server can never even assume that which service is nearest to the client.

## REFERENCES

[1] H. T. Dinh, C .Lee, D. Niyato and P.wang, "A survey of mobile cloud computing: Architecture, application and Approaches," accepted in wireless communication & mobile computing-wiley.

[2] A. Nasir Khan.M.L Mat Kiah, S.U.Khan, "Towards Secure Mobile Cloud Computing: A survey," Future Generation of Computer system, ELSEVIER, July 2012.

[3] S.Wang and X.S.Wang, "In device spatial cloaking for mobile user privacy assisted by the cloud", *11th IEEE International Conference on Mobile Data Management*, pp. 381-386, 2012.

[4] M. Gruteser, and D. Grunwald, "Anonymous usage of location based services through spatial and temporal cloaking," *ACM 1st international conference on Mobile systems, applications and services*, pp. 31-42, May 2003.

[5] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location based services," *IEEE International Conference on Pervasive Services*, pp. 88-97, July 2005.

[6] Y.Gahi, M.Guennoun, Z.Guennoun, and K.El-Khatib, "Privacy preserving scheme for location based services," *Journal of Information Security,* pp. 105-112, April 2012.

[7] A. Khoshgozaran, C. Shahabi, "Private information retrieval techniques for enabling location privacy in location based services," *Privacy in Location-Based Applications, Lecture Notes in Computer Science*, Vol 5599, 2009, pp. 59-83, Springer Link.

[8][Online]http://www.allianzworldwidecare.com/hospital-doctor.

[9] [Online]http://www.theswiftcodes.com/south-korea/

[10][Online]http://www.mytravelguide.com/restaurants/ctrestaurants.