5-13-2014

Privacy-Preserving and Content-Protecting Location Based Queries

Russell Paulet
Victoria University, Melbourne, Xun.Yi@vu.edu.au

Mohammed Golam Kaosar
Victoria University, Melbourne

Xun Yi
Victoria University, Xun.Yi@vu.edu.au

Elisa Bertino
Purdue University, bertino@cs.purdue.edu

Paulet, Russell; Kaosar, Mohammed Golam; Yi, Xun; and Bertino, Elisa, "Privacy-Preserving and Content-Protecting Location Based Queries" (2014). *Cyber Center Publications.* Paper 653.
http://dx.doi.org/10.1109/TKDE.2013.87

Privacy-Preserving and Content-Protecting Location Based Queries

Russell Paulet, Md. Golam Kaosar, Xun Yi, and Elisa Bertino, *Fellow, IEEE*

Abstract—In this paper we present a solution to one of the location-based query problems. This problem is defined as follows: (i) a user wants to query a database of location data, known as Points Of Interest (POIs), 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 desires to have some control over its data, since the data is its asset. We propose a major enhancement upon previous solutions by introducing a two stage approach, where the first step is based on Oblivious Transfer and the second step is based on Private Information Retrieval, to achieve a secure solution for both parties. The solution we present is efficient and practical in many scenarios. We implement our solution on a desktop machine and a mobile device to assess the efficiency of our protocol. We also introduce a security model and analyse the security in the context of our protocol. Finally, we highlight a security weakness of our previous work and present a solution to overcome it.

Index Terms—Location based query, private query, private information retrieval, oblivious transfer

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 operating 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, spends its resources to compile information about various interesting POIs. Hence, it is expected that the LS would not disclose any information without fees. Therefore the LBS has to ensure that LS's data is not accessed by any unauthorized user. During the process of transmission the users should not be allowed to discover any information for which they have not paid. It is thus crucial that solutions be devised that address the privacy of the users issuing queries, but also prevent users from accessing content to which they do not have authorization.

- *R. Paulet, Md. G. Kaosar, and X. Yi are with the School of Engineering and Science, Victoria University, Melbourne, VIC 8001, Australia. E-mail: russell.paulet@live.vu.edu.au; glmksr@live.com; xun.yi@vu.edu.au.*
- *E. Bertino is with the Department of Computer Science and Cyber Center, Purdue University, West Lafayette, IN 47907 USA. E-mail: bertino@cs.purdue.edu.*

Manuscript received 12 June 2012; revised 14 Nov. 2012; accepted 14 Feb. 2013. Date of publication 26 May 2013; date of current version 7 May 2014. Recommended for acceptance by E. Ferrari.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.87

1.1 Related Work

The first solution to the problem was proposed by Beresford [3], in which the privacy of the user is maintained by constantly changing the user's name or pseudonym within some mix-zone. It can be shown that, due to the nature of the data being exchanged between the user and the server, the frequent changing of the user's name provides little protection for the user's privacy. A more recent investigation of the mix-zone approach has been applied to road networks [29]. They investigated the required number of users to satisfy the unlinkability property when there are repeated queries over an interval. This requires careful control of how many users are contained within the mix-zone, which is difficult to achieve in practice.

A complementary technique to the mix-zone approach is based on k-anonymity [4], [10], [16]. The concept of k-anonymity was introduced as a method for preserving privacy when releasing sensitive records [32]. This is achieved by generalisation and suppression algorithms to ensure that a record could not be distinguished from $(k-1)$ other records. The solutions for LBS use a trusted anonymiser to provide anonymity for the location data, such that the location data of a user cannot be distinguished from $(k-1)$ other users.

An enhanced trusted anonymiser approach has also been proposed, which allows the users to set their level of privacy based on the value of k [24], [25]. This means that,

given the overhead of the anonymiser, a 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. Choosing a value for k, however, seems unnatural. There have been efforts to make the process less artificial by adding the concept of feeling-based privacy [23], [33]. Instead of specifying a k, they propose that the user specifies a cloaking region that they feel will protect their privacy, and the system sets the number of cells for the region based on the popularity of the area. The popularity is computed by using historical footprint database that the server collected.

New privacy metrics have been proposed that captures the users' privacy with respect to LBSs [5]. The authors begin by analysing the shortcomings of simple k-anonymity in the context of location queries. Next, they propose privacy metrics that enables the users to specify values that better match their query privacy requirements. From these privacy metrics they also propose spatial generalisation algorithms that coincide with the user's privacy requirements.

Methods have also been proposed to confuse and distort the location data, which include 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 k-anonymised. Position confusion has also been proposed as an approach to provide privacy [19], [25]. 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. A common problem with general CR techniques is that there may exist some semantic information about the geography of a location that gives away the user's location. For example, it would not make sense for a user to be on the water without some kind of boat. Also, different people may find certain places sensitive. Damiani *et al.* have presented a framework that consists of a obfuscation engine that takes a users profile, which contains places that the user deems sensitive, and outputs obfuscated locations based on aggregating algorithms [7].

As solutions based on the use of a central anonymiser are not practical, Hashem and Kulik presented 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 [17]. 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 [8], [20]. 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 [21], 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 scheme [14]. The 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 disclosing the index of the data to be retrieved to the database server [6]. Ghinita *et al.* used a variant of PIR which is based on the quadratic residuosity problem [22]. 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 [12], [13]. 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 his/her 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 [27]. 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 in which he/she is located, without disclosing his/her location.

1.2 Our Contributions

In this paper, we propose a novel protocol for location based queries that has major performance improvements with respect to the approach by Ghinita at el. [12] and [13]. Like such protocol, our protocol is organized according to two stages. In the first stage, the user privately determines his/her location within a public grid, using oblivious transfer. This data contains both the *ID* and associated symmetric key for the block of data in the private grid. In the second stage, the user executes a communicational efficient PIR [11], to retrieve the appropriate block in the private grid. This block is decrypted using the symmetric key obtained in the previous stage.

Our protocol thus provides protection for both the user and the server. The user is protected because the server is unable to determine his/her location. Similarly, the server's data is protected since a malicious user can only decrypt the block of data obtained by PIR with the encryption key acquired in the previous stage. In other words, users cannot gain any more data than what they have paid for. We remark that this paper is an enhancement of a previous work [28]. In particular, the following contributions are made.

1) Redesigned the key structure
2) Added a formal security model
3) Implemented the solution on both a mobile device and desktop machine

As with our previous work, the implementation demonstrates the efficiency and practicality of our approach.

1.3 Paper Organization

The rest of the paper is organised as follows. Section 2 presents the protocol model and other preliminaries. Section 3 presents and describes our proposed protocol. Section 4 analyses the security of the protocol. Section 5 analyses the performance and efficiency of the protocol. Section 6 reports the performance results of a working prototype using two platforms: a desktop and a mobile, and discusses feasibility. Section 7 summarises the key contributions of this paper and future directions.

2 Protocol Model

Before describing our protocol we introduce the system model, which defines the major entities and their roles. The description of the protocol model begins with the notations and system parameters of our solution.

2.1 Notations

Let $x \leftarrow y$ be the assignment of the value of variable y to variable x and $E \Leftarrow v$ be the transfer of the variable v to entity E. Denote the ElGamal [9] encryption of message m as $E(m, y) = \boldsymbol{A} = (A_1, A_2) = (g^r, g^m y^r)$, where g is a generator of group G, y is the public key of the form $y = g^x$, and r is chosen at random. This will be used as a basis for constructing an adaptive oblivious transfer scheme [26]. Note that $\boldsymbol{A}$ is a vector, while A_1, A_2 are elements of the vector. The cyclic group G_0 is a multiplicative subgroup of the finite field F_p, where p is a large prime number and q is a prime that divides $(p-1)$. Let g_0 be a generator of group G_0, with order q. Let G_1 be a multiplicative subgroup of finite field F_q, with distinct generators g_1 and g_2 where both have prime order $q' | (q-1)$. Based on this definition, groups G_0 and G_1 can then be linked together and have the form $g_0^{g_1^x g_2^y}$, where x and y are variable integers. This will be used in our application to generate an ElGamal cryptosystem instance in group G_1. We denote $|p|$ to be the bit length of p, $\oplus$ to be the exclusive OR operator, $a||b$ to be the concatenation of a and b, and $\langle |g| \rangle$ to be the order of generator g.

We require for security reasons, that $|q'| = 1024$ and p has the form $p = 2q' + 1$. We also require that the parameters $G_0, g_0, G_1, g_1, g_2, p, q'$ be fixed for the duration of a round of our protocol and be made publicly accessible to every entity in our protocol.

2.2 System Model

The system model consists of three types of entities (see Fig. 1): the set of users[1] who wish to access location data U, a mobile service provider SP, and a location server LS. From the point of view of a user, the SP and LS will compose a server, which will serve both functions. The user does not need to be concerned with the specifics of the communication.

The users in our model use some location-based service provided by the location server LS. For example, what is the nearest ATM or restaurant? The purpose of the mobile service provider SP is to establish and maintain the communication between the location server and the user. The location server LS owns a set of POI records r_i for $1 \leq r_i \leq \rho$. Each record describes a POI, giving GPS coordinates to its location (x_{gps}, y_{gps}), and a description or name about what is at the location.

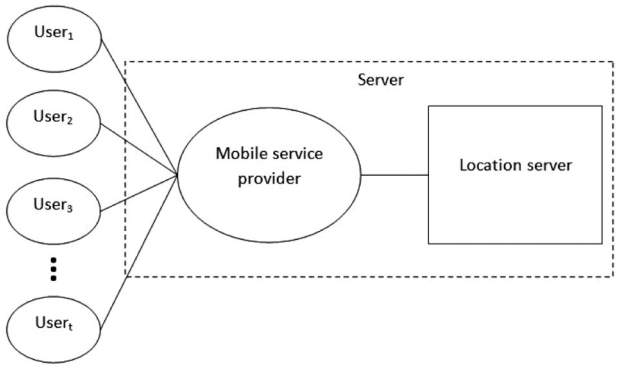

Fig. 1. System model.

We reasonably assume that the mobile service provider SP is a passive entity and is not allowed to collude with the LS. We make this assumption because the SP can determine the whereabouts of a mobile device, which, if allowed to collude with the LS, completely subverts any method for privacy. There is simply no technological method for preventing this attack. As a consequence of this assumption, the user is able to either use GPS (Global Positioning System) or the mobile service provider to acquire his/her coordinates.

Since we are assuming that the mobile service provider SP is trusted to maintain the connection, we consider only two possible adversaries. One for each communication direction. We consider the case in which the user is the adversary and tries to obtain more than he/she is allowed. Next we consider the case in which the location server LS is the adversary, and tries to uniquely associate a user with a grid coordinate.

2.3 Security Model

Before we define the security of our protocol, we introduce the concept of k out of N adaptive oblivious transfer as follows.

Definition 1 (k out of N adaptive oblivious transfer ($OT^N_{k\times 1}$) [26]). *$OT^N_{k\times 1}$ protocols contain two phases, for initialization and for transfer. The initialization phase is run by the sender (Bob) who owns the N data elements $X_1, X_2, \ldots, X_N$. Bob typically computes a commitment to each of the N data elements, with a total overhead of O(N). He then sends the commitments to the receiver (Alice). The transfer phase is used to transmit a single data element to Alice. At the beginning of each transfer Alice has an input I, and her output at the end of the phase should be data element X_I. An $OT^N_{k\times 1}$ protocol supports up to k successive transfer phases.*

Built on the above definition, our protocol is composed of initialisation phase and transfer phase. We will now outline the steps required for the phases and then we will formally define the security of these phases.

1. In this paper we use the term "user" to refer to the entity issuing queries and retrieving query results. In most cases, such user is a client software executing on behalf of a human user.

Our initialisation phase is run by the sender (server), who owns a database of location data records and a 2-dimensional key matrix $\mathcal{K}_{m\times n}$, where m and n are rows and columns respectfully. An element in the key matrix is referenced as $k_{i,j}$. Each $k_{i,j}$ in the key matrix uniquely encrypts one record. A set of prime powers $\mathcal{S} = \{p_1^{c_1}, \ldots, p_N^{c_N}\}$, where N is the number of blocks, is available to the public. Each element in $\mathcal{S}$ the p_i is a prime and c_i is a small natural number such that $p_i^{c_i}$ is greater than the block size (where each block contains a number of POI records). We require, for convenience that the elements of $\mathcal{S}$ follow a predictable pattern. In addition, the server sets up a common security parameter k for the system.

Our transfer phase is constructed using six algorithms: QG1,RG1,RR1,QG2,RG2,RR2. The first three compose the first phase (Oblivious Transfer Phase), while the last three compose the second phase (Private Information Retrieval Phase). The following six algorithms are executed sequentially and are formally described as follows.

Oblivious Transfer Phase

1) *QueryGeneration*$_1$ (*Client*) (QG1):
Takes as input indices i, j, and the dimensions of the key matrix m, n, and outputs a query $\mathcal{Q}_1$ and secret s_1, denoted as $(\mathcal{Q}_1, s_1) = QG_1(i, j, m, n)$.
2) *ResponseGeneration*$_1$(*Server*) (RG1):
Takes as input the key matrix $\mathcal{K}_{m\times n}$, and the query $\mathcal{Q}_1$, and outputs a response $\mathcal{R}_1$, denoted as $(\mathcal{R}_1) = RG_1(\mathcal{K}_{m\times n}, \mathcal{Q}_1)$.
3) *ResponseRetrieval*$_1$ (*Client*) (RR1):
Takes as input indices i, j, the dimensions of the key matrix m, n, the query $\mathcal{Q}_1$ and the secret s_1, and the response $\mathcal{R}_1$, and outputs a cell-key $k_{i,j}$ and cell-id $ID_{i,j}$, denoted as $(k_{i,j}, ID_{i,j}) = RR_1(i, j, m, n, (Q_1, s_1), \mathcal{R}_1)$.

Private Information Retrieval Phase

4) *QueryGeneration*$_2$ (*Client*) (QG2):
Takes as input the cell-id $ID_{i,j}$, and the set of prime powers $\mathcal{S}$, and outputs a query $\mathcal{Q}_2$ and secret s_2, denoted as $(\mathcal{Q}_2, s_2) = QG_2(ID_{i,j}, \mathcal{S})$.
5) *ResponseGeneration*$_2$ (*Server*) (RG2):
Takes as input the database D, the query $\mathcal{Q}_2$, and the set of prime powers $\mathcal{S}$, and outputs a response $\mathcal{R}_2$, denoted as $(\mathcal{R}_2) = RG_2(D, \mathcal{Q}_2, \mathcal{S})$.
6) *ResponseRetrieval*$_2$ (*Client*) (RR2):
Takes as input the cell-key $k_{i,j}$ and cell-id $ID_{i,j}$, the query $\mathcal{Q}_2$ and secret s_2, the response $\mathcal{R}_2$, and outputs the data d, denoted as $(d) = RR_2(k_{i,j}, ID_{i,j}, (\mathcal{Q}_2, s_2), \mathcal{R}_2)$.

Our transfer phase can be repeatedly used to retrieve points of interest from the location database.

With these functions described, we can build security definitions for both the client and server [11], [26].

Definition 2 (Client's Security (Indistinguishability) [26]). *In a $OT^N_{k\times 1}$ protocol, for any step $1 \le t \le k$, for any previous items $I_1, \ldots, I_{t-1}$ that the receiver has obtained in the first t-1 transfers, for any $1 \le I_t, I'_t \le N$ and for any probabilistic polynomial time $\mathcal{B}'$ executing the server's part, the views that $\mathcal{B}'$ sees in case the client tries to obtain X_{I_t} and in the case the client tries to obtain $X_{I'_t}$ are computationally indistinguishable given $X_1, X_2, \ldots, X_N$.*

Definition 3 (Server's Security (Comparison with Ideal Model) [26]). *We compare a $OT^N_{k\times 1}$ protocol to the ideal implementation, using a trusted third party that gets the server's input $X_1, X_2, \ldots, X_N$ and the client's requests and gives the client the data elements she has requested. For every probabilistic polynomial-time machine $\mathcal{A}'$ substituting the client, there exists a probabilistic polynomial-time machine $\mathcal{A}''$ that plays the receiver's role in the ideal model such that the outputs of $\mathcal{A}'$ and $\mathcal{A}''$ are computationally indistinguishable.*

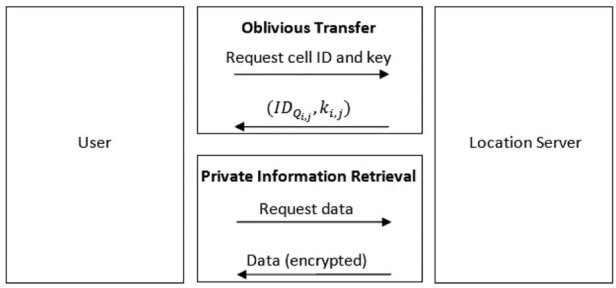

Fig. 2. High level overview of the protocol.

3 Protocol Description

We now describe our protocol. We first give a protocol summary to contextualise the proposed solution and then describe the solution's protocol in more detail.

3.1 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 data stored at the server. We achieve this by applying a two stage approach shown in Fig. 2. The first stage is based on a two-dimensional oblivious transfer [26] and the second stage is based on a communicationally efficient PIR [11]. 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[2]. 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 for each cell $Q_{i,j}$ in the server's partition there is at least one $P_{i,j}$ cell from the public grid. This is illustrated in Fig. 3.

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

2. An oblivious transfer query is such that a server cannot learn the user's query, while the user cannot gain more than they are entitled. This is similar to PIR, but oblivious transfer requires protection for the user and server. PIR only requires that the user is protected.

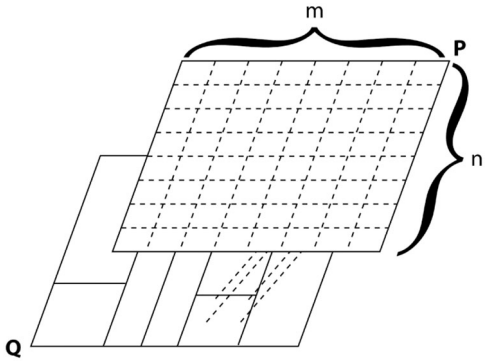

Fig. 3. Public grid superimposed over the private grid.

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. Before we describe the protocol in detail, we describe some initialisation performed by both parties.

3.2 Initialization

A user u from the set of users U initiates the protocol process by deciding a suitable square cloaking region CR, which contains his/her location. All user queries will be with respect to this cloaking region. The user also decides on the accuracy of this cloaking region by how many cells are contained within it, whose size cannot be smaller than the minimum size defined by the location server. which is at least the minimum size defined by the server. This information is combined with the dimensions of the CR to form the public grid P and submitted to the location server, which partitions its records or superimposes it over pre-partitioned records (see Fig. 3). This partition is denoted Q (note that the cells don't necessarily need to be the same size as the cells of P). Each cell in the partition Q must have the same number r_{max} of POI records. Any variation in this number could lead to the server identifying the user. If this constraint cannot be satisfied, then dummy records can be used to make sure each cell has the same amount of data. We assume that the *LS* does not populate the private grid with misleading or incorrect data, since such action would result in the loss of business under a payment model.

Next, the server encrypts each record r_i within each cell of Q, $Q_{i,j}$, with an associated symmetric key $k_{i,j}$. The encryption keys are stored in a small (virtual) database table that associates each cell in the public grid P, $P_{i,j}$, with both a cell in the private grid $Q_{i,j}$ and corresponding symmetric key $k_{i,j}$. This is shown by Fig. 4.

The server then processes the encrypted records within each cell $Q_{i,j}$ such that the user can use an efficient PIR [11], to query the records. Using the private partition Q, the server represents each associated (encrypted) data as an integer C_i, with respect to the cloaking region. For each C_i, the server chooses a set of unique prime powers $\pi_i = p_i^{c_i}$, such that $C_i < \pi_i$. We note that the c_i in the exponent must be small for the protocol to work efficiently. Finally, the server uses the Chinese Remainder Theorem to find the smallest integer e such that $e = C_i \ (mod\ \pi_i)$ for all C_i. The integer e effectively represents the database. Once the initialisation is complete, the user can proceed to query the location server for POI records.

Algorithm 1 *Initialisation*

Input: $X_{1,1}, ..., X_{m,n}$, where $X_{i,j} = ID_{Q_{i,j}} || k_{i,j}$
Output: $Y_{1,1}, ..., Y_{m,n}$

1: $K_{i,j} \leftarrow K_{i,j} = g_0^{g_1^{R_i} g_2^{C_j}}$, for $1 \le i \le n$ and $1 \le j \le m$, where R_i and C_j are randomly chosen
2: $Y_{i,j} \leftarrow X_{i,j} \oplus H(K_{i,j})$, for $1 \le i \le n$ and $1 \le j \le m$, where H is a fast secure hash function
3: **return** $Y_{1,1}, ..., Y_{m,n}$ {Encryptions of $X_{1,1}, ..., X_{m,n}$ using $K_{i,j}$}

3.3 Oblivious Transfer Phase

The purpose of this protocol is for the user to obtain one and only one record from the cell in the public grid P, shown in Fig. 4. We achieve this by constructing a 2-dimensional oblivious transfer, based on the ElGamal oblivious transfer [2], using adaptive oblivious transfer proposed by Naor *et al.* [26].

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 or $(ID_{Q_{i,j}}, k_{i,j})$, 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 initialised by the server by generating $m \times n$ keys of the form $g_0^{g_1^{R_i} g_2^{C_j}}$. We remark that this key structure of this form is an enhancement from [28], as the client doesn't have access to the individual components of the key. This initialisation is presented in Algorithm 1.

Algorithm 1 is executed once and the output $Y_{1,1}, \ldots, Y_{m,n}$ is sent to the user. At which point, the user can query this information using the indices i, and j, as input. This protocol is presented in Algorithm 2.

At the conclusion of the protocol presented by Algorithm 2, the user has the information to query the location server for the associated block.

Theorem 1 (Correctness). *Assume that the user and server follow Algorithms 1 and 2 correctly. Let $X_{i,j}$ be the bit string encoding the pair $(ID_{Q_{i,j}}, k_{i,j})$ and let $X'_{i,j}$ the bit string generated by Algorithm 2 (Step 19) as $X'_{i,j} = Y_{i,j} \oplus H(K_{i,j})$. Then $X'_{i,j} = X_{i,j}$.*

Proof. We begin this proof by showing that $K_{i,j} = K'_{i,j}$, where $K'_{i,j}$ is the key obtained by the user according to the Algorithm 2 (step 18). In the initialisation algorithm (1) $K_{i,j}$ is calculated as $K_{i,j} = g_0^{g_1^{R_i} g_2^{C_j}}$. At the end of the transfer protocol, the user computes $K'_{i,j}$ as $\gamma^{W_3 W_4}$, where W_3 can be simplified as follows when $i = \alpha$.

$$\begin{aligned} W_3 &= V_{1,i} W_1 \\ &= g_1^{R_i} r_R (g_1^{\alpha} g_1^{-i} y_1^{r_1})^{r'_1} U_{1,i}^{-x_1} \\ &= g_1^{R_i} r_R (y_1^{r_1 r'_1}) U_{1,i}^{-x_1} \end{aligned}$$

Algorithm 2 $Transfer$

Input: **User:**i, j
Output: **User:**$(ID_{Q_{i,j}}, k_{i,j})$

1: **User** (*QG1*)
2: $y_1 \leftarrow g_1^{x_1}$, where y_1 is the public key for the row and x_1 is chosen at random
3: $y_2 \leftarrow g_2^{x_2}$, where y_2 is the public key for the column and x_2 is chosen at random
4: $\boldsymbol{C_1} \leftarrow (A_1, B_1) = (g_1^{r_1}, g_1^{-i} y_1^{r_1})$
5: $\boldsymbol{C_2} \leftarrow (A_2, B_2) = (g_2^{r_2}, g_2^{-j} y_2^{r_2})$
6: $Server \Leftarrow \boldsymbol{C_1}, \boldsymbol{C_2}$
7: **Server** (*RG1*)
8: $\boldsymbol{C'_{1,\alpha}} \leftarrow (A_1^{r'_\alpha}, g_1^{R_\alpha} r_R (g_1^\alpha B_1)^{r'_\alpha})$ for $1 \leq \alpha \leq n$ and $r_R = g_1^s$, where s is chosen randomly
9: $\boldsymbol{C'_{2,\beta}} \leftarrow (A_2^{r'_\beta}, g_2^{C_\beta} r_C (g_2^\beta B_2)^{r'_\beta})$ for $1 \leq \beta \leq m$ and $r_C = g_2^t$, where t is chosen randomly
10: $\gamma \leftarrow g_0^{1/r_R r_C}$
11: $User \Leftarrow \boldsymbol{C'_{1,1}}, ..., \boldsymbol{C'_{1,n}}, \boldsymbol{C'_{2,1}}, ..., \boldsymbol{C'_{2,m}}, \gamma$
12: **User** (*RR1*)
13: Let $(U_{1,i}, V_{1,i}) = \boldsymbol{C'_{1,i}}$ and $(U_{2,j}, V_{2,j}) = \boldsymbol{C'_{1,j}}$
14: $W_1 \leftarrow U_{1,i}^{-x_1}$
15: $W_2 \leftarrow U_{2,j}^{-x_2}$
16: $W_3 \leftarrow V_{1,i} W_1$
17: $W_4 \leftarrow V_{2,j} W_2$
18: $K'_{i,j} \leftarrow \gamma^{W_3 W_4}$
19: $X'_{i,j} \leftarrow Y_{i,j} \oplus H(K'_{i,j})$
20: Reconstruct $(ID_{Q_{i,j}}, k_{i,j})$ from $X'_{i,j}$
21: **return** $(ID_{Q_{i,j}}, k_{i,j})$ {Cell id of grid Q, with associated cell key}

$$
\begin{aligned}
&= g_1^{R_i} r_R (g_1^{x r_1 r'_1})(g_1^{r_1 r'_1})^{-x_1} \\
&= g_1^{R_i} r_R (g_1^{x r_1 r'_1})(g_1^{-(x r_1 r'_1)}) \\
&= g_1^{R_i} r_R \ (mod\ q)
\end{aligned}
$$

By similar means we can show that $W_4 = g_2^{C_j} r_C$, when $j = \beta$. So we have the following.

$$
\begin{aligned}
\gamma^{W_3 W_4} &= (g_0^{1/r_R r_C})^{g_1^{R_i} r_R g_2^{C_j} r_C} \\
&= g_0^{g_1^{R_i} g_2^{C_j}} \ (mod\ p)
\end{aligned}
$$

Algorithm 3 $PIRProtocol$

Input: **User:**$ID_{Q_{i,j}}$
Output: **User:**C_i

1: **User** (*QG2*)
2: $\pi_0 \leftarrow \pi_i$, where π_i is chosen based on the value of $ID_{Q_{i,j}}$
3: Generate random group G and group element g, such that π_0 divides the order of g
4: $q \leftarrow |\langle g \rangle| / \pi_0$
5: $h \leftarrow g^q$
6: $Server \Leftarrow G, g$
7: **Server** (*RG2*)
8: $g_e \leftarrow g^e$
9: $User \Leftarrow g_e$
10: **User** (*RR2*)
11: $h_e \leftarrow g_e^q$
12: $C_i \leftarrow log_h h_e$, where log_h is the discrete log base h
13: **return** C_i {The requested (encrypted) data}

This proves $K_{i,j} = K'_{i,j}$. Since $\oplus$ is self inverse and given that $Y_{i,j} = X_{i,j} \oplus H(K_{i,j})$, it follows that $X_{i,j} = Y_{i,j} \oplus H(K_{i,j})$. Using knowledge of $K'_{i,j}$, the user can compute $X_{i,j}$, which is the same as $X'_{i,j}$ as desired. This completes the proof. □

3.4 Private Information Retrieval Phase

With the knowledge about which cells are contained in the private grid, and the knowledge of the key that encrypts the data in the cell, the user can initiate a private information retrieval protocol with the location server to acquire the encrypted POI data. Assuming the server has initialised the integer *e*, the user u_i and *LS* can engage in the following private information retrieval protocol using the $ID_{Q_{i,j}}$, obtained from the execution of the previous protocol, as input. The $ID_{Q_{i,j}}$ allows the user to choose the associated prime number power π_i, which in turn allows the user to query the server. The protocol is presented in Algorithm 3.

Theorem 2. *[(Correctness)] Assume that the user and the server follow the protocol correctly, then the user successfully acquires C_i for his/her chosen prime index.*

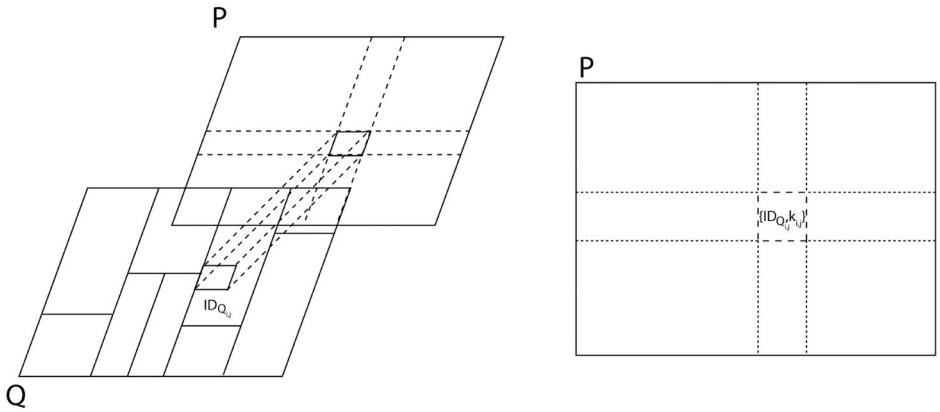

Fig. 4. Association between the public and private grids.

Proof. It is easy to see that $C_i = e \ (mod\ \pi_i)$ and $h_e = g_e^{|\langle g\rangle|/\pi_i}$. Then C_i is the discrete logarithm of h_e to the base h, since $g_e^{|\langle g\rangle|/\pi_i} = g^{e|\langle g\rangle|/\pi_i} = g^{e_{\pi_i}|\langle g\rangle|/\pi_i} = h^{e_{\pi_i}}$, where e_{π_i} stands for $e \ (mod\ \pi_i)$. This completes the proof. □

At the conclusion of the protocol, the user has successfully acquired the block that contain the encrypted POI records. With the knowledge of the cell key $k_{i,j}$, the user can decrypt C_i and obtain the requested data, thus concluding one round of the protocol. Using the same set-up, the user can execute several more rounds very efficiently and effectively without compromising his/her privacy. Similarly, the server's data remains protected based on the fact the user can only acquire one key per round. The security is analysed in more detail next.

4 Security Analysis

In this section, we analyse the security of the client and the server. While the client does not want to give up the privacy of his/her location, the server does not want to disclose other records to the client. This would not make much business sense in a variety of applications. Our analysis will be with respect to the security definitions in Section 2.3.

4.1 Client's Security

Fundamentally, the information that is most valuable to the user is his/her location. This location is mapped to a cell $P_{i,j}$. In both phases of our protocol, the oblivious transfer based protocol and the private information retrieval based protocol, the server must not be able to distinguish two queries of the client from each other. We will now describe both cases separately.

In the oblivious transfer phase, each coordinate of the location is encrypted by the ElGamal encryption scheme, e.g., $(g_1^{r_1}, g_1^{-i} y_1^{r_1})$. It has been shown that ElGamal encryption scheme is semantically secure [9]. This means that given the encryption of one of two plaintexts m_1 and m_2 chosen by a challenger, the challenger cannot determine which plaintext is encrypted, with probability significantly greater than 1/2 (the success rate of random guessing). In view of it, the server cannot distinguish any two queries of the client from each other in this phase.

In the private information retrieval phase, the security of the client is built on the Gentry-Ramzan private information retrieval protocol, which is based on the phi-hiding (ϕ-hiding) assumption [11].

On the basis of the above security analysis, we can conclude with the following theorem.

Theorem 3. *Assume that the ElGamal encryption scheme is semantically secure and the Gentry-Ramzan PIR has client security, our protocol has client security, i.e., the server cannot distinguish any two queries of the client from each other.*

4.2 Server's Security

Intuitively, the server's security requires that the client can retrieve one record only in each query to the server, and the server must not disclose other records to the client in the response. Our protocol achieves the server's security in the oblivious transfer phase, which is built on the Naor-Pinkas oblivious transfer protocol [26].

Our Algorithm 1 is the same as the Naor-Pinkas oblivious transfer protocol except from the one-out-of-n oblivious transfer protocol, which is built on the ElGamal encryption scheme. In the generation of the first response (RG_1), the server computes $C'_{1,\alpha} = (A_1^{r'_\alpha}, g_1^{R_\alpha} r_R (g_1^\alpha B_1)^{r'_\alpha})$ for $1 \le \alpha \le n$, where $B_1 = g_1^{-i} y_1^{r_1}$, and sends $C'_{1,\alpha}$ $(1 \le \alpha \le n)$ to the client. Only when $\alpha = i$, $C'_{1,i} = (g_1^{r_i r'_i}, g_1^{R_i} r_R y_1^{r_i r'_i})$ is the encryption of $g_1^{R_i} r_R$. When $\alpha \ne i$, $C'_{1,\alpha}$ is the encryption of $g_1^{R_\alpha} r_R g_1^{r'_\alpha}$, where r'_α is unknown to the client. Because the discrete logarithm is hard, the client cannot determine r'_α from $A_1^{r'_\alpha}$. Therefore, $g_1^{R_\alpha} r_R$ is blinded by the random factor $g_1^{r'_\alpha}$. In view of it, the client can retrieve the useful $g_1^{R_i} r_R$ only from $C'_{1,\alpha}$ $(1 \le \alpha \le n)$. Then following the Naor-Pinkas oblivious transfer protocol, the client can retrieve the encryption key k_{ij} only in the end of the phase.

In the private information retrieval phase, even if the client can retrieve more than one encrypted records, he/she can decrypt only one record with the encryption key k_{ij} retrieved in the first phase.

Based on the above analysis, we obtain the following result.

Theorem 4. *Assume that the discrete logarithm is hard and the Naor-Pinkas protocol is a secure oblivious transfer protocol, our protocol has server security.*

The previous solution [28] used the cell key of the form $g^{R_i} || g^{C_j}$ where g^{R_i}, g^{C_j} are the row and column keys, respectively. If the user queries the database once, the user can get one cell key only. However, if the user queries the database twice, the user is able to get 4 cell keys. In this paper we overcome this security weakness by using the cell key of the form $g_0^{g_1^{R_i} g_2^{C_j}}$, where both key components are protected by the discrete logarithm problem.

5 Performance Analysis

We now analyse the performance of our solution and show that it is very practical. The performance analysis consists of the computation analysis and the communication analysis. We supplement this analysis with a comparison with the protocol by Ghinita *et al.* [13], [14].

5.1 Computation

Since the most expensive operation in our protocol is the modular exponentiation, we focus on minimising the number of times it is required. We assume that some components can be precomputed, and hence we only consider the computations needed at runtime. Furthermore, we reduce the number of exponentiations required by the PIR protocol to the number of multiplications that are required. This will make the computational comparison between our solution and the solution of Ghinita *et al.* easier to describe.

The transfer protocol is initiated by the user, who chooses indices i and j. According to our protocol the user needs to compute $(A_1, B_1) = (g_1^{r_1}, g_1^{-i} y_1^{r_1})$ and $(A_2, B_2) = (g_2^{r_2}, g_2^{-j} y_2^{r_2})$. Since the user knows the discrete logarithm of

TABLE 1
Stage 1 Performance Analysis Summary

	Computation			Communication
	User	Server	Total	
Our Solution	7	$3n + 3m + 1$	$6 + 3n + 3m$	$4L + 2(m + n)2L + L$
Ghinita et al.	$4 + 4(n \times m)$	$4(n \times m)$	$4 + 4(n \times m) + 4(n \times m)$	$4L + 4(m \times n)2L$

both y_1 and y_2 (i.e. x_1 and x_2 respectively), the user can compute (A_1, B_1) and (A_2, B_2) as $(A_1, B_1) = (g_1^{r_1}, g_1^{-i+x_1 r_1})$ and $(A_2, B_2) = (g_2^{r_2}, g_2^{-j+x_2 r_2})$ respectively. Hence, the user has to compute 4 exponentiations to generate his/her query.

Upon receiving the user's query, the server needs to compute $((A_1)^{r'_\alpha}, g_1^{R_\alpha} r_R (g_1^\alpha (A_2))^{r'_\alpha})$ for $1 \leq \alpha \leq n$ and $((B_1)^{r'_\beta}, g_2^{C_\beta} r_C (g_2^\beta (B_2))^{r'_\beta})$ for $1 \leq \beta \leq m$. Since g^α and g^β can be precomputed and the server knows the discrete log of r_R and r_C, the server has to compute $3n+3m$ exponentiations, plus an additional exponentiation for computing γ.

The user requires 3 additional exponentiations to determine $K_{i,j}$. After the user has determined $K_{i,j}$, he/she can determine $X_{i,j}$ and proceed with the PIR protocol. This protocol requires 3 more exponentiations, 2 performed by the user and 1 performed by the server. In terms of multiplications, the user has to perform $2|N|$ operations and the server has to perform $|e|$ operations. The user also has to compute the discrete logarithm base h, log_h, of h_e. This process can be expedited by using the Pohlig-Hellman discrete logarithm algorithm [30]. The running time of the Pohlig-Hellman algorithm is proportional to the factorisation of the group order $O(\sum_{i=1}^{r} c_i(\lg n + \sqrt{p_i}))$, where r is the number of unique factors and n is the order of the group. In our case, the order of the group is $\pi_i = p_i^{c_i}$ and the number of unique factors is $r = 1$, resulting in running time $O(c(\lg p^c + \sqrt{p}))$.

When we compare our approach with the one by Ghinita *et al.* we find that our approach is computationally more efficient. Their protocol uses the homomorphic properties of the Paillier encryption scheme [27] in order to test whether a user is located in a cell or not. This requires the user to perform 4 exponentiations to compute the ciphertext of his/her coordinates, x and y. The server then has to compute $(4 \times (n \times m))$. The user has to decrypt at most all these ciphertexts $(4 \times (n \times m))$.

Once the user has determined his/her cell index he/she can proceed with the PIR protocol (described in [14]) to retrieve the data. The PIR is based on the Quadratic Residuosity Problem [22], which allows the user to privately query the database. Let t be the total number of bits in the database, where there are a rows and b columns. The user and server have to compute $2(\sqrt{a \times b}) \times \frac{|N|}{2}$ and $a \times b$ multiplications respectively. We remark that multiplying the whole database by a string of numbers, which is required by the PIR protocol based on the quadratic residuosity problem, is equivalent to computing g^e in our PIR protocol. The size of number e is principally defined by the prime powers. In general, it takes about $\eta = \sum_{i=1}^{N} log_2(\pi_i)$ bits to store e and we would expect to be multiplying $\eta/2$ of the time using the square-and-multiply method for fast exponentiation. This is roughly equivalent to $a \times b$ multiplications as required in the Ghinita *et al.* protocol.

5.2 Communication

Since we require the discrete logarithm to be intractable for security reasons, we set q' to be 1024 bits, which makes p roughly 1025 bits. Since q' and p are about the same size we set a common L as 1024 bits for analysis. In our proposed solution, the user needs $4L$ communications, while the server requires $2(m + n)2L + L$ communications in the oblivious transfer protocol. In the PIR protocol, the user and server exchange one group element each.

Since the solution by Ghinita *et al.* uses the Paillier encryption scheme, the size of one ciphertext in their scheme is $2L$. Based on this parameter, the user has to submit $4L$ bits to the server as his/her encrypted location. Then the server has to send $4 \times n \times m \times 2L$, for the user to determine his/her location. For the PIR based on the QRA, the user and server have to send $\sqrt{a \times b} \times L$. The performance analysis for stage 1 (user location test) and stage 2 (private information retrieval) are summarised in Tables 1 and 2 respectively, where the computation in Table 1 is in terms of exponentiation and the computation in Table 2 is in terms of multiplication.

When we analyse the difference in performance between our solution and the one by Ghinita *et al.*, we find that our solution is more efficient. The performance of the first stage of each protocol is about the same, except that our solution requires $O(m+n)$ operations while the solution by Ghinita *et al.* requires $O(m \times n)$. In the second stage, our protocol is far more efficient with respect to communication, in that it requires the transmission of only 2 group elements whereas the Ghinita *et al.* solution requires the exchange of an $a \times b$ matrix.

6 Experimental Evaluation

We implemented our location based query solution on a platform consisting of: a desktop machine, running the server software of our protocols; and a mobile phone, running the client software of our protocols. For both platforms, we measured the required time for the oblivious

TABLE 2
Stage 2 Performance Analysis Summary

	Computation			Communication
	User	Server	Total	
Our Solution	$O(c(\lg p^c + \sqrt{p})) + 2\|N\|$	$\|e\|$	$O(c(\lg p^c + \sqrt{p})) + 2\|N\| + \|e\|$	$2L$
Ghinita et al.	$2(\sqrt{a \times b}) \times \frac{\|N\|}{2}$	$a \times b$	$2(\sqrt{a \times b}) \times \frac{\|N\|}{2} + a \times b$	$\sqrt{a \times b}L$

TABLE 3
Oblivious Transfer Experimental Results for Desktop and Mobile Platforms

	Average Time (s)	
Component	**Desktop**	**Mobile**
$QueryGeneration_2$	—	23.90666
$ResponseGeneration_2$	4.57127	—
$ResponseRetrieval_2$	—	0.49123

TABLE 4
Private Information Retrieval Experimental Results for Desktop and Mobile Platforms

	Average Time (s)	
Component	**Desktop**	**Mobile**
$QueryGeneration_2$	—	23.90666
$ResponseGeneration_2$	4.57127	—
$ResponseRetrieval_2$	—	0.49123

transfer and private information retrieval protocols separately to test the performance of each protocol and the relative performance between the two protocols. The desktop machine that was used in the experiment is equipped with a Intel Core 2 Duo E8200 2.66GHz processor and 2GB of RAM. The implementation on this platform was written using Visual C++ under the Windows XP operating system. We used the Number Theory Library (NTL) [31] for computations requiring large integers and OpenSSL [1] to compute the SHA-1 hash.

The implementation on the mobile phone platform is programmed using the Android Development Platform, which is a Java-based programming environment. The mobile device used was a Sony Xperia S with a Dual-core 1.5 GHz CPU and 1 GB of RAM. The whole solution was executed for 100 trials, where the time taken (in seconds) for each major component was recorded and the average time was calculated. The parameters for our experiment were the same on both platforms, which are described next.

6.1 Experimental Parameters

6.1.1 Oblivious Transfer Protocol

In our implementation experiment for the oblivious transfer protocol, we generated a modified ElGamal instance with $|p| = 1024$ and $|q| = 160$, where $q|(p-1)$. We also found a generator a, and set $g_0 = a^q$ (g has order q).We also set a generator g_1, which has order $q-1$. We set the public matrix P to be a 25×25 matrix of key and index information.

We first measured the time required to generate a matrix of keys according to Algorithm 1. This procedure only needs to be executed once for the lifetime of the data. There is a requirement that each hash value of $g_0^{g_1^{R_i} g_1^{C_j}}$ is unique[3]. We use the SHA-1 to compute the hash $H(\cdot)$, and we assume that there is negligible probability that a number will repeat in the matrix.

6.1.2 Private Information Retrieval Protocol

In the PIR protocol we fixed a 15×15 private matrix, which contains the data owned by the server. We chose the prime set to be the first 225 primes, starting at 3. The powers for the primes were chosen to allow for at least a block size of 1024 bits ($3^{647}, 5^{442}, \ldots, 1429^{98}$). Random values were chosen for each prime power $e = C_i \ (mod\ \pi_i)$, and the Chinese Remainder Theorem was used to determine the smallest possible e satisfying this system of congruences.

Once the database has been initialised, the user can initiate the protocol by issuing the server his/her query. The query consists of finding a suitable group whose order is divisible by one of the prime powers π_i. We achieve this in a similar manner to Gentry and Ramzan [11]. We choose primes q_0 and q_1 and compute "semi-safe" primes $Q_0 = 2q_0\pi_i + 1$ and $Q_1 = 2q_1 + 1$. We set the modulus as $N = Q_0Q_1$ and group order as $\phi(N) = \phi(Q_0Q_1) = (Q_0 - 1)(Q_1 - 1)$. Hence, the order $\phi(N)$ has π_i as a factor. We set g to be a quasi-generator, such that the order of g also contains π_i. In our experiment, we set $|q_0| = |q_1| = 128$. This results in a modulus N which is roughly 1024 bits in length, which is equivalent to an RSA modulus.

6.2 Experimental Results

In both phases of our solution, there are 3 major steps: the user's query, the server's response, and the user decoding. Table 3 displays the average runtime on the desktop and mobile platforms, for each component of the oblivious transfer phase. Similarly, Table 4 presents the average times for each component of the private information retrieval protocol.

When we compare this outcome with our previous result [28], we find that the protocol is still practical. For this comparison, we consider the performance of the client the most important, since we assume that a server is very powerful. Compared with the previous work, the first stage on the client side is 4-7 times faster, while in the second stage the client side is 2 times slower. We must keep in mind that the client side was implemented on a desktop machine in the previous work, and hence made the second stage slower. Also, we replaced the hash algorithm with an exponentiation operation that reduced the group space for $g^{R_i}g^{C_j}$ from 1024 to 160 bits. This security of this structure was protected by an outer group of 1024 bits. Because the client cannot directly access $g^{R_i}g^{C_j}$, since the discrete logarithm is hard in the outer group, the client must operate in the outer group to remove the blinding factors. This contributed to faster execution in the first stage.

7 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 for a user to privately determine his/her location using oblivious transfer on a public grid. The second step involves 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 solution by Ghinita *et al.*, which is the most recent solution. We implemented a software prototype using a desktop machine and a mobile device. The software prototype demonstrates that our protocol is within practical limits.

3. We remark that we used one generator in G_1 to simplify the experiment.

Future work will involve testing the protocol on many different mobile devices. The mobile result we provide may be different than other mobile devices and software environments. Also, we need to reduce the overhead of the primality test used in the private information retrieval based 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 [15]. Once suitable strong solutions exist for the general case, they can be easily integrated into our approach.

ACKNOWLEDGMENTS

This work was supported in part by ARC Discovery Project (DP0988411) "Private Data Warehouse Query" and in part by NSF award (1016722) "TC: Small: Collaborative: Protocols for Privacy-Preserving Scalable Record Matching and Ontology Alignment".

REFERENCES

[1] (2011, Jul. 7) *Openssl* [Online]. Available: http://www.openssl.org/
[2] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," in *Proc. CRYPTO*, 1990, pp. 547–557.
[3] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan.–Mar. 2003.
[4] C. Bettini, X. Wang, and S. Jajodia, "Protecting privacy against location-based personal identification," in *Proc. 2nd VDLB Int. Conf. SDM*, W. Jonker and M. Petkovic, Eds., Trondheim, Norway, 2005, pp. 185–199, LNCS 3674.
[5] X. Chen and J. Pang, "Measuring query privacy in location-based services," in *Proc. 2nd ACM CODASPY*, San Antonio, TX, USA, 2012, pp. 49–60.
[6] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.
[7] M. Damiani, E. Bertino, and C. Silvestri, "The PROBE framework for the personalized cloaking of private locations," *Trans. Data Privacy*, vol. 3, no. 2, pp. 123–148, 2010.
[8] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. 3rd Int. Conf. Pervasive Comput.*, H. Gellersen, R. Want, and A. Schmidt, Eds., 2005, pp. 243–251, LNCS 3468.
[9] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inform. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[10] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. ICDCS*, Columbus, OH, USA, 2005, pp. 620–629.
[11] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," in *Proc. ICALP*, L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., Lisbon, Portugal, 2005, pp. 803–815, LNCS 3580.
[12] G. Ghinita, P. Kalnis, M. Kantarcioglu, and E. Bertino, "A hybrid technique for private location-based queries with database protection," in *Proc. Adv. Spatial Temporal Databases*, N. Mamoulis, T. Seidl, T. Pedersen, K. Torp, and I. Assent, Eds., Aalborg, Denmark, 2009, pp. 98–116, LNCS 5644.
[13] G. Ghinita, P. Kalnis, M. Kantarcioglu, and E. Bertino, "Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection," *GeoInformatica*, vol. 15, no. 14, pp. 1–28, 2010.
[14] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proc. ACM SIGMOD*, Vancouver, BC, Canada, 2008, pp. 121–132.
[15] G. Ghinita, C. R. Vicente, N. Shang, and E. Bertino, "Privacy-preserving matching of spatial datasets with protection against background knowledge," in *Proc. 18th SIGSPATIAL Int. Conf. GIS*, 2010, pp. 3–12.
[16] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. MobiSys*, 2003, pp. 31–42.
[17] T. Hashem and L. Kulik, "Safeguarding location privacy in wireless ad-hoc networks," in *Proc. 9th Int. Conf. UbiComp*, Innsbruck, Austria, 2007, pp. 372–390.
[18] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proc. 1st Int. Conf. SecureComm*, 2005, pp. 194–205.
[19] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1719–1733, Dec. 2007.
[20] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. Int. Conf. ICPS*, 2005, pp. 88–97.
[21] J. Krumm, "A survey of computational location privacy," *Pers. Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, Aug. 2009.
[22] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *Proc. FOCS*, Miami Beach, FL, USA, 1997, pp. 364–373.
[23] L. Marconi, R. Pietro, B. Crispo, and M. Conti, "Time warp: How time affects privacy in LBSs," in *Proc. ICICS*, Barcelona, Spain, 2010, pp. 325–339.
[24] S. Mascetti and C. Bettini, "A comparison of spatial generalization algorithms for lbs privacy preservation," in *Proc. Int. Mobile Data Manage.*, Mannheim, Germany, 2007, pp. 258–262.
[25] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proc. VLDB*, Seoul, Korea, 2006, pp. 763–774.
[26] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proc. CRYPTO*, vol. 1666, Santa Barbara, CA, USA, 1999, pp. 791–791.
[27] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, vol. 1592, Prague, Czech Republic, 1999, pp. 223–238.
[28] R. Paulet, M. Golam Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," in *Proc. ICDE*, Washington, DC, USA, 2012, pp. 44–53.
[29] B. Palanisamy and L. Liu, "MobiMix: Protecting location privacy with mix-zones over road networks," in *Proc. ICDE*, Hannover, Germany, 2011, pp. 494–505.
[30] S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.)," *IEEE Trans. Inform. Theory*, vol. 24, no. 1, pp. 106–110, Jan. 1978.
[31] V. Shoup, (2011, Jul. 7). *Number thoery library* [Online]. Available: http://www.shoup.net/ntl/
[32] L. Sweeney, "k-Anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.
[33] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proc. 16th ACM CCS*, Chicago, IL, USA, 2009, pp. 348–357.

Russell Paulet received the B.Sc. degree in computer science, the associated honors degree, and is currently studying for the Ph.D. degree at Victoria University in Melbourne, Australia. His honors thesis was on image processing, specifically dealing with atmospheric noise such as rain. The subject of his Ph.D. is cryptography and computer privacy, which is supervised by Dr. Xun Yi and co-supervised by Dr. Alasdair McAndrew. His current research interests include, but not limited to, cryptography, data mining and privacy-preserving data mining, applied mathematics, and computer protocols.

Md. Golam Kaosar is currently a Research Assistant with the School of Engineering and Science, Victoria University, Melbourne, VIC, Australia. He has received the B.Sc. degree in computer science and engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh, the M.S. degree in computer engineering from King Fahd University of Petroleum and Minerals (KFUPM), KSA, and the Ph.D. degree from School of Engineering and Science, Victoria University. Previously, he was with the Research Institute (RI), KFUPM, KSA, and several other universities.

Xun Yi is an Associate Professor with the School of Engineering and Science, Victoria University, Melbourne, VIC, Australia. His current research interests include applied cryptography, computer and network security, mobile and wireless communication security, and privacy-preserving data mining. He has published more than 100 research papers in international journals, such as *IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Wireless Communications, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Circuit and Systems, IEEE Transactions on Vehicular Technologies, IEEE Communication Letters, IEE Electronic Letters,* and conference proceedings. He has also been a program committee member for more than 20 international conferences, and leads numerous Australia Research Council (ARC) Discovery Projects.

Elisa Bertino is a Professor of Computer Science at Purdue University, West Lafayette, IN, USA and serves as a Research Director of CERIAS. Previously, she was a Faculty Member in the Department of Computer Science and Communication, University of Milan, Italy. Her current research interests include security, privacy, digital identity management systems, database systems, distributed systems, and multimedia systems. She is a fellow of IEEE and ACM. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems, and the 2005 IEEE Computer Society Tsutomu Kanai Award for pioneering and innovative research contributions to secure distributed systems.