Turkish Journal of Electrical Engineering and Computer Sciences

Volume 21
Number 7 *Sup. 1*

Article 3

1-1-2013

Preserving location privacy for a group of users

MAEDE ASHOURI-TALOUKI

AHMAD BARAANI DASTJERDI

ALİ AYDIN SELÇUK

TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1857 – 1870

doi:10.3906/elk-1109-8

Preserving location privacy for a group of users

Maede ASHOURI-TALOUKI,[1,*] **Ahmad BARAANI-DASTJERDI,**[1] **Ali Aydın SELÇUK**[2]
[1]Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran
[2]Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey

Received: 05.09.2011 • **Accepted:** 16.04.2012 • **Published Online:** 24.10.2013 • **Printed:** 18.11.2013

Abstract: Location privacy is an interesting problem that has been receiving considerable attention. This problem has been widely discussed from the individual point of view; however, there exist only a few works that support location privacy for a group of users. In this paper we consider the problem of supporting location privacy for a group of users during the use of location-based services (LBSs). We assume a group of users who want to benefit from a LBS and find the nearest meeting place that minimizes their aggregate distance. Each user in this scenario wants to protect his or her location from the LBS, outside attackers, and other group members. We show that individual solutions for location privacy cannot be directly applied to the group location privacy problem and a special solution must be developed. We identify the privacy issues for this group scenario and propose a resource-aware solution in order to satisfy these group privacy issues. Our solution is based on secure multiparty computation and the anonymous veto network protocol. The proposed protocol decreases the number of group queries to a large extent, as it only sends a single query to the LBS. Consequently, the LBS overhead to evaluate the query and the size of the LBS result are significantly decreased. The proposed protocol also protects the LBS from the excessive disclosure of points of interest and the LBS provider only needs to apply an existing private nearest neighbor (NN) query algorithm instead of an aggregate NN query algorithm. The performance and security analysis show that the protocol is secure against a partial collusion attack and a denial-of-service attack in a malicious model.

Key words: Location privacy, secure multiparty computation, location-based service, AV-net

1. Introduction

Location-based services (LBSs) offer a wide range of capabilities, because mobile users have the ability to ask for location-dependent queries from the spatial database and get the desired information based on their location at any time and from anywhere [1].

However, to get the correct answer, a user (or a group of users) must reveal his or her (their) exact location(s) to the LBS. This may raise many concerns about the location privacy [2]. Knowing the location of a user (or a group of users) could reveal sensitive information about her (their) health status, financial status, future activity, and political affiliations. As a result, several techniques have been proposed to protect the user's location privacy during the use of LBSs [3,4]. Unfortunately, most of these techniques only consider the location privacy of an individual user [4] and do not take into account the location privacy for a group of users. In this paper, we consider the location privacy problem for a group of users during the use of LBSs and propose a secure multiparty-based technique to solve it.

*Correspondence: maede.ashouri@gmail.com

Consider a scenario in which a group of users (working group) wants to have a critical face-to-face meeting (shown in Figure 1). They can use a LBS to find the nearest meeting place that minimizes their aggregate distance [5]. In order to find a meeting place, each user of the group submits a nearest neighbor (NN) query along with her location to the LBS. The LBS then returns the point(s) of P (a set of points of interest (POIs) that resides at the LBS database) with the smallest aggregate distance(s) to the set of queries [5]. This kind of NN query is known as a group NN query in the literature [5]. The aggregate distance may be the total distance of all of the group members or their maximum distance to the meeting point [5]. In this paper, we assume the total distance of all members as the aggregate distance.

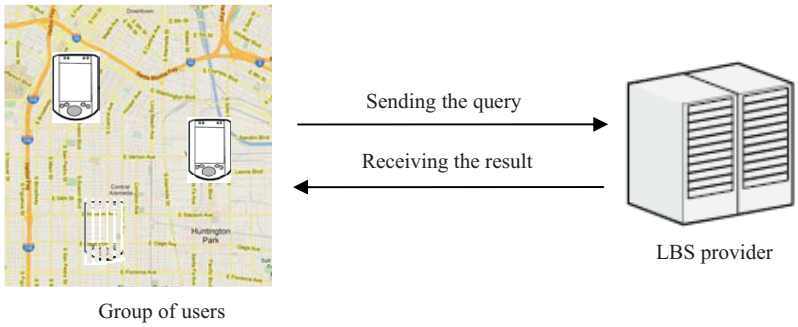

Figure 1. Model of the problem.

Providing exact locations to the LBS may jeopardize the location privacy of the group members. To avoid this privacy risk, Hashem et al. proposed a 2-phase method [6], where each user sends an imprecise location to the LBS and the LBS returns a set of candidate answer points with respect to the set of received imprecise locations. To determine the actual answer point, Hashem et al. proposed a private filtering algorithm that finds the exact result from the candidate answer set without violating the members' location privacy.

Although the work of Hashem et al. preserves the location privacy of each user in the group, it is an expensive method in terms of communication cost because it requires each user to send a distinct query (her cloaked region) to the LBS and the LBS must send back an answer set (not the exact answer), which has to be refined by the group members to determine the exact location.

In this paper, we identify the location privacy issues for a group of users and propose a resource-aware solution to satisfy them.

There are several major privacy issues in a group scenario:

- Preserving the location privacy within the group,
- Preserving the location privacy from anyone outside of the group,
- Preserving the meeting point location privacy.

By solving the first privacy issue, the location of each member will be protected from the other group members. The second privacy issue protects the location privacy of all of the members from anyone outside of the group, including the LBS.

The third privacy issue refers to protecting any location data that belong to the whole group. For example, in the above scenario, the location of the meeting place can be considered as the location data belonging to all

of the members, and therefore it must be protected. Or, if the members' meetings tend to be secret, the third privacy issue must also be satisfied.

Here, we present 2 definitions about the location privacy issues for a group of users:

Definition 1 *Let G_1 be a group of users. The IntraGroup Location Privacy encompasses the location privacy issues within G_1. Based on this property, the location of each member will be kept secret from the other members of G_1.*

Definition 2 *Let G_1 be a group of users. The InterGroup Location Privacy protects the privacy of all of the location data that belong to a single member or the data of the whole group from anyone outside of the group; this includes preserving the location privacy of all of the group members and the meeting point from anyone outside of G_1 (for example, LBSs and outside attackers).*

According to the above definitions, the focus of the group location privacy is on protecting the location privacy of all of the members and the meeting point, while the individual location privacy aims to protect one user's location. The group location privacy problem requires an additional privacy-preserving phase to find the exact result from the answer set; thus, special solutions need to be developed.

In this paper, we aim to preserve the location privacy of all of the members within the group and from anyone outside of the group. We leave preserving the meeting point location privacy for a future work.

Our main idea in this paper is to compute a location indicator as a group location and send that indicator to the LBS. The group's location indicator may be a minimum bounding rectangle (MBR) that encloses all of the group members or the centroid point of all of the group members. Both group location indicators (a MBR or a centroid point) guarantee that all of the users of the group will get the exact answer. When using a centroid as the group's location indicator, the answer set size becomes lower than that of a MBR. In particular, it is enough for the LBS to compute the nearest POI to the centroid in the case of a NN query (or the k nearest POIs in the case of a k-NN query) and to send it back to the group. Therefore, we use a centroid as the group's location indicator.

In general, the contribution of this paper can be summarized as follows:

- We identify the issues of the group location privacy and propose a decentralized protocol to protect these issues, even in the case of collusion.
- We propose a solution that is resource-aware, as it takes care as regards the communication cost and computation cost of each member.
- The proposed protocol preserves the privacy of the LBS content, as it discloses only a single POI in the case of a NN query (or a set of$k$ POIs in the case of a k-NN query), while previous works may lead to excessive disclosure of the LBS database [7–9].

The rest of the paper is organized as follows: the next section reviews related works in the field of location privacy. In Section 3, the proposed protocol is presented. The security analysis and performance discussion are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper.

2. Related works

There is a wide range of literature on preserving the user's location privacy during the use of a LBS. First, we briefly review the individual location privacy solutions, and then we focus on the approaches that support the location privacy for a group of users.

For the individual location privacy, there are 2 types of solutions [4]. The first is based on a trusted third party (TTP) that is responsible for cloaking the users' locations and mediating the communication link [10–14]. Thus, a LBS provider receives a query from the TTP and then sends the result back to the TTP. One drawback of this approach is that the TTP is a single point of failure and can affect the security of all of the users. Furthermore, the user must trust the TTP enough to disclose her exact location.

To overcome these drawbacks, a second approach is proposed: TTP-free methods [4]. In these methods, each user cloaks her location without the use of a TTP. The methods of this approach are collaboration-based [4,15–18], obfuscation-based [19–21], and private information retrieval-based (PIR-based) [22,23]. In collaboration-based methods [4,15–18], a user perturbs her exact location by collaboration with her peers. Obfuscation-based methods [19,20] degrade the quality of the location information by reducing the location precision. PIR-based methods [22,23] apply PIR cryptographic techniques to support the location privacy; in these methods, the LBS answers the queries without knowing the users' exact locations. TTP-free methods have some drawbacks; for example, the user must trust her peers [15,16]. Moreover, some of these methods need cooperation from the LBS, are not applicable in real-world scenarios, and/or are expensive in terms of the computation cost [22,23].

It is worth mentioning that collaboration-based methods are similar to the group location privacy paradigm because they protect the user location privacy through a group formation. Thus, in the following, we review these methods in more detail.

Chow et al. were the first to apply the group formation technique to cloak a single user's location [16]. In their method, the mobile user forms a group from her peers by contacting them via single-hop or multihop communication. The mobile user can then blur her exact location into a spatial cloaked region that covers the entire group of peers. The drawback of this approach is that the mobile user can learn the exact location of her peers.

PRIVÉ [24] and MobiHide [25] are 2 distributed approaches presented by Ghinita et al. that preserve the anonymity of a user issuing spatial queries to the LBS. Both methods are based on the Hilbert space-filling curve and assume that the user trusts her peers.

Solanas and Martínez-Ballesté [17] proposed a cryptographic-based method to preserve a single user's location privacy. In this method, which is similar to our protocol, a mobile user contacts peers in her cover range to learn their locations. The centroid point is then computed by the user as her fake location. The users' locations are masked by adding Gaussian noise with a zero mean to allow them to freely share their location without trusting their peers. However, if this procedure is applied several times with static users, then the user's location will be disclosed due to the cancellation of the Gaussian noise. To solve this problem, Solanas applies a privacy homomorphic encryption system [26], where each user encrypts her masked location with the LBS's public key and then shares the result with her peers.

Although applying privacy homomorphic encryption solves the static user problem, there is another problem with this method. Assume that the LBS is able to eavesdrop on the users' internal communications; therefore, in the consecutive usages with static users, the LBS can deduce the user's exact location due to the noise cancellation. Furthermore, this method is expensive in terms of the computation and communication cost because it requires one encryption per user; consequently, the exchanged message size will be large as well.

The protocol by Hu and Xu [27] preserves the individual user's location privacy by forming a group with no need for the user to trust her peers. In general, their method consists of 2 phases. In the first phase, the mobile user identifies her k peers through the proximity information and in the second phase, the minimum

bounding rectangle of this set of users is constructed through a specialized secure multiparty protocol. Although this approach alleviates the need for peer trusting, it constructs a large cloaked region and results in a larger answer set.

In contrast to the above methods, our protocol computes a single point as the group's location indicator and results in the smallest answer set of the methods: a single POI in the case of a NN query. In addition, the proposed protocol results in low computation and communication costs.

Although there are some works in the field of processing group NN queries [5,28], they do not consider user location privacy. To the best of our knowledge, there are only 2 papers on the subject of group location privacy: one written by Huang and Vishwanathan [29] and the other by Hashem et al. [6], which has 2 phases.

Huang and Vishwanathan's method [29], based on a garbled circuit (GC) and oblivious transfer (OT), assumes that the group members know the set of candidate POIs and presents a cryptographic solution to find the nearest POI from the members' locations. In this method, there are 2 special users: a creator who creates the encrypted circuit and an evaluator who evaluates the circuit. Each user starts an OT protocol with the creator to get her encrypted input bits and then transmits her encrypted input to the evaluator, and this should be repeated for each POI in the answer set. Thus, increasing the number of POIs and the number of users results in an increasing number of OT protocols, leading to protocol inefficiency.

Regarding Hashem's method, the first phase blurs the exact location of each user based on her peers' imprecise local location [30]. Each user then submits her cloaked location along with a query ID to the LBS. The query ID is issued by a group coordinator that is responsible for managing the communication of the group and submitting the NN query to the LBS.

Upon receiving all of the requests, the LBS provider evaluates them and returns a set of candidate answers (that includes the actual POI) along with their aggregate maximum and minimum distances to the users' cloaked regions. The second phase of Hashem's method is to determine the exact POI without revealing the users' locations. In this phase, each user updates the maximum and minimum aggregate distance to the candidate answers with respect to her actual location and the total travel distance to each data point is computed.

Although Hashem's method preserves the location privacy of all of the members, it has major drawbacks. For example, it still requires the group to send n distinct NN queries, which imposes a high communication cost. Moreover, computing the imprecise location requires each member to find her $k - 1$ peers and contact them to collect their local imprecise locations. Thus, the cloaking process requires additional communication and computation costs. Moreover, the LBS overhead to evaluate a group of NN queries is much higher than that of a single NN query, because the LBS processes each POI against a set of regions instead of a single region or a single point. Furthermore, the private filtering algorithm in Hashem's method imposes additional computation and communication costs.

Our proposed protocol protects the location privacy for all of the group members in an effective manner in terms of the computation and communication cost. The protocol not only sends a single NN query instead of n distinct NN queries, but also receives the actual POI from the LBS. Therefore, there is no need to apply a private filtering algorithm.

Our paper differs from [6] in 3 ways:

- We propose a solution based on secure multiparty computation to compute a location indicator for the group, while preserving the location privacy of all of the members.
- Our proposed solution avoids the need for evaluating a group of NN queries because the LBS only receives

one NN query. Thus, the LBS can apply any NN query-processing algorithm and, consequently, the LBS overhead will be very low.

- The size of the candidate answer set in the proposed approach is $O(1)$ in the case of a NN query or $O(k)$ in the case of a k-NN query. Moreover, there is no need to use a private filtering algorithm because the answer set only contains the exact result.

3. Proposed protocol

As mentioned earlier, the proposed protocol computes the centroid among the users of the group as the group's location indicator. The computation of a centroid must be carried out in a secure fashion, such that the location privacy of all of the members remains intact. Specifically, members of the group jointly and securely compute a function of their private inputs (their locations), such that the function outcome is the centroid coordinates. To protect the members' location privacy, group users must start a secure multiparty computation. The secure 2-party computation was first presented by Yao [31] and then extended to a secure multiparty computation [32]. Due to the inefficiency of Yao's protocol (GC), some research has focused on finding efficient protocols for specific problems of secure computation.

In this paper, we adopt the anonymous veto network (AV-net) protocol [33] to propose a secure multiparty centroid computation. This protocol was developed by Hao and Zielinski [34] in 2006 to solve the anonymous sender problem. In the first round of the AV-net, each member publishes an ephemeral public key g^{a_i}, and then each member is able to compute g^{b_i} based on Eq. (1):

$$g^{b_i} = \prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^{n} g^{a_j} \tag{1}$$

In the second round of the AV-net, each member publishes $g^{c_i b_i}$ (where c_i is a random number) if she wants to veto, or she publishes $g^{a_i b_i}$ if she does not want to veto the protocol. Upon aggregating all of the values, if no one vetoes, the result is 1 ($\prod g^{a_i b_i} = 1$) [33]. Otherwise, if even 1 user vetoes, the result would be a random number, unequal to 1 ($\prod g^{c_i b_i} \neq 1$), while preserving the anonymity of the vetoing user(s) [33].

We adopt the AV-net to mask the location coordinates of each party to compute the centroid. Similar to Hao's work, we assume that there is an authenticated public channel for each member of the group, which is essential for general secure multiparty computations [32]. Moreover, we assume that G is a finite cyclic group of the prime order q, in which the decision Diffie–Hellman problem is intractable and g is a generator of G [33]. All of the members $\{U_1, U_2, ..., U_n\}$ agree on $(G; g)$. We consider a malicious model as the protocol threat model. In a malicious model, the adversary is active and could behave arbitrarily, while in a semihonest model, each participant follows the protocol specification but tries to deduce some private information about the other participants [32].

The proposed protocol has only one phase: it is responsible for the secure computation of the centroid coordinates. This operation needs 2 broadcast rounds. The first round establishes AV-net masks and the second round securely computes the centroid coordinates using the masks. The protocol rounds are presented in Figure 2.

The Algorithm of the Proposed Protocol	
Round1 (AV-net values computation):	
i.	U_i selects a_i, a_i', where $a_i, a_i' \in_R Z_q$
ii.	$U_i \rightarrow *: \left(g^{a_i}, g^{a_i'}\right)$
iii.	U_i computes $g^{b_i} = \prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^{n} g^{a_j}$ and $g^{b_i'} = \prod_{j=1}^{i-1} g^{a_j'} / \prod_{j=i+1}^{n} g^{a_j'}$
Round 2 (secure centroid computation):	
i.	$U_i \rightarrow *: \left(g^{x_i} g^{a_i b_i} \bmod q, g^{y_i} g^{a_i' b_i'} \bmod q\right)$
ii.	U_i computes $X = \prod_{i=1}^{n} (g^{x_i} g^{a_i b_i}) \bmod q$, $Y = \prod_{i=1}^{n} \left(g^{y_i} g^{a_i' b_i'}\right) \bmod q$
iii.	$U_a \rightarrow LBS: (X, Y, n) = \left(g^{\sum x_i}, g^{\sum y_i}, n\right)$

Figure 2. The algorithm of the proposed protocol.

In the first round, each member of the group publishes a secret random value. Specifically, each participant U_i selects a random secret value, $a_i \in_R Z_q$, and broadcasts g^{a_i} along with a zero-knowledge proof for a_i. After finishing this round, each party U_i computes g^{b_i} based on Eq. (1) (the same is true for $g^{a_i'}$ and $g^{b_i'}$).

Next, each party masks her location coordinate by publishing $g^{x_i} g^{a_i b_i}$ and $g^{y_i} g^{a_i' b_i'}$ along with a zero-knowledge proof of a_i, x_i, and y_i, where x_i and y_i are her location coordinates. Multiplying the published values results in the canceling of the AV-net masks and computing of the final summation of x and y.

More exactly, since a_i and b_i are AV-net values, then $\sum a_i b_i = 0$ [35]. Thus, multiplying all of the published values for $g^{x_i} g^{a_i b_i}$ results in the summation of the x coordinates of all of the users ($X = g^{\sum x_i}$), which are discrete logarithms to the base g. The same is done to compute the summation of the y coordinates ($Y = g^{\sum y_i}$).

Finally, U_a, a member of the group randomly chosen as the group agent to communicate with the LBS, sends a single NN query along with the summation of the x and y coordinates and the number of group members (n).

Upon receiving the query, the LBS finds the discrete logarithm of X and Y by applying the kangaroo method [36] (since x_i and y_i are location coordinates, the upper bounds of $\sum x_i$ and $\sum y_i$ will be clear) and then divides the result by n to get the centroid coordinates. It is worth mentioning that the coordinate data is usually an integer between a 6- or 7-decimal digit that requires about 20 bits. Thus, $\sum x_i$ (or $\sum y_i$) will be a small number and determining $\sum x_i$ from $g^{\sum x_i}$ (or $\sum y_i$ from $g^{\sum y_i}$) will be done efficiently. Afterwards, the LBS executes a conventional NN query-processing algorithm to obtain the point(s) of P with the smallest distance from the centroid and returns the result to U_a. Finally, U_a broadcasts the result to the group and the protocol terminates.

As explained earlier, as we consider a malicious model as the protocol threat model, the protocol participants may deviate from the protocol specification, for example by sending incorrect values during the protocol rounds. To force members to behave according to the protocol specification, we must use a zero-knowledge proof in each round of the protocol. Using a zero-knowledge proof makes each member follow the protocol specification; otherwise, her misbehavior will be discovered by the honest members, since the proof

verification fails. Because of noninteractivity properties, we use Schnorr's signature [37], which is similar to Hao's work. To prove the knowledge of the exponent, the prover sends $\{g^v, r = v - a_i h$, where $v \in_R Z_q$ and $h = H(g, g^v, g^{a_i}, i)$. To verify this proof, one can check whether g^v is equal to $g^r g^{a_i h}$.

To prove the knowledge of exponents x_i in the second round of the protocol, each party goes through the following 3 steps (the same is true for the proof of knowledge of exponent y_i):

- Select at random $v \in Z_q$.
- Compute $h=H\left(g,g^v,g^{v'},g^{b_i},(g^{b_i})^v,g^{a_i},g^{x_i}g^{a_i b_i},i\right)$.
- Send $\left(g^v,(g^{b_i})^v g^{v'}, r = v-a_i h, r'=v'-x_i h\right)$.

The proof can be verified by the following 2 checks:

1. $g^v =^? g^r (g^{a_i})^h$.
2. $(g^{b_i})^v g^{v'} =^? (g^{b_i})^r g^{r'} \left(g^{x_i} g^{a_i b_i}\right)^h$.

In the following section, the security properties of the proposed protocol are investigated and it is shown that the proposed protocol achieves its goal even in the case of malicious members (active adversaries).

4. Security analysis

In this section, we first analyze the protocol's behavior in the case of malicious members, and then we present the privacy properties of the proposed protocol.

According to the definition by Goldreich et al. [32], a malicious member may abort the protocol execution at any time. She can also send fake values, i.e. she can modify her AV-net masks to prevent the protocol from achieving its goal. It is worth mentioning that it is not possible to prevent malicious parties from changing their location coordinates [32]; this factor is the same for every protocol that runs in a malicious model [32]. Moreover, malicious parties may collude to violate the honest members' privacy. We consider these misbehaviors and analyze how the protocol can overcome them.

An abortion of the protocol execution can occur in the first or second round. If a malicious member refuses to participate in the protocol execution before the protocol starts, other members can enter the protocol and get the desired results. Refusing to participate after finishing the first round can easily be rectified. At this point, the honest parties can identify and exclude the malicious member through the zero-knowledge verification of the second round and they can restart the protocol at the second round.

Publication of an incorrect value during the computation of an AV-net mask can cause a denial-of-service (DoS) attack to occur, which prevents the protocol from fulfilling its task. To cause a DoS attack, a malicious party must use a fake b_i value instead of correct one as $\sum a_i b_i \neq 0$. Because of the zero-knowledge proof, however, no one can do this [35] because it requires the malicious party to demonstrate a consistent knowledge proof for the fake value. Upon attempting to verify the zero-knowledge proof, everyone would realize that an attack had occurred because the verification would fail. The group could then expel the attacker and restart the protocol without violating their location privacy.

Generally, because of the zero-knowledge proof, even malicious parties follow the protocol for fear of being detected, and consequently the protocol achieves its goal.

In a collusion attack, some malicious members may collude to discover the location of an honest member. There are 2 types of collusion attacks: full collusion and partial collusion. Generally, in a full collusion attack, all of the participants collude against one user in the network. However, it is impractical to have all of the participants colluding against just one [35]. Thus, we only consider a partial collusion, which involves only some participants.

Assume that all of the group members except U_j collude against U_i to discover U_i's location. The colluding members ($n-2$ members) aim to compute x_i from $g^{x_i}g^{a_ib_i}$. Computing x_i requires the colluders to find the AV-net masks. To reveal the AV-net masks, it is enough for the attackers to find b_i, but the AV-net structure guarantees that "b_i is a secret random value to attackers in partial collusion against participant U_i" [35]. Therefore, the colluding parties cannot get any information about b_i, and they consequently fail to discover the location coordinates of U_i.

According to Yang et al., a protocol is called t-private "if no collusion containing at most t parties can get any additional information from its execution" [38]. Based on the above discussion, our proposed protocol will be a $(n-2)$-private protocol.

Considering the privacy properties, the proposed protocol preserves the location privacy of all of the members within the group. If a malicious member tries to discover the location coordinates of another member U_i, she has to cancel the AV-net masks of U_i; however, as mentioned earlier, the AV-net masks cannot be cancelled in a partial collusion. In the worst case, if the AV-net masks have been revealed by a full collusion attack, the attackers will learn the user's coordinates, but, as noted above, full collusion is an impractical situation [13].

The protocol also protects all of the members' location from anyone outside of the group. An LBS provider only learns the centroid coordinates of the members, nothing else. Moreover, the LBS cannot obtain any useful information by eavesdropping on the group's communications, as each member masks her location coordinates with the AV-net values. Discovering a user's coordinates by eavesdropping on the group's messages requires the LBS to cancel the AV-net masks or solve the discrete logarithm (DL) problem, but, as discussed in the previous paragraph, b_i is a secret random value to the attackers in a partial collusion attack. Moreover, under the difficulty of the DL problem [7], the LBS cannot get any useful information by eavesdropping on the group's communication; thus, the LBS cannot disturb members' location privacy. The situation is the same for the other attackers. As a result, the location coordinates of each user are kept hidden from the LBS and other outside attackers.

5. Experimental results

In this section, we evaluate the performance of the proposed protocol through extensive experiments. We use the Sequoia dataset (www.rtreeportal.com), which contains 62,556 real location coordinates (POIs) in California, normalize it in a square of 10,000 $\times$ 10,000 units, and index it using an R*-tree index. We use the MATLAB environment to implement the protocol; we also implement the LBS algorithm proposed by Hashem et al. [6]. For fairness, we first consider a semihonest model for the proposed protocol, as in Hashem's method, and present the results, and then we measure the required time of the proposed protocol in the malicious model and compare it with Hashem's method in a semihonest model. The Table summarizes the values used for each parameter in our experiments.

Table. Parameters of the system.

System parameter	Values	Default value
k (required data points)	2, 4, 8, 16, 32	2
Group size	16, 64, 256, 1024	256
User query rectangle area	0.001% to 0.01%	0.005
Group area size	1% to 10%	1%

We use various group sizes: 16, 64, 256, and 1024 members. We vary the size of the area that encloses the set of group members, from 1% up to 10% of the total space, and then we randomly generate 1024 point locations, which are uniformly distributed in the considered areas. The size of module q for the cryptographic operation is set to 1024 bits. The experiments are run on an Intel P3 2.01 GHz desktop with 1 GB of RAM. We compare the experimental results of our protocol against Hashem's method. As mentioned in the related work section, Huang's method leads to a high computation and communication cost, and thus we do not compare it with our protocol.

In terms of efficiency, we measure the query round-trip time and present the results in Figure 3. This time consists of the time taken by each phase of the protocol plus the LBS evaluation time. Figure 3a shows that Hashem's method provides a higher query response time than that of our protocol, especially as the group's size grows. Specifically, the larger the size of the group, the higher the LBS computation would be in Hashem's method (Figure 3b). In our protocol, the LBS always receives the centroid point and retrieves the nearest POI for it; thus, the LBS overhead does not change by increasing the size of the group. Apart from the LBS overhead, in the proposed protocol, there is no need to refine the answer because the LBS delivers the exact nearest POI. Hence, the overall required time to complete our protocol rounds is much lower than that of Hashem's.

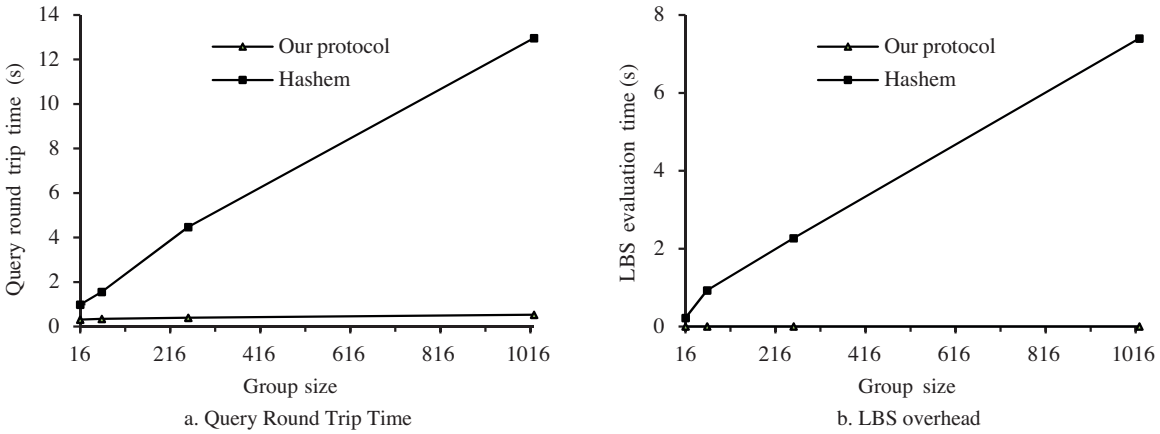

Figure 3. Query round-trip time and LBS evaluation time for different group sizes.

Figure 4a shows that Hashem's method causes a higher round-trip time than that of our protocol, especially when the area size of the group increases. This is because the LBS overhead will increase by increasing the size of the entire region that encloses all of the members, as shown in Figure 4b.

The proposed protocol uses the zero-knowledge proof to demonstrate the knowledge of the discrete logarithms, but this imposes an additional computation cost. We use an efficient knowledge proof [37] system to decrease this cost. It is important to note that any secure multiparty protocol needs a zero-knowledge proof system to be secure against malicious adversaries [33]; thus, this cost is unavoidable. It is worth mentioning that

Hashem assumes a semihonest threat model, while our protocol is secure in the malicious model. Thus, under fair conditions, where both protocols consider a malicious model, the zero-knowledge operation time would be added to Hashem's protocol, thus resulting in the scale of the diagrams in Figure 3a. In Figure 5, the required time of the proposed protocol in the malicious model is presented for a group area size of 2%, whereas the running time of Hashem's method is presented in a semihonest model. As shown in Figure 5, the total time of the proposed protocol is still lower than that of Hashem's, because the LBS evaluation time of Hashem's method dominates the time of the zero-knowledge operations in our protocol.

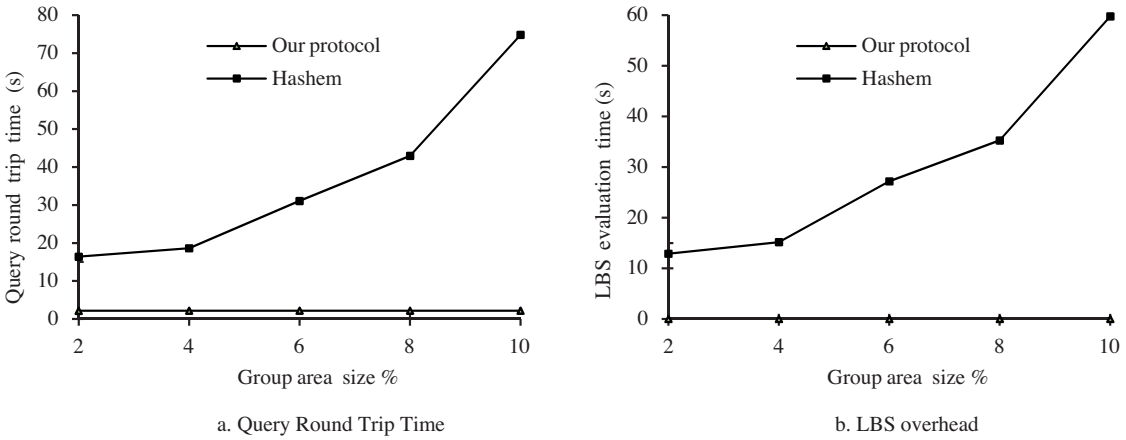

Figure 4. Query round-trip time and LBS evaluation time for different group area sizes.

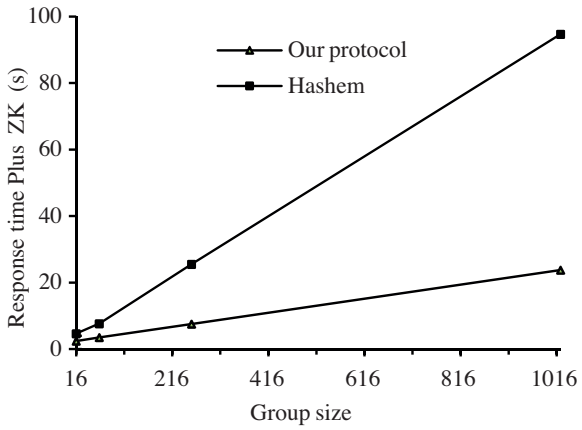

Figure 5. Query round-trip time considering the required time of the zero-knowledge operations (ZK).

In Figure 6, the size of the answer set is presented. Since the proposed protocol only sends the centroid coordinates to the LBS and receives only one POI in the case of a NN query (or k POIs in the case of a k-NN query), the size of the LBS response is much smaller than in Hashem's method. Therefore, the proposed protocol not only decreases the bandwidth consumption, it also prevents the LBS from excessive disclosure. It is worth mentioning that the LBS message in Hashem's method consists of the candidate answer set plus the maximum and minimum aggregate distance values of each point in the answer set to the centroid, and so the size of the LBS message is larger than that of our protocol.

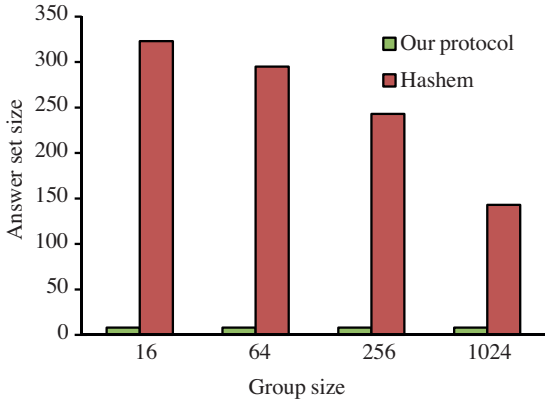

Figure 6. Comparing the answer set size.

As previously mentioned, the proposed protocol is a resource-aware method. This property is verified by the experimental evaluation, since it decreases the bandwidth by sending only 1 request and by receiving only the needed POIs.

To compare the intragroup communication cost, we count the number of intragroup messages exchanged during the protocol execution. In our protocol, each user publishes 2 messages during the execution, so the total number of messages will be *2n*. In contrast, in phase 1 of Hashem's method, each user collaborates with her neighbors to find her imprecise location. If the number of neighbors of each user is equal to m, then the user will receive m messages containing the neighbors' local cloaked regions; hence, the total number of intragroup messages in phase 1 will be equal to nm. In addition, phase 2 of Hashem et al.'s method requires 1 message to be sent per user. Therefore, the number of intragroup messages will be $nm + n$.

Preserving the meeting place location privacy in the proposed protocol is achieved via a single modification. Specifically, after finding the centroid in phase 1, U_a forms a small rectangle that contains the centroid and sends this area to the LBS instead of sending the centroid. The LBS then returns a set of answer points to the group, rather than a single answer point. Each member can find the exact meeting point from the answer set by determining the point with the minimum distance to the centroid, and so there is no need to refine the answer. It is worth mentioning that the area of the cloaked rectangle must be small enough to preserve the reasonable cardinality of the answer set.

6. Conclusion

This paper considers the problem of the location privacy for a group of users who may ask a LBS provider for a meeting place that minimizes their aggregate distance. We identify the location privacy issues in a group scenario and propose a distributed protocol to address them. The proposed protocol protects the location privacy of each group member from other members in the group and from anyone outside the group in a malicious model. Our protocol relies on the AV-net structure to hide the users' locations and computes the centroid as the group's location indicator. The proposed protocol decreases the bandwidth consumption to a high extent because it sends a single NN query (or a single k-NN query) to the LBS and receives a single POI (or a set of k POIs) from it. Our protocol also protects the LBS from excessive disclosure, while the previous works lead to the disclosure of a large number of POIs. The experimental results show that our protocol is more efficient than previous works in terms of the computation and communication costs. Furthermore, the security analysis of the proposed protocol shows that the proposed solution is secure against collusion and disruption attacks in a malicious model.

Acknowledgments

This work was partially supported by the CyberSpace Research Institute of the Islamic Republic of Iran.

References

[1] G. Zhong, U. Hengartner, "A distributed k-anonymity protocol for location privacy", IEEE International Conference on Pervasive Computing and Communications, pp. 253–262, 2009.

[2] P. Bhaskar, S.I. Ahamed, "Privacy in pervasive computing and open issues", 2nd International Conference on Availability, Reliability and Security, pp. 147–154, 2007.

[3] M. Langheinrich, "A privacy awareness system for ubiquitous computing environments", Proceedings of the 4th International Conference on Ubiquitous Computing, pp. 237–245, 2002.

[4] A. Solanas, J. Domingo-Ferrer, A. Martínez-Ballesté, "Location privacy in location-based services: beyond TTP-based schemes", Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, pp. 12–23, 2008.

[5] D. Papadias, Y. Tao, K. Mouratidis, C.K. Hui, "Aggregate nearest neighbor queries in spatial databases", ACM Transactions on Database Systems, Vol. 30, pp. 529–576, 2005.

[6] T. Hashem, L. Kulik, R. Zhang, "Privacy preserving group nearest neighbor queries", Proceedings of the 13th International Conference on Extending Database Technology, pp. 489–500, 2010.

[7] W. Diffie, M.E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Vol. 22, pp. 644–654, 1976.

[8] M.F. Mokbel, C.Y. Chow, W.G. Aref, "The new Casper: query processing for location services without compromising privacy", Proceedings of the 32nd International Conference on Very Large Data Cases, pp. 763–774, 2006.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, D. Papadias, "Preserving location-based identity inference in anonymous spatial queries", IEEE Transactions on Knowledge and Data Engineering, Vol. 19, pp. 1719–1733, 2007.

[10] B. Bamba, L. Liu, P. Pesti, T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid", Proceedings of the 17th International Conference on World Wide Web, pp. 237–246, 2008.

[11] M. Gruteser, D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking", Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42, 2003.

[12] U. Hengartner, P. Steenkiste, "Protecting access to people location information", Proceedings of the 1st International Conference on Security in Pervasive Computing, pp. 25–38, 2003.

[13] L. Sweeney, "k-Anonymity: a model for protecting privacy", International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, Vol. 10, pp. 557–570, 2002.

[14] R. Srikanth, L.K. Awasthi, "Privacy for mobile users in location-based services", MES Journal of Technology and Management, Vol. 2, pp. 93–98, 2011.

[15] A. Solanas, A. Martínez-Ballesté, "Privacy protection in location-based services through a public-key privacy homomorphism", Proceedings of the 4th European Conference on Public Key Infrastructure: Theory and Practice, pp. 362–368, 2007.

[16] C. Chow, M.F. Mokbel, X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based services", Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, pp. 171–178, 2006.

[17] A. Solanas, A. Martínez-Ballesté, "A TTP-free protocol for location privacy in location-based services", Computer Communications Journal, Vol. 31, pp. 1181–1191, 2008.

[18] T. Hashem, L. Kulik, "Don't trust anyone: privacy protection for location-based services", Journal of Pervasive Mobile Computing, Vol. 7, pp. 44–59, 2011.

[19] C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, P. Samarati, "An obfuscation-based approach for protecting location privacy", IEEE Transactions on Dependable and Secure Computing, Vol. 8, pp. 13–27, 2011.

[20] M.L. Yiu, C.S. Jensen, X. Huang, H. Lu, "Spacetwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services", Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, pp. 366–375, 2008.

[21] R. Dewri, "Location privacy and attacker knowledge: who are we fighting against?", 7th International ICST Conference on Security and Privacy in Communication Networks, pp. 1–20, 2011.

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

[23] F. Olumofin, P.K. Tysowski, I. Goldberg, U. Hengartner, "Achieving efficient query privacy for location based services", Proceedings of the 10th International Conference on Privacy Enhancing Technologies, pp. 93–110, 2010.

[24] G. Ghinita, P. Kalnis, S. Skiadopoulos, "PRIVÉ: Anonymous location-based queries in distributed mobile systems", Proceedings of the 16th International Conference on World Wide Web, pp. 371–389, 2007.

[25] G. Ghinita, P. Kalnis, S. Skiadopoulos, "MobiHide: a mobile peer-to-peer system for anonymous location-based queries", Proceedings of the 10th International Conference on Advances in Spatial and Temporal Databases, pp. 221–238, 2007.

[26] T. Okamoto, S. Uchiyama, "A new public-key cryptosystem as secure as factoring", International Conference on the Theory and Application of Cryptographic Techniques, pp. 308–318, 1998.

[27] H. Hu, J. Xu, "Non-exposure location anonymity", Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 1120–1131, 2009.

[28] K. Deng, S. Sadiq, X. Zhou, H. Xu, G.P.C. Fung, Y. Lu, "On group nearest group query processing", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, pp. 295–308, 2012.

[29] Y. Huang, R. Vishwanathan, "Privacy preserving group nearest neighbour queries in location-based services using cryptographic techniques", Proceedings of the IEEE Global Communications Conference, pp. 1–5, 2010.

[30] T. Hashem, L. Kulik, "Safeguarding location privacy in wireless ad-hoc networks", Proceedings of the 9th International Conference on Ubiquitous Computing, pp. 372–390, 2007.

[31] A.C. Yao. "How to generate and exchange secrets", 27th IEEE Symposium on Foundations of Computer Science, pp. 162–167, 1986.

[32] O. Goldreich, S. Micali, A. Wigderson, "How to play any mental game or a completeness theorem for protocols with honest majority", 9th ACM Conference on Theory of Computing, pp. 218–229, 1987.

[33] F. Hao, P. Ryan, P. Zielinski, "Anonymous voting by 2-round public discussion", IET Information Security, Vol. 4, pp. 62–67, 2010.

[34] F. Hao, P. Zielinski, "A 2-round anonymous veto protocol", 14th International Workshop on Security Protocols, pp. 202–211, 2006.

[35] F. Hao, P. Zielinski, "The power of anonymous veto in public discussion", Springer Transactions on Computational Sciences Journal, Vol. 5430, pp. 41–52, 2009.

[36] G. Zhong, I. Goldberg, U. Hengartner. "Louis, Lester and Pierre: three protocols for location privacy", Proceedings of the 7th International Conference on Privacy Enhancing Technologies, pp. 62–76, 2007.

[37] C.P. Schnorr, "Efficient signature generation by smart cards", Journal of Cryptology, Vol. 4, pp. 161–174, 1991.

[38] B. Yang, H. Nakagawa, I. Sato, J. Sakuma, "Collusion-resistant privacy-preserving data mining", Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 483–492, 2010.