Privacy Preserving Location-Based Service Protocol

George Antony Jasmine. D
Vel Tech Multi tech  Dr.R.R,Dr.S.R
Engineering College, Avadi, Chennai , India

S.Vinod
Vel Tech Multi tech  Dr.R.R,Dr.S.R
Engineering College, Avadi, Chennai , India

Abstract:

Location-based service (LBS) is booming up in recent years with the rapid growth of mobile devices and the emerging of cloud computing paradigm. Along with the challenges to establish LBS and the user privacy issue becomes the most important concern. So successful privacy-preserving LBS must be secure and provide accurate query results. In this paper we present a solution to one of the location-based query problems. This problems are defined as follows: (i) a user wants to query a database of location data, known as Points Of Interest (POI), and does not want to reveal his/her location to the server due to privacy concerns; (ii) the owner of the location data, that is, the location server, does not want to simply distribute its data to all users. The location server needs to have some control over its data, since the data is its asset. For this work, we propose a private circular query protocol (PCQP) to deal with the privacy and the accuracy issues of privacy-preserving LBS. The protocol includes space filling curve and a public key homomorphic cryptosystem. As a result, the security level of the proposed protocol is close to perfect secrecy without the help of a trusted third party and simulation results show that the -NN query accuracy rate of the proposed protocol is higher than 90% even when is large.

Index Terms—Privacy preserving, space-filling curve, paillier encryption.

1. Introduction:

A location based service (LBS) is an information, entertainment and utility service generally accessible by mobile devices such as, mobile phones, GPS devices, pocket PCs, and operates through a mobile network. A LBS can offer many services to the users based on the geographical position of their mobile device.

The services provided by a LBS are typically based on a point of interest database. By retrieving the Points Of Interest (POIs) from the database server, the user can get answers to various location based queries, which include but are not limited to - discovering the nearest ATM machine, gas station, hospital, or police station. In recent years there has been a dramatic increase in the number of mobile devices querying location servers for information about POIs. Among many challenging barriers to the wide deployment of such application, privacy assurance is a major issue. For instance, users may feel reluctant to disclose their locations to the LBS, because it may be possible for a location server to learn who is making a certain query by linking these locations with a residential phone book database, since users are likely to perform many queries from home. The Location Server (LS), which offers some LBS and spends its resources to compile information about various interesting POIs. So, it is expected that the LS would not disclose any information without fees. Therefore the LBS has to ensure. In the most representative research work [1], the accuracy of -NN search is near to 100% when however, it will drop when increases. Therefore, on the basis of connected space-filling curves and homomorphism cryptosystems, an effective secure -NN search protocol, Private Circular Query Protocol (PCQP), is proposed to deal with the aforementioned two challenges. In PCQP, the Moore’s version of Hilbert curve [2], [3] (or Moore curve in short) is selected as the mapping tool to transform POIs in 2-D space into 1-D space, and the LBS query is resolved in the 1-D transformed space with the proposed secret circular shift scheme. The time-consuming space transformation effort is paid only in the initialization phase for building an LBS. The resultant 2-D to 1-D space transformation can be repeatedly reused. Section II Shows the Related work for this paper, Section III presents and describes our proposed protocol. Section IV analyses the security of the protocol. Section V summarises the key contributions of this paper and future directions.

2. Related Work:

The first solution to the problem was proposed by Beresford [13], in which the privacy of the user is maintained by constantly changing the user’s name or pseudonym within some mix-zone. It
shown that, due to the nature of the data being exchanged between the user and the server, the repeated changing of the user’s name provides little protection for the user’s privacy. The recent investigation of the mix-zone approach has been applied to road networks [25]. They investigated the required number of users to satisfy the unlinkability property when there are repeated queries over an intervals. So it requires careful control of how many users are contained within the mix-zone, which may be difficult to achieve in practice.

A complementary technique to the mix-zone approach is based on k-anonymity [18], [8], [14]. The concept of kanonymity was introduced as a method for preserving privacy when releasing sensitive records [28]. This is achieved by generalisation and suppression algorithms to ensure that a record could not be distinguished from \((k - 1)\) other records. The solution for LBS uses a trusted anonymiser to provide anonymity for the location data, like the location data of a user cannot be distinguished from other users. An enhanced trusted anonymiser approach has also been proposed, which may allows the users to set their level of privacy based on the value of \(k\) [22], [21]. This means that, given the overhead of the anonymiser, small value of \(k\) could be used to increase the efficiency. Conversely, a large value of \(k\) could be chosen to improve the privacy, if the users felt that their position data could be used maliciously. Methods have also been proposed to confuse and distort the location data, which includes path and position confusion. Path confusion was presented by Hoh and Gruteser [18]. The basic idea is to add uncertainty to the location data of the users at the points the paths of the users cross, making it hard to trace users based on raw location data that was kanonymised. Position confusion has also been proposed as an approach to provide privacy [16], [19]. The idea is for the trusted anonymiser to group the users according to a cloaking region (CR), thus making it harder for the LS to identify an individual.

As solutions based on the use of a central anonymiser are not practical. Hashem and Kulik offered a scheme whereby a group of trusted users construct an ad-hoc network and the task of querying the LS is delegated to a single user [19]. This idea improves on the previous work by the fact that there is no single point of failure. If a user that is querying the LS suddenly goes offline, then another candidate can be easily found. However, generating a trusted ad-hoc network in a real world scenario is not always possible.

Another method for avoiding the use of a trusted anonymiser is to use ‘dummy’ locations [17], [16]. The basic idea is to confuse the location of the user by sending many random other locations to the server, such that the server cannot distinguish the actual location from the fake locations. This incurs both processing and communication overhead for the user device. The user has to randomly choose a set of fake locations as well as transmitting them over a network, wasting bandwidth. We refer the interested reader to Krumm [19], for a more detailed survey in this area. Most of the previously discussed issues are solved with the introduction of a private information retrieval (PIR) location based server scheme [11]. Their basic idea is to employ PIR to enable the user to query the location database without compromising the privacy of the query. Generally speaking, PIR schemes allow a user to retrieve data (bit or block) from a database, without disclose the index to the database server [15]. Ghinita et al. used a variant of PIR which is based on the quadratic residuosity problem [20]. Basically the quadratic residuosity problem states that is computationally hard to determine whether a number is a quadratic residue of some composite modulus \(n (x^2 = q (mod n))\), where the factorisation of \(n\) is unknown. This idea was extended to provide database protection [10], [11]. This protocol consists of two stages. In the first stage, the user and server use homomorphic encryption to allow the user to privately determine whether his/her location is contained within a cell, without disclosing their coordinates to the server. In the second stage, PIR is used to retrieve the data contained within the appropriate cell. The homomorphic encryption scheme used to privately compare two integers is the Paillier encryption scheme [24]. The Paillier encryption scheme is known to be additively homomorphic and multiplicatively-by-a-constant homomorphic. This means that we can add or scale numbers even when all numbers are encrypted. Both features are used to determine the sign (most significant bit) of \((a - b)\), and hence the user is able to determine the cell that he/she is located, without disclosing their location.

3. Protocol Summary:

The ultimate goal of our protocol is to obtain a set (block) of POI records from the LS, which are close to the user’s position, without compromising the privacy of the user or the server. We achieve this by applying a two stage approach. The first stage is based on a two dimensional oblivious transfer [4] and the second stage is based on a communicationally efficient PIR [9]. The oblivious transfer based protocol is used by the user to obtain the cell ID, where the user is
located, and the corresponding symmetric key. The knowledge of the cell ID and the symmetric key is then used in the PIR based protocol to obtain and decrypt the location data. The user determines his/her location within a publicly generated grid \( P \) by using his/her GPS coordinates and forms an oblivious transfer query. The minimum dimensions of the public grid are defined by the server and are made available to all users of the system. This public grid superimposes over the privately partitioned grid generated by the location server’s POI records, such that there is at least one \( P_{i,j} \) cell within the server’s partition \( Q_{i,j} \). Since PIR does not require that a user is constrained to obtain only one bit/block, the location server needs to implement some protection for its records. This is achieved by encrypting each record in the POI database with a key using a symmetric key algorithm, where the key for encryption is the same key used for decryption. This key is augmented with the cell info data retrieved by the oblivious transfer query. Hence, even if the user uses PIR to obtain more than one record, the data will be meaningless resulting in improved security for the server’s database.

### 3.1 Oblivious Transfer Based Protocol

The purpose of this protocol is for the user to obtain one and only one record from the cell in the public grid \( P \). We achieve this by constructing a 2-dimensional oblivious transfer, based on the ElGamal oblivious transfer [5], using adaptive oblivious transfer proposed by Naor et al. [4]. The public grid \( P \), known by both parties, has \( m \) columns and \( n \) rows. Each cell in \( P \) contains a symmetric key \( k_{i,j} \) and a cell id in grid \( Q \) i.e., \((IDQ_{i,j}, k_{i,j})\), which can be represented by a stream of bits \( X_{i,j} \). The user determines his/her \( i, j \) coordinates in the public grid which is used to acquire the data from the cell within the grid. The protocol is initialized by the server by generating \( m \times n \) keys of the form \( gR_i || gC_i \).

### 3.2 Private Information Retrieval Based Protocol

With the knowledge about which cells are contained in private grid, and the knowledge of the keys that encrypts the data in the cell, the users can initiate a private information retrieval protocol4 with the location server to acquire the encrypted POI data. Assume the server has initialised the integer \( e \), the user \( ui \) and \( LS \) can engage in the following private information retrieval protocol using \( IDQ_{i,j} \), obtained from the execution of the previous protocol, as input. The \( IDQ_{i,j} \) allows the user to choose the associated prime number power \( \pi_i \), which in turn allows the user to query the server.

### 3.3 Private circular query protocol.

![Fig. 1. Private circular query protocol.](image)

The whole architecture of PCQP is illustrated in Fig. 1, which belongs to 2-tier LBS architecture category, and the whole protocol consists of the following 6 steps:

**Initialization Processes**

step-1: Server constructs a Moore curve and generates H-indexes for all registered POIs on the target map. In this initial step, LBS-server constructs an appropriate Moore curve covering up the target map and builds a table containing all the information related to all POIs. Each stored POI in the table (called H-table, thereafter) is numbered according to an evenly distributed H-index with common difference instead of the originally associated H-value. The mapping from H-values to H-indexes is stored in a lookup-table on the server side and the formal definition of H-index. Since the generation of H-indexes needs to be done only once in the server side, the required efforts can be treated as a part of the service setup costs.
step-2: Server publicly announces the lookup-table and Moore curve’s setting parameters to the registered users so that a querying user can compute the H-index of his/her current location on the map. The lookup-table and the Moore curve’s setting parameters could be made publicly to every registered user, and this public announcement won’t affect the privacy issue. Of course, a registered user can download this information just from LBS-Website.

step-3: User decides his/her public and private key pairs of the Paillier cryptosystem and sends the public-key to server.

Query Processes

step-4: User issues a -NN query to server.
(a) User choose an -offset circular shift permutation matrix , where denotes the number of POIs in H-table and the -th element of the first row of is the only nonzero element in that row.
(b) User adds a number to the H-index of his/her current location to generate a shifted-H-index.
(c) Users send the shifted-H-index and the encrypted first row of by the public-key selected in step-3, denoted as , to server and issues a -NN query.

step-5: Server performs a secret circular shift, which is defined by , on the POI-info column of H-table based on the Additive and Multiplicative homomorphisms of Paillier cryptosystem with user’s public key. Server, then, conducts a -NN search upon the circularly shifted H-table, and returns the encrypted search results back to user.

step-6: User uses his/her private key (selected in step-3) to decrypt the received results and finds the required -NN solutions. Since H-index of the querying user and the POI-info column of H-table have respectively been added by (in step-4) and secretly circularly shifted by (in step-5), where , the secret -NN search results done in the shifted H-table may be the same as that of the original -NN search results done in their plaintext version.

The benefits of the PCQP protocol are listed in the following points:

a) Resistance to Correlation Attack and Background Knowledge Attack.

The proposed secret circular shift is performed before each query and the amount of shift is determined only by the querying user, it can be regarded as an one-time pad encryption method, and therefore, providing high security. Servers cannot infer any knowledge about the user’s location from the query history and the user’s profiles, since the amount of shift has been scrambled by user and the POI information has also been encrypted. Under that circumstance, the Correlation Attacks and Background Knowledge Attack made by the server can not succeed.

b) Supporting multiuser scenario.

The public-key characteristics of Paillier cryptosystem, one of the key components of our protocol, can easily adapt to the multiuser environment. The proposed protocol only requires users to keep their private keys on the client side and send the corresponding public keys to the server side, which decouples the relation among users. The key management issue of newly joined user can be intuitively solved by the properties of the adopted public-key cryptosystem.

c) Providing high accuracy -NN search results.

In general, the security challenge and accuracy challenge cannot be jointly addressed. There are lots of work which can obtain accurate -NN results but are vulnerable to the Correlation Attack [6]–[10]. On the other hand, there are research works providing high security whereas the accuracy of -NN results drops largely when [11]. Noticeably, the proposed protocol, with some simple modifications, can achieve high accuracy rate for -NN search (larger than 90% even if is large) without compromising the robustness of security.

4. Security Analysis

In this section, we analyse the security of the user and LS. While the user does not want to give up the privacy of his/her location, the server does not want to simply transfer all records to the user. This would not make much business sense in a variety of applications.

A. User’s security

Fundamentally, the user does not want to disclose the cell which contains his/her location to the server. Two assumptions must be maintained in order to effectively render location private. The server must not be able to determine which cell the user is querying in the oblivious transfer protocol, and the server must not be able to determine which cell the
user is querying in the private information retrieval protocol. The oblivious transfer assumption is based on the discrete logarithm assumption. This essentially means that given $g^x \pmod p$, where $p$ is a large prime and $g$ is a generator of some cyclic group, it is computationally infeasible to determine $x$. In our case, if the user supplies $(g_{r_1}, g^{−jy_1})$ and $(g_{r_2}, g^{−jy_2})$ to the server, then the server is unable to determine $i$ and $j$. If the discrete logarithm assumption holds, then we claim this is secure.

B. Server’s security

The server’s security is based on keeping the boundaries of its records private. Since disclosing this information may enable the user to infer more information about the database than he/she is allowed. In our solution this information is protected by the oblivious transfer protocol. The user is forced to retrieve one and only one record from the public grid $Pi,j$. This is because of the two sets of random values $r_a$ and $r_\beta$. Only when $i = a$ and $j = \beta$, the user can decrypt to find $g_{r_i}$ and $g_{Ci}$. All other times, the result will be indistinguishable from random. Under the discrete logarithm problem assumption, it is computationally intractable to determine any exponent from the ciphertext. Hence, the user is only able to determine one and only one result.

4. Complexity Analysis of PCQP:

As we conduct PCQP with a homomorphic cryptosystem, the following discussion of complexity will base on the homomorphic cryptosystem we adopted.

About the generation on the user side, instead of computing ciphers of by direct Paillier encryption which introduces unacceptable overhead, we reduce the overhead by utilizing the addition homomorphism, i.e., (1), of Paillier cryptosystem during the generation of ciphers of which can be illustrated as follows(13) where , but As a result, the total computation time for generating the of 62555 POIs only requires about 1.4 seconds. And the computation time for decryption is less than 0.3 seconds per result. This implies that the required computational complexity in user side is quite lightweight in PCQP. Therefore, the performance bottleneck of PCQP comes from the secret circular shifting performed on the server side. For each -NN search, the server will multiply rows of the encrypted circular shift matrix by the data of POI-info column, ones around the corresponding row position of querying H-index. The ordinary matrix multiplication requires multiplication and additions in plaintext which corresponds to exponentials by the homomorphism in Paillier cryptosystem. On average, the server requires seconds for completing a -NN search in our settings. Therefore, the required computation time can be speeded up to seconds by using an 8 core parallel computing on the server side. Notice that, the Paillier domain matrix-vector multiplication of each row is independent of each other and can, therefore, be computed in parallel. This fact implies that the computation required in the server side is rather suitable to be implemented in a cloud computing environments. From providing an LBS points of view, the corresponding computational costs for achieving privacy preservation should be handled by the LBS server, which would have high computing power to make LBS success.

Conclusion

In this paper we have presented a location based query solution that employs two protocols that enables a user to privately determine and acquire location data. The first step is used for a user to privately determine his/her location using oblivious transfer on a public grid. The second step involved a private information retrieval interaction that retrieves the record with high communication efficiency. We analysed the performance of our protocol and found it to be both computationally and communicationally more efficient than the other existing solutions.

Future work will involve porting the software to a mobile device to test the actual feasibility of our proposed protocol. Additionally, the problem concerning the LS supplying misleading data to the client is also interesting. Privacy preserving reputation techniques seem a suitable approach to address such problem. A possible solution could integrate methods from [12]. Once suitable strong solutions exist for the general case, they can be easily integrated into our approach.

References:


