MaPIR: Mapping-Based Private Information Retrieval for Location Privacy in LBISs

P. M. Wightman, M. Zurbarán, M. Rodríguez
Departamento de Ingeniería de Sistemas
Universidad del Norte
Barranquilla, Colombia
{pwightman, mzurbaran, erodriguezm}@uninorte.edu.co

M. A. Labrador
Department of Computer Science and Engineering
University of South Florida
Tampa, FL, USA
mlabrador@usf.edu

***Abstract*— The popularization of Location-Based Services (LBSs) has brought along benefits for users and service provider, in terms of improved quality of existing services and a better user experience. At the same time, location privacy has become one of the most critical concerns for ensuring users' right to protection. Despite the fact that one of the best ways to protect the location information is not to reveal it, there are advantages on using this information to personalize services, however it is necessary to guarantee its protection. Private Information Retrieval is a technique that creates a common language between users and service provider so that external actors cannot understand most of the information being transferred. This paper introduces MaPIR, a mapping-based private information retrieval technique that uses mathematically generated mapping to create redundancy in order to provide multiple answers to a user with an undistinguishable location. This technique is decentralized and focuses on Point of Interest Search-based application, not on tracking services. For performance evaluation, were compared in two scenarios MaPIR, a regular spatial query and the Dummy Query technique, the results show that MaPIR takes only half the time of regular geographical queries on database processing, and 5 times less than the Dummy Query technique, while providing a similar level of redundancy.**

***Index Terms*— Location obfuscation; privacy; point of interest; geographical query; LBIS**

I. Introduction

Location-Based Services (LBS) provide users with information based on their geographical position, which can be obtained from the mobile device they are carrying and from which they are accessing the service, or using a manually defined location [1].

Due to a global growth trend in the last years in the demand and offer of Location-based services and location-based information systems (LBISs), which was made possible thanks to factors like the development of faster cellular-based Internet communication technologies, the worldwide adoption of smartphones with localization capabilities, and the availability of open source APIs to work with maps, geographic databases, etc., many large companies have included in their business model the location of their users as a critical ingredient of the services they offer.

For many users, the inclusion of location and the associated features signify more personalized services, which satisfy their needs faster and more accurately. However, this opportunity also brings some associated risks such as: The service provider now have access to the location information of its clients, which could be misused for purposes other than the one initially defined; such as, selling this information to third party companies which could use this data for spamming, undesired advertisement, or even worse, the information could be obtained via a security breach and used later against the integrity of the users, potentially putting them in danger.

In general, this problem is related to privacy issues, in the matter of location information. Many governments have issued legislation about this topic [2, 3, 4]. In the literature there are many approaches to protect the privacy of the users, but still today real policies to protect this information have not been implemented.

The best and most trivial way to protect the location information is not to share it, however LBIS will not work properly, or may not work at all, thus the user will not be able to enjoy the advantages of these applications.

The main argument is that the service provider does not need to have the exact location of the user in order to provide an adequate level of service. In addition, it is desirable that, in case of a security breach, system logs will not have detailed information about the user's activity to determine their location history.

One way for LBIS to accomplish this is by using Private Information Retrieval (PIR): A technique that allows a communication between users and a service provider, based on a known standard that, despite been possibly known by external actors, still allows interaction without receiving or revealing location information. PIR was first introduced in [13] as: "*…schemes that enable a user to access k replicated copies of a database (k≥2) and privately retrieve information stored in the database. This means that each individual database gets no information on the identity of the item retrieved by the user*", but this technique was not initially intended to be used on a single database, until [14].

This paper introduces the Map-based Private Information Retrieval algorithm (MaPIR), that allows the creation of a geographical map, mutually understood by users and service provider, which offers redundant responses to service queries in order to hide the actual location of the user or its intended area of interest and also protects the location information of the

978-1-4799-0540-9/13/$31.00 ©2013 IEEE

user in the query logs. One of the great advantages of this technique is that, requires minimal preprocessing and does not increase the load of the database engine, which is the main downside of classical PIR. This technique is focused on Point of Interest Search-based application, and it is not intended for tracking services.

This paper is organized as follows: In Section II, some related work in the area will be presented. In Section III, the proposed technique will be presented. Section IV will present the experimental design and Section V will present the results of the experiments. Finally, Section VI will be dedicated to the conclusions and future work.

II. Related Work

Location privacy has been an active topic of research for the last few years, as a way to protect the location information of the users and avoid arbitrary use to establish mobility patterns of the users, favorite places, etc. in order to put their integrity at risk. Privacy in LBISs can be divided into two general fields: Identity Privacy and Location Privacy.

Identity Privacy includes the mechanisms that focus on protecting a subject's identity, either inside the service infrastructure (how is the user identified in the information systems, i.e. nicknames, profile information, email, pin, etc.) or in the Internet infrastructure (how is the user identified in the communication infrastructure, i.e. IP address, MAC address, LAN name, etc.). In these techniques, location is not actually protected but the user who is reporting it, thus location becomes meaningless because it cannot be associated with a single user. However, these techniques are not in the main focus of this paper.

Location Privacy can be defined as "*The right of individuals to decide how, when, and for which purposes their location information can be released to other parties*", either via transforming the user's obtained location when reported [6], adding random noise or other mathematical transformations [11]; altering the query process so that the actual location information is not revealed directly, i.e. generating redundant random queries [5, 12]; or Private Information Retrieval (PIR) methods, which also aim to provide location privacy and are usually supported on mapping the spatial context in order to provide a private way to query the service without releasing any location information that could be obtained by third parties.

The transformation-based approach assumes that the user can define a strong privacy policy (for example, a large enough radius of generalization for noise adding techniques), but it can be shown that these techniques still reveal the whereabouts of the users, and that, if the added noise is too large, the server may not be able to answer the queries properly. For example, if the user reports its location 2 miles from its current location, which in terms of protection may be acceptable, it may show restaurants that are too far, compromising the overall user experience.

Query-based techniques like [5, 12] provide location privacy by generating N random fake or dummy queries for arbitrary locations on the greater search area (city, county, state). All these requests are formed by a coordinate and a specific type of location to query (restaurants, malls, drugstores, etc.). Queries are sent individually and sequentially including the one containing the user coordinates and location interest, the order in which the fake and original request are sent is random to increase protection.

The main disadvantage of the dummy query technique is that it is necessary to send N requests each time a query is issued; the greater the value of N the higher the location privacy; on the down side, the overall cost of the transaction also rises, specially on the server side, which needs to process many redundant queries that are not critical for the users. This overhead in terms of computation and communication may not be scalable for large systems. In addition, one of the queries will contain the original location and interest of the user so the original trajectory could be traced based on the query records from a user, especially if no intelligence is applied when generating the dummies. There are improvements to this method in [9] where rather than sending each point on a separate query, all the dummies and real location are sent along with the location interest specified by the user. In [15], the authors propose a method to avoid the random generation of points for each iteration, which should reduce the possibility of detecting the trend in the real points, but this technique requires many resources from the device for generating trajectories for each dummy path, generates separate queries per path and still reveals the user's location.

PIR-based methods offer a higher level of privacy but usually require special implementation on both client and server sides. Authors in [7, 8] introduce methods that use PIR to ensure location privacy. Most of techniques used to retrieve information privately from databases could be treated as black boxes and therefore other mapping techniques could be applied to the same PIR architecture.

PIR could be cryptography-based or hardware-based according to [7]. Hardware-based methods use a special kind of secure coprocessors (SC) that act as a securely protected space in which the PIR query is processed in a way that the service or any other agent cannot decipher, as in [7]. Cryptography-based techniques only use logic resources and do not require special physical disposition on server nor client, which is why MaPIR fits under this category.

In [16], the authors present a hybrid technique that uses a cloaking method through various-size grid Hilbert curves to limit the search domain of a generic cryptography-based PIR algorithm. However the PIR processing on the database is still expensive, as shown on their experiments, and is not practical for a user-defined high level of privacy. The method does not allow the cloaking grid cells size to be specified by the user and cannot be changed once the whole grid is calculated; this is, no new PoIs can be added to the system.

III. The MaPIR Technique

The MaPIR algorithm is a cryptographic PIR-based technique for PoI-searching on location-based information systems, that protects the information shared between user and server, provides redundancy, in order to make the query

location indistinguishable, without introducing large overhead to the system. In Section IV, MaPIR will be compared to the Dummy Query technique and to the use of no privacy protection technique, in terms of server overhead in query time and number of query results.

MaPIR uses a low complexity mapping function to divide a geographical area into grid cells identified in a non unique manner with repeated IDs. The user does not report its current location but the ID of the grid cell that she is interested on for consulting a certain type of PoIs. This is the main difference with the DQ technique, since DQ does report N+1 locations and one of those is actually the user's.

Also, the fact that there are cells with equal IDs means that the actual location of the user cannot be distinguished among all the instances with the same identification. These key characteristics of the solution guarantee that both private interpretation and redundancy are included in a privacy preserving solution. The mapping function and the general process of MaPIR will be explained now.

A. Mapping Function

The main aspect of the MaPIR technique is the way the mapping is done. Many possible mapping functions can be implemented, but the following criteria is suggested: a) The technique must generate redundant identification of the grid cells; b) the redundancy must be a non-trivial geographical function, i.e. the same ID's should be uniformly sparse over the area; c) the function must be computationally low-cost because it will be executed often on mobile devices. This paper presents one solution from a large variety of functions that comply with these characteristics.

The mapping function introduced here works in two phases: Location Reduction and ID calculation.

The Location Reduction phase takes as parameter a geographical location point (latitude, longitude) and applies a technique to transform each of the dimensions into a number that still contains information about location, without revealing the original one. In addition, this reduction technique should also preserve the geographical relation between near-by points of interest, and should be able to manage different scales (hundreds of meters, kilometers, tens of kilometers, etc.) in order to adapt the searching area to the user's scope of detail. This will guarantee that geographically nearby PoIs will be returned like in a regular geo-query.

The second phase is the ID calculation, with the numbers generated in the location reduction phase for each latitude and longitude, calculates the ID of the containing grid cell. The function must provide a non-unique identification of the grid cells, such that they are sparsely- and randomly- distributed.

The proposed Location Reduction function in this work is the following:

1) Kth Digit Location Reduction

This technique is based on a simple idea. Approximately, $9e10^{-6}$ degrees is equivalent to a meter in the Equator (360 degrees by 40400 kilometers). Thus, each significant digit on the location represents a scale of the distance, like shown in Fig. 1.

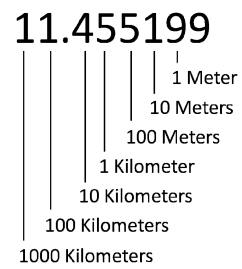

Fig. 1. Reduction based on scale

For example, 1e10-2 of a degree is closely equivalent to 1 kilometer. This means that locations can be grouped by the kilometer in which they are contained; this is, the second digit on the decimal portion of degree. For example, the following two latitudes 11.455199 and 11.455891 produce the same reduction number, 5, thus they are within a kilometer from each other. In addition, this reduction allows grouping in different scales, depending on the search area defined in the application.

Formally, the proposed function is the use of the *k* th least significant digit from the decimal portion of the location information, in order to reduce this information. The main advantages of this technique are: It is very simple to calculate; the reduced values are always in the same range [0-9]; and it can be adapted to different search scales, without increasing the complexity.

The next section will introduce how the grid identifications are calculated based on the reduced latitude and longitude.

2) Pseudo-random ID Calculation

Once both latitude and longitude have been reduced, these values are used to define the ID of the grid cell in which the point is located. The proposed technique uses a mathematical function based on a well known simple pseudo-random number generation.

Let the reduced values of latitude and longitude be *i* and *j*, and let *p* be the next prime value after the maximum value that the reduction function can generate. Then, the ID of the grid cell where the reduced point is located is calculated using (1).

$$ID(i, j) = ((i + 1) * (j + 1)) \bmod p \qquad (1)$$

For the kth digit location reduction function, the *p* variable should be 11 as it is the next prime number after 9, which is the maximum value that the location reduction function can generate. The addition of 1 to the reduced components of the coordinates is necessary in order to avoid the value 0, which would nullify the multiplication. The final grid for this mapping function is shown in Fig. 2.

Each ID exists p-1 times in a single grid (10 in this case). This implies that the resultant grid cell of a single location point queried by a user will be indistinguishable from other p-2 in the same grid, so there is not a unique geographical correspondence. The p value must be a prime in order to guarantee that each id exists on each row; otherwise, all ID values multiples of p will generate redundant patterns in a single row.

j

i	10	9	8	7	6	5	4	3	2	1
10	1	2	3	4	5	6	7	8	9	10
9	2	4	6	8	10	1	3	5	7	9
8	3	6	9	1	4	7	10	2	5	8
7	4	8	1	5	9	2	6	10	3	7
6	5	10	4	9	3	8	2	7	1	6
5	6	1	7	2	8	3	9	4	10	5
4	7	3	10	6	2	9	5	1	8	4
3	8	5	2	10	7	4	1	9	6	3
2	9	7	5	3	1	10	8	6	4	2
1	10	9	8	7	6	5	4	3	2	1

Fig. 2. Mapping function

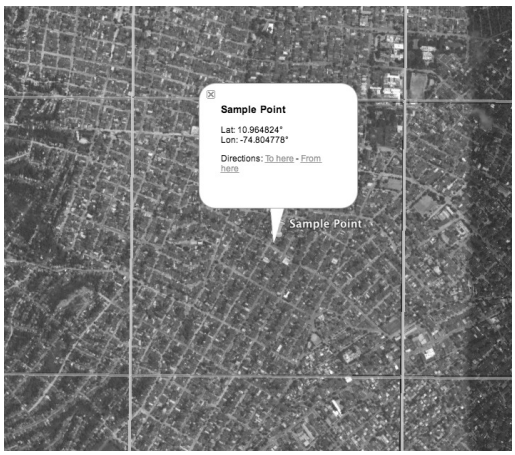

Fig. 3. Example of actual map with a grid scale of 2 on Google Earth and a PoI on a cell with ID of 4.

Fig. 3 shows a sample point mapped on a grid with a scale of 2 and ID equals 4 using Google Earth, note that a single location will have different IDs depending on the scale; for example, if the user is in a location (10.964824,-74.804778), the final mapping grid cells would be 10, 4 and 3, using scales k = 3, 2 and 1, respectively. TABLE I. summarizes the corresponding ID's of this location.

TABLE I. MAPPING FUNCTION FOR A SINGLE LOCATION USING DIFFERENT SCALES

k	Reduced Latitude	Reduced Longitude	i	j	Id
3	4	4	5	5	3
2	6	0	7	1	7
1	9	8	10	9	2

The scale allows dividing the area of search in different ways in order to restrict the amount of results offered by the system and also to generalize the location of the user. Basically, each grid cell can be divided in another 100 smaller grid cells based on the following order of magnitude, for example, Fig 3 shows a single location inside a grid cell with scale 2. A complete new grid could be defined inside it, using the scale 3, which works with hundreds of meters. Thus, the user could restrict the scale of the search area. It is very important to mention that the smaller the search area is, the least protection the technique can offer, because the user's location is just generalized within the search area.

To illustrate, in Fig. 6, a single grid of scale 1 can cover a medium-sized city. If larger areas are needed to cover the search area, multiple grids can be included in the domain area. The grids will be identical replicas of each other.

B. The MaPIR Process

The overall MaPIR process involves both user and server on two stages. The first one runs on a mobile device, capable of calculating the user's location through GPS, Global Systems for Mobile communications phone protocol (GSM) or Wi-Fi spots as explained on [10]. For the server side, the provider must have a database of PoIs that has been translated using the mapping function.

The MaPIR technique in the mobile device compounds two parts: the querying process and the results processing. On the server side, there are also two parts: preprocessing the PoIs, and the user querying process. The processes are explained now with more detail.

1) Preprocessing of the PoIs

This part of the MaPIR process has three stages: Data Preparation, Standard Mapping and Augmented Mapping.

For the first stage; Data Preparation, the service provider should identify the geographical areas where the service will be available in order to calculate the grids for those areas. If the area covered is bigger than a city, a country for instance, there will be too many irrelevant PoIs retrieved per request, if no limits are applied.

On the second stage, location reduction and mapping functions must be defined previous to the processing of the PoIs. Depending on the techniques, the parameters must be decided; in the case of the ones presented in this work, it is required from the user to define the values of the scales that will be used. Then, all of the PoIs are mapped to their corresponding grid cells, based on their location, for all the different scales offered by the service provider.

Every time there is a new point added to the systems, this process has to be performed just for the new location without affecting the rest of the points. The mapping must only be restored completely if the mapping technique itself is altered. This table, as shown in Fig. 4. , will be the one used to answer requests from clients.

POI ID	POI type	Cell ID Scale 3	Cell ID Scale 2	Cell ID Scale 1	Search Area
1	Restaurant	5	10	2	1
2	Restaurant	2	4	2	1
3	Gas Station	4	3	2	1
4	Theater	7	6	2	1

Fig. 4. Structure of the PoI-ID table in the server's database.

2) User location and query processing

This part of the MaPIR process is divided in three stages: Query Construction I, Location Processing and Query Construction II.

On the first stage, the application in the user's mobile device, when triggered by the user or by predefined criteria, will ask the user for information relevant to the query: type of PoI being searched (restaurant, gas station, etc.) and the scale of the query, the user may specify more than one type of PoI in order to disguise her real interest. This scale can be selected depending on the level of detail the user wants from the query. For example, a smaller scale will produce fewer results, but will reveal more detailed information about the user's location.

The device then calculates its current location via GPS or any other location mechanism. Based on this location, the device will apply the reduction method defined and calculate the corresponding IDs of the grid cell in which the user is currently located, at the different scales, starting with the one specified by the user for the query, i.e., if the user wants to search something at scale 3, the application will calculate the grid cell in scales 3, 2 and 1. These multiscale calculations are necessary in order to identify the general geographical zone from where the individual is querying the system especially in applications with information from multiple cities, given that there is a large set of possible cells with identical IDs at different scales; for example, if the scale 1 grid cell is not specified, the system may not be able to differentiate between different cities with scale 2 cell distributions. Depending on the size of the possible search domain, then some of the higher scale calculation may not be necessary or maybe even more scales would be required (if the areas covered are at a country or continent level). Finally, the complete information, including the PoI type and the different grid IDs per scale, will be sent in the query to the server.

3) Server query processing

The LBIS has a database that contains every point of interest with the associated augmented grid cell IDs per scale offered and the type of location. With this information, the server will send back to the user all PoIs that match the IDs that were requested. This means that the server will include the PoIs in both its actual location and all the grid cells with same ID at the scale defined by the user. The other IDs will be used to define the context of the search. This means that the server does not have any idea about the real location of the user, thus the location privacy is protected.

MaPIR differs from the Dummy Query technique that will be used for testing in the following sense: DQ technique sends the user location along with the dummy ones in a set of locations reported, thus the location is at last revealed, and could be found with trivial attacks; also, DQ suppose the processing of N+1 geographical query predicates, which imposes a higher cost on the service provider's infrastructure, while MaPIR is able to obtain redundant results with just one non geographical query.

4) User query result processing

The user will receive from the server the complete list of PoIs that matched the ID sent in the query. The results will include the exact location of the PoIs, as shown in Fig. 5. Then the application, which knows the exact location of the user, can filter out the PoIs whose locations are too far to be relevant and keep the ones that are closer to the user.

POI Name	Latitude	Longitude
Restaurant 1	10.8848552	-74.7773735
Restaurant 2	10.8867316	-74.7781744
Restaurant 3	10.8876633	-74.7740786
Restaurant 4	10.887011	-74.7774279
...		

Fig. 5. Structure of the response from the server

C. Defense Model

MaPIR is a secure way to perform queries, mainly because the actual location of the user is never shared, and that, even if an attacker knows the mapping function, very little could be done to infer the location of the user. One possible scenario of attack is if an attacker could obtain the log of MaPIR queries and the LBS server response. Even in this case, the attacker still could not identify any location on any of the intervened information since the results will include only the PoIs and their location, without any discrimination of the actual grid cell from which the query was generated. This makes undecipherable the whereabouts of the client.

With the application of sophisticated techniques of data mining, the attacker could be able to infer a connection between the queried locations that make more sense in the context of the user, and could be able to identify grid cells and discard cells in which the user is least likely to be. In addition, if a number of queries are generated during the day, a path of cells could be put together, however this is not always the case since the assumption is too vague and could be misleading. Besides, as seen on Fig. 2, the mapped grid forms a symmetric matrix, which will give results tracing at least two possible paths to describe the user trajectory. In addition, given that the IDs do not change in time, if the user does not change drastically his position while querying the system, the redundant grid cell will stay the same, while in DQ there is a chance that the cluster of nearby queries can be detected easily if the random locations across queries do not show any potentially real pattern.

IV. Performance Evaluation

This section is devoted to testing MaPIR and comparing it with the dummy query-based algorithm in [9], which was chosen because of its similar functionality in terms of added redundancy to hide the original location of the user and the fact that requires only one query to the server per point. In addition, and in order to evaluate the overhead imposed by the location

privacy protection techniques, regular unsecure geographical queries are also tested.

A. Experimental design

The performance of these algorithms was tested based on the following scenario: 30 users query PoIs on different paths formed by 20 locations each. Every user connects simultaneously and sends a query with a random PoI type per location visited; they should receive the PoIs close to the reported location on every request of the specific requested type. This scenario will be executed once per querying technique: Dummy Query, MaPIR and regular query.

The DQ algorithm implementation for the experiment sends a single request containing a location interest, N dummy locations and the real user location. This request is sent to the server in the form of: $Q = (pos_1, pos_2, \ldots, pos_n, pos_{n+1}, LI)$, where LI is the location interest, and then processed by a geographical database engine.

Two variables are tested for every query: average query processing time, measured from the moment the request is sent to the server until the moment results are retrieved; and the average number of resultant PoIs obtained. These values are obtained on a per user's basis, and then shown per scenario in a box plot to illustrate the variability of the data.

The preprocessing from MaPIR on the server side was not considered because, given that this process only takes place once, the overall impact in the long run is minimum. The tests were performed directly on the server machine for all the techniques, in order to measure in a more precise manner the load on the server and to avoid variations caused by network traffic. The server machine is a Mac Mini, dual core processor, 4 GB RAM, 1TB HD, running a PostgreSQL 9.1 database with PostGIS 2.0 to execute spatial queries. The paths were obtained from real tracking traces from Barranquilla, Colombia. The PoIs were randomly generated and uniformly distributed in the square area.

TABLE II. EXPERIMENTAL PARAMETERS

Technique	Parameter	Value
MaPIR	l	0.01°
	Δ	0.0005°
	p	11
	Scale for queries	2
Dummy Query	N	9
	r	0.055°
Regular Query	r	0.055°

TABLE II. summarizes the parameters for the evaluated techniques. The values for the DQ algorithm and the regular geographical query where defined to match some of the characteristics of MaPIR; for example, the value of N equal to 9, so the overhead is similar, and the radius value for the geographical queries was set to 0.0055°; which results on queried areas of $(0.000095°)^2 \approx (0.0001°)^2$ for each request. The total of PoIs present on the database for the three tests was 1000 and the database was the same for all techniques.

Fig. 6. and Fig. 7. show an example of the results from a query using Dummy Query and MaPIR, respectively. The query was used for a location with the coordinates (10.968901°, -74.831048°) and the PoI type was set to search for parks. In the case of Dummy Query, there are a total of 7 PoIs returned and only 1 that satisfies the real query out of 72 parks in the database.

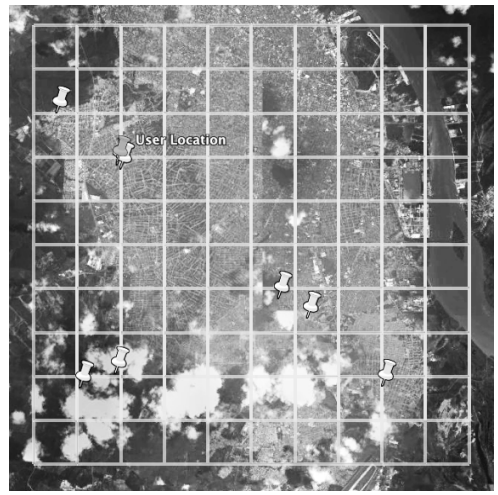

Fig. 6. A Dummy Query illustrated results on actual map using Google Earth.

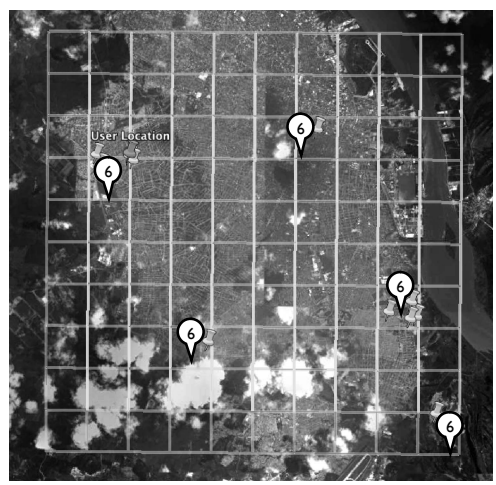

Fig. 7. Results for a MaPIR request on a grid with scale of 2

The results obtained by the same request using the MaPIR technique show that there were 9 results of the specified PoI type on 5 locations out of 10 possible with ID equal to 6 and there are 2 relevant results near the real location. In Fig. 7, the numbers on the white balloons show the ID of each cell that contains results.

B. Query Processing Time

Fig. 8 shows the results of the query time results of each method on the 30 routes and 1000 PoIs. In terms of processing time, most of MaPIR queries are performed in half the time, compared to the regular geographical queries, and 5.5 times

faster than Dummy Query. The fact MaPIR's query time has the lowest among the ones from the regular geographical query and the Dummy Query ones can be explained by the fact that spatial queries need extra processing on the database engine, compared to the simpler queries generated by MaPIR, in which all PoIs are associated with numerical IDs at the preprocessing stage. Thus, the redundancy that MaPIR generates not only does not tax the server but provides even a more efficient way of dealing with geographical PoI-based services.

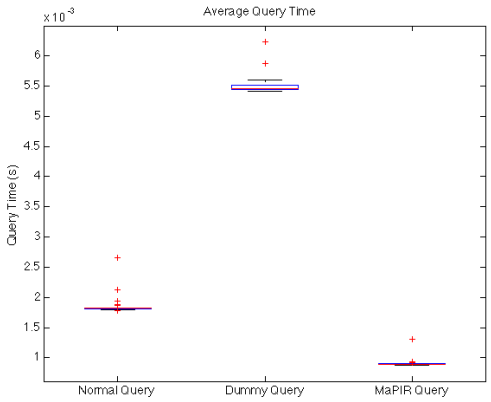

Fig. 8. Query Time Box Plot with 1000 PoIs

C. Query Results

The same statistics were calculated for the number of registers returned on each request of the 30 paths used for testing. The results are shown on Fig. 9. This variable tests if the level of redundancy offered by MaPIR and Dummy Query are similar. The number of results retrieved by performing a regular query is the shortest, since the area is precise and limited to the reported locations. The number of Dummy Query retrieved results is similar to MaPIR's on a grid with a scale of 2, being around 10 times larger row count than using a regular geographical query.

These results show that the level of redundancy offered by these 2 techniques is independent of the query processing time, given the large difference between MaPIR and Dummy Query processing times and the very little difference in redundancy.

D. Discussion

An important issue with privacy protection techniques is how well the location information is actually protected. Dummy Query protects the user's real location by generating 9 random points per actual location. On Fig. 6 and 7, the results of a single query can be seen. They both look very similar in terms of protection in the sense that the original location is indistinguishable from the others. In this case, both DQ and MaPIR offer similar protection against attacks, without taking into account performance.

Fig. 10. and 11 show all the location that would be returned by the query logs, in the scenario in which the user performed multiple queries in a short period of time and without changing much his location. The original locations are highlighted as white marks in Fig. 10. Despite the fact that the high number of dummy locations makes it difficult to visually identify patterns when all the locations are shown at once, the main problem of this technique is that the real locations are revealed and that by using simple data mining techniques and including time of each request in the analysis, an attacker could infer the actual path.

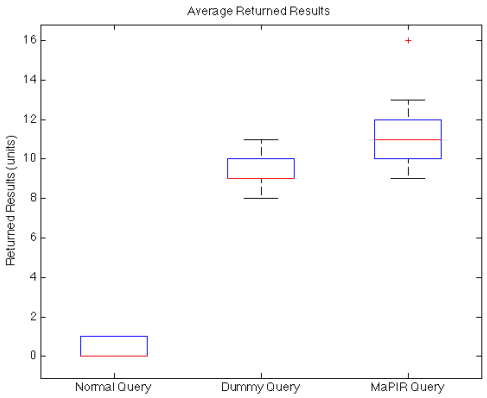

Fig. 9. Rowcount Box Plot with 1000 PoIs

In the implementation of MaPIR there are 10 predefined squared areas that are processed for each query, not specific locations. Thus, as it can be seen in Fig. 11. , just the grid cells are revealed for the same requests performed with DQ, thus the actual locations are never accessible. In the case of a user that does not move much and remains in the same cell, the sum of results of several queries will look just like single query, because the set of cells will always be the same, thus there will be no extra information that could be used for discarding fake locations. In DQ, multiple queries in the same location could be visually obvious, especially if just random locations are generated. In addition, the grid used by MaPIR, thanks to the mapping function defined, is symmetric, thus at least two paths could be inferred by an attacker if the user moves along several cells. Despite the fact that some techniques based on context could be used to discard one of them, still there is very general data that could be used for such attack. This problem magnifies when using a greater search area, like a scale 1 grid (100km2) which retrieves registers from 10 possible grids of scale 2 that are also symmetrically distributed.

V. Conclusions and Future Work

The MaPIR technique is a private information retrieval – based privacy protection technique, which enforces location privacy and allows querying PoIs with great accuracy, in an efficient manner, and without revealing the actual location of the users. As seen on the performance results, MaPIR is able to protect the location information of the user by generating redundant information to hide the original source of the query, obtaining similar protection results compared to the Dummy Query technique. In addition, this redundancy is shown to generate no overhead of the server side, and even, thanks to the mapping function, the preprocessing and the nature of the queries performed under MaPIR, it shows even better processing time than the regular geographical queries without any privacy protection technique implemented.

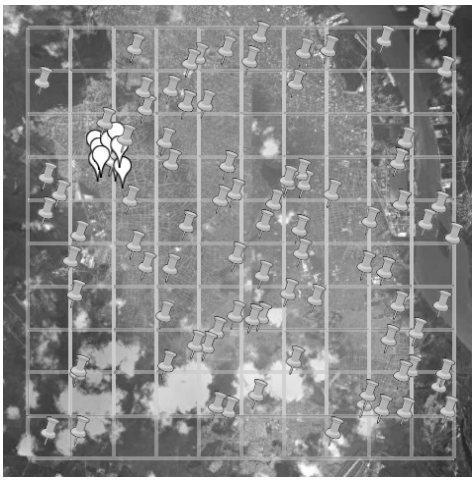

Fig. 10. Locations reported to the LBS using Dummy Query.

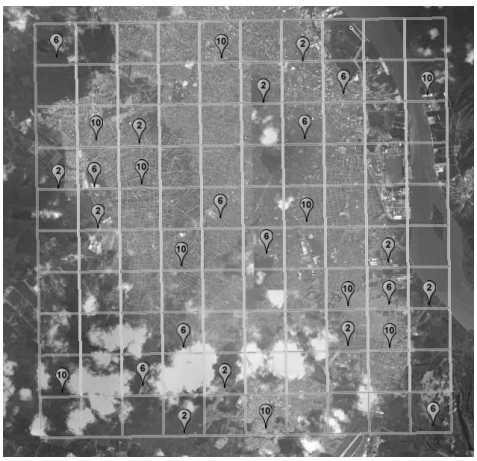

Fig. 11. Areas reported to the LBS using the MaPIR technique.

For future work it would be pertinent to implement different mapping techniques to work with MaPIR, such techniques would provide mapping with different redundancy and options for adjusting the number of results retrieved, and it is applicable to different scales. For testing the privacy offered by the method, it is being adapted to be evaluated by the testing framework proposed in [17], which is a formal technique applicable to location privacy protection mechanisms in order to quantify the real level of protection, based on the estimation of the probability of the user being in a certain cell. In addition, it is necessary to define a way in which the service provider could obtain some feedback about the decision that the user made based on the information returned by the query, so the system could offer a better service for future similar queries.

ACKNOWLEDGMENT

This project received founding from Colciencias and Universidad del Norte - Young Researcher Program 2012.

REFERENCES

[1] A. Perez, M. Labrador and P. Wightman, Location-Based Information Systems. USA. CRC Computer and Information Science Series. CRC Press, 2011.

[2] European GNSS Agency. Opportunities Abound in Growing Location-Based Services Market. In http://www.gsa.europa.eu/go/news/opportunities-abound-in-growinglocation-based-services-market, April 2010.

[3] Directive 2002/58/EC of the European Parliament and of the Council. "European Union Directive on Privacy and Electronic Communications," 2002.

[4] Location Privacy Protection Act, S.1223. 2012

[5] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo and J. Crowcroft, "SpotME if you can: Randomized responses for location obfuscation on mobile phones," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on,* 2011, pp. 363-372.

[6] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati and P. Samarati, "An Obfuscation-Based Approach for Protecting Location Privacy," *Dependable and Secure Computing, IEEE Transactions on,*vol. 8, pp. 13-27, 2011.

[7] A. Khoshgozaran, H. Shirani-Mehr and C. Shahabi, "SPIRAL: A scalable private information retrieval approach to location privacy," in Mobile Data Management Workshops, 2008. MDMW 2008. Ninth International Conference on, pp. 55-62.

[8] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi and K. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data,* pp. 121-132.

[9] H. Lu, C. S. Jensen and M. L. Yiu, "Pad: Privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 16-23.

[10] C. L. Bowen III and T. L. Martin, "A survey of location privacy and an approach for solitary users," in System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, pp. 163c-163c.

[11] P. Wightman, W. Coronell, D. Jabba, M. Jimeno and M. Labrador, "Evaluation of location obfuscation techniques for privacy in location based information systems," in Communications (LATINCOM), 2011 IEEE Latin-American Conference on, pp. 1-6.

[12] H. Kido, Y. Yanagisawa and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on,* pp. 88-97.

[13] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM) on* 1998, *45*(6), 965-981.

[14] Kushilevitz, E.; Ostrovsky, R., "Replication is not needed: single database, computationally-private information retrieval," Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on, pp.364,373, 1997, ACM.

[15] P. Shankar, V. Ganapathy and L. Iftode (2009, September), "Privately querying location-based services with sybilquery". In Proceedings of the 11th international conference on Ubiquitous computing, 2009, pp. 31-40.

[16] F. Olumofin, P. K. Tysowski, I. Goldberg, and U. Hengartner, "Achieving efficient query privacy for location based services," in *Privacy Enhancing Technologies*, 2010, pp. 93–110.

[17] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying Location Privacy," 2011, pp. 247–262.