Computer Communications 35 (2012) 1527–1533

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

GLP: A cryptographic approach for group location privacy

Maede Ashouri-Talouki [a,*], Ahmad Baraani-Dastjerdi [a], Ali Aydın Selçuk [b]

[a] Dept of Comp. Eng., Faculty of Engineering, The University of Isfahan, Isfahan, Iran
[b] Dept of Comp. Eng., Faculty of Engineering, Bilkent University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 24 October 2011
Received in revised form 22 April 2012
Accepted 23 April 2012
Available online 1 May 2012

Keywords:
Location privacy
Group nearest neighbor query
Secure multiparty computation
Location-based service

ABSTRACT

Recently, location privacy during the use of location-based services (LBSs) has raised considerable concerns. There is a wide literature on location privacy from the individual point of view; however, there exist only a few works to support location privacy for a group of users. In this paper, we consider location privacy issues for a group of users who may ask an LBS for a meeting place that minimizes their aggregate distance. The proposed solution, which we call the Group Location Privacy (GLP) protocol, is based on the Anonymous Veto network (AV-net) and homomorphic encryption. It preserves the location privacy of all users even in the case of collusion. Our solution also tries to minimize the LBS overhead for nearest neighbor (NN) queries and communication, i.e., to decrease the number of NN queries sent to an LBS and the number of points of interest (POIs) it returns. Furthermore, GLP greatly decreases the bandwidth usage to a high extent and protects the LBS provider from excessive disclosure of POIs. We discuss the performance and security analysis of the GLP protocol and show that the proposed protocol is secure against partial collusion in a malicious model.

1. Introduction

With location-based services (LBSs), mobile users are able to ask location-dependent queries and receive the desired information based on their location at any time and from anywhere [1]. For example, a user can ask "Where is the nearest restaurant to my location?" or a group of users can ask "Where is the nearest meeting place that minimizes our aggregate distances?" [2].

To benefit from these services, a user must reveal her exact location to the LBS, but this jeopardizes her location privacy [3]. Knowing a user's location could reveal sensitive private information such as her health status, financial status, future activity and political affiliations. Several techniques have been proposed to preserve user location privacy during the use of LBSs [4,5], but most of them solely preserve the location privacy of an individual user [4] and do not provide location privacy for a group of users. In this paper, we consider the location privacy problem for a group of users and propose a resource-aware technique to solve it.

We believe that the group location privacy problem is somewhat different from the individual location privacy problem. In particular, there are various privacy issues in the group location privacy problem. As an example, consider a scenario in which a group of users (a working group) needs to urgently meet. They can use an LBS provider to find the nearest meeting place that minimizes their aggregate distances [6]. To get the desired result, each user sends a nearest neighbor (NN) query along with her location to the LBS. The LBS evaluates the set of NN queries (a group of NN queries or an aggregate NN query [6]) and retrieves the points of interest (POIs) with the smallest aggregate distances from them. Here, the aggregate distance is the total distance of all group members from the meeting location [6].

We distinguish three major privacy issues in the group location privacy scenario: (*i*) Preserving the location privacy of each group member inside the group (this is called *intragroup location privacy* throughout this paper), (*ii*) preserving the location privacy of each group member from anyone outside the group, including the LBS and outside attackers and (*iii*) preserving the location privacy of the meeting place. The latter two issues together are called *intergroup location privacy*. The third privacy issue on its own is needed whenever group members want to have a secret meeting.

According to the above discussion, the focus of group location privacy is on protecting the location privacy for all group members (from users inside and from anyone outside the group) and also on preserving meeting place location privacy (in the case of a secret meeting); individual location privacy aims to protect one user's location privacy. For the above reasons, the techniques of the latter cannot directly be applied to the former; special solutions need to be developed.

In this paper, we aim to preserve the location privacy of all members inside the group and from anyone outside the group. We assume that our group does not intend to have a secret

* Corresponding author. Tel.: +98 311 7934010; fax: +98 311 793 2670.
E-mail addresses: ashoori@eng.ui.ac.ir, maede.ashouri@gmail.com (M. Ashouri-Talouki), ahmadb@eng.ui.ac.ir (A. Baraani-Dastjerdi), selcuk@cs.bilkent.edu.tr (A.A. Selçuk).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.04.019

meeting, and thus only provide some brief information about preserving meeting location privacy, leaving the details for a future work.

To the best of our knowledge, Hashem et al. [7] proposed the first solution for preserving group location privacy. In the first phase of this method, each user submits her imprecise location to the LBS and the LBS returns a set of candidate answers with respect to the set of received imprecise locations. In the next phase, Hashem et al. propose a private filtering algorithm that determines the exact result from the answer set without violating members' location privacy.

Although Hashem et al.'s work preserves the location privacy of each user inside and outside the group, it suffers from a high communication cost. It requires each user to send a distinct query (her cloaked region) to the LBS and the LBS to send back a set of candidate answer points that contains the exact result (instead of only sending back the exact result). Further, the set of candidate answers must be refined by the group members to determine the exact answer, which imposes an additional communication cost.

In this paper, we propose a decentralized resource-aware protocol called Group Location Privacy (GLP) to protect location privacy for a group of users while considering communication and computation costs. In GLP, instead of sending several messages, group members collaboratively construct a single message that contains one group location descriptor. This descriptor could be a minimum bounding rectangle (MBR) that encloses all group members or it could be the centroid point of all group members. Using the centroid as the group location descriptor results in some interesting properties. The first one decreases the answer set size, because it is enough for the LBS to return the nearest POI to the centroid in the case of an NN query (or the k nearest POIs in the case of a k-NN query). Secondly, there is no need to refine the answer set because it only contains the exact result. Preserving the privacy of the LBS content is the third property, since the LBS only discloses a single POI in the case of an NN query (or a set of k POIs in the case of a k-NN query); previous works may lead to excessive disclosure of LBS content [2,3,7]. Because of these properties, we use the centroid as the group location descriptor.

The rest of the paper is organized as follows: The next section reviews related works in the field of location privacy. Section 3 presents the proposed protocol. The security and efficiency analysis are presented in Section 4. Section 5 compares the proposed protocol with the previous work and Section 6 concludes the paper.

2. Related works

The works presented here preserve individual user location privacy based on the group formation idea; Chow et al. [8] were the first to propose a cloaking method based on this technique. In their method, the mobile user forms a group of her peers by contacting them via single-hop or multi-hop communication. Then, she blurs her exact location into a cloaked region that encompasses the entire group. The weakness of Chow's method is that the mobile user can learn the exact location of her peers.

PRIVE [9] and MOBIHIDE [10], based on the Hilbert Space Filling Curve, assume that users trust their peers. In PRIVE, users are partitioned according to their Hilbert values and the cluster head is responsible for location cloaking. In MOBIHIDE, the mobile user constructs a hash table of other users' locations and cloaks her location by randomly selecting a number of consecutive entries in the table.

Solanas et al. [11] propose a cryptography-based modular method to preserve single-user location privacy. In a simple scheme, a mobile user contacts her peers to learn their masked locations. Then, the centroid point is computed by the user as her fake location. Since locations are masked by adding Gaussian noise with zero mean, users can freely share their locations without trusting their peers. But, if this procedure is applied several times by static users, the user location will be disclosed due to the cancellation of Gaussian noise [11]. As a solution, Solanas et al. extend their scheme by applying a random chain and a privacy homomorphic encryption system, such that each user receives a value from her predecessor in the random chain and then adds her encrypted masked location (with the LBS's public key) to it, then sends the result to another randomly selected peer. This protocol protects user location privacy from the peers and from the LBS. In Solanas et al.'s method, however, if the LBS eavesdrops on the group's internal communication, users' noise-added locations would be revealed in consecutive usages. This factor may lead to reveling the actual location if the user is static, which may or may not be a realistic scenario, depending on the application.

Hu et al. propose a two-phase protocol [1] to preserve individual user location privacy; they apply a group formation technique such that there is no need for users to trust their peers. In the first phase of their method, the mobile user identifies her k peers through proximity information; in the second phase, the MBR of this set of users is constructed through a specialized secure multiparty protocol. This approach is designed for k-anonymity [1], so although it alleviates the need for peer trusting, it constructs a large cloaked region; having a large cloaked region increases the size of the answer set and communication cost.

As explained in the previous section, Hashem et al.'s method [7] is specially designed for a group location privacy scenario, thus, we describe it here in more detail. The protocol consists of two phases. In the first phase, each user U_i registers with the coordinator and gets a query ID (the coordinator is selected randomly before the protocol starts). Then, each U_i blurs her exact location l_i based on her neighbors' imprecise location [12] and submits her cloaked location R_i to the LBS. The coordinator sends a description of the query to the LBS. After receiving all cloaked regions, the LBS returns a set of candidate POIs A (that includes the meeting location) to the coordinator, along with the aggregate maximum $d_{\max}(p_j)$ and minimum $d_{\min}(p_j)$ distances of each $p_j \in A$ from the cloaked regions.

The second phase privately determines the exact POI through sequentially updating the aggregate maximum and minimum distances of each POI in A with the users' actual distances. This phase can be conducted with or without a coordinator. In the first case, the coordinator sends the LBS result to a randomly selected member U_i. The user U_i updates $d_{\max}(p_j)$ and computes $d'_{\max}(p_j)$ for each $p_j \in A$ by subtracting the value $MaxDist(R_i, p_j)$ and adding $Dist(l_i, p_j)$, where $MaxDist(R_i, p_j)$ is the maximum distance between U_i's cloaked region and a POI $p_j \in A$, and $Dist(l_i, p_j)$ is the actual distance of U_i's exact location (l_i) from $p_j \in A$:

$$d'_{\max}(p_j) = d_{\max}(p_j) - MaxDist(R_i, p_j) + Dist(l_i, p_j).$$

After updating the answer set, U_i returns it to the coordinator. Then, the coordinator chooses another member and repeats the process until all members update the answer set; finally the coordinator sends the actual result to all members.

In the second case (without the coordinator), the coordinator sends the answer set with a list of unvisited users' identities to a randomly selected member. Then the selected user updates the answer set according to the above equation and passes the updated answer set to a randomly selected user among the unvisited users. After the candidate answer set has been updated by all users, the last user sends the exact result (a POI $p \in A$ with the minimum aggregate distance) to all group users.

Although Hashem et al.'s method preserves the location privacy of all members, it suffers from high computation and communication costs. The method requires the group to send n distinct nearest neighbor queries and receive a set of candidate answers. Moreover, the cloaking process requires additional communication and computation costs. In particular, computing the imprecise location requires each member to find her $k-1$ neighbors and contact them to collect their imprecise locations. Also, the LBS overhead to evaluate a group of NN queries is much higher than that of a single NN query. Furthermore, the private filtering algorithm in Hashem et al.'s method imposes additional computation and communication costs. The protocol is also vulnerable to a partial collusion attack: If the LBS colludes with the coordinator (in the first scenario) or with two users U_{i-1} and U_{i+1} (in the second scenario), then all members' locations (in the first scenario) or U_i 's exact location (in the second scenario) will be revealed. We described this attack in more detail in Section 5.2.

In comparison, the GLP protocol preserves group location privacy in a secure and efficient manner, with lower communication and computation costs. It sends a single NN query instead of n NN queries, and receives the smallest answer set: a single POI in the case of an NN query. In addition, GLP does not need to apply a private filtering algorithm because it receives only a single POI.

3. Protocol

In this section we describe the model of the proposed protocol; the protocol itself is presented in Section 3.2.

3.1. Model

The GLP protocol assumes a group of n users $\{U_1, U_2, \ldots, U_n\}$ having wireless devices with location positioning modules, such as a GPS. Users can establish Internet connections to external servers, and point-to-point connections to neighboring devices. There is an untrusted LBS provider who provides location-based services.

As the GLP protocol is based on a secure multiparty computation (SMC) [13,14], we assume an authenticated public channel for each member of the group, which is a common assumption in general secure multi-party computations [13,14]. This channel can be realized using physical means or a public bulletin board [15], where authentication can be done using digital signatures [15]. To apply digital signatures, we assume users obtain their certificates at the time of registration in the system.

Regarding the threat model, GLP considers a malicious model and allows the existence of active adversaries. Generally, there are two types of threat models: (*i*) a semi-honest model and (*ii*) a malicious model. In the semi-honest model, each participant follows the protocol specification but tries to deduce some private information about other participants; thus, only passive attackers are allowed. In the malicious model, the adversary is active and can behave arbitrarily.

Under the malicious model, GLP must satisfy three privacy requirements:

1. Preserving intragroup location privacy.
2. Preserving the first issue of intergroup location privacy.
3. Preserving privacy against active adversaries.

Intragroup location privacy protects users' location privacy from anyone inside the group; supporting this property prohibits a malicious member (insider) from learning honest members' locations. Protecting user location privacy from possible outside attackers, including the LBS, is provided by the second privacy issue above. The first two privacy issues preserve user location privacy within the group and from anyone outside the group, while the third privacy issue supports user location privacy in the case of active adversaries who may collude to jeopardize user location privacy, i.e. the collusion of malicious member with or without an LBS.

It is important to note that we consider Euclidean distance and a 2D point database server [6] for the GLP protocol. Based on this notation and the above model, we present the GLP protocol in the next subsection.

3.2. GLP protocol

The GLP protocol is based on computing a centroid as a group location descriptor and minimizing the meeting point's distance from the centroid. Computing the centroid must be done in a secure fashion, so that no information is revealed to any user except the centroid coordinate. To achieve this goal, GLP applies two building blocks: (*i*) a secure multiparty computation scheme [13] and (*ii*) a homomorphic encryption system [16]. In the SMC protocol, group members jointly and securely compute a function of their private inputs. Here, the private inputs are users' locations and the function outcome is the centroid coordinate. We use the Anonymous Veto network (AV-net) [15] and Paillier encryption [16] to design a secure centroid computation.

The AV-net protocol was developed by Hao et al. in 2006 to solve the anonymous veto problem [17]. We adopt AV-net to hide the exact locations of users during centroid computation. The original AV-net assumes a finite cyclic group G of prime order q, in which the Decision Diffie-Hellman (DDH) problem is intractable [18]. In this paper, we consider G to be a Paillier group ($Z^*_{N^2}$) in which the Computational Diffie-Hellman (CDH) problem is intractable [19]. The selection of the AV-net generator is important in this group; it cannot be the same as the Paillier generator because that generator (g_s) is a special generator ($g_s = 1 \bmod N$) that results in solving the discrete logarithm problem (DLP), and consequently the Diffie-Hellman assumption will not hold [19].

Hence, we select the AV-net generator (g) at random from group of quadratic residue in $Z^*_{N^2}$ to hold the CDH assumption. All group

GLP Protocol

Phase 1 (*AV-net values computation*):

i. U_i selects a_i, where $a_i \in_R Z_{N^2}$

ii. $U_i \rightarrow * : \left(g^{a_i}\right)$

iii. U_i computes $g^{b_i} = \prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^{n} g^{a_j}$

Phase 2 (*secure centroid computation*):

i. $U_i \rightarrow * : \left(g_s^{x_i} g^{a_i b_i}, g_s^{y_i} g^{a_i b_i}\right)$, and

$U_\alpha \rightarrow * : \left(g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N, g_s^{y_\alpha} g^{a_\alpha b_\alpha} r_2^N\right)$

ii. U_α computes $\left(g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N\right) \prod_{i=1, i\neq\alpha}^{n} \left(g_s^{x_i} g^{a_i b_i}\right)$ and

$\left(g_s^{y_\alpha} g^{a_\alpha b_\alpha} r_2^N\right) \prod_{i=1, i\neq\alpha}^{n} \left(g_s^{y_i} g^{a_i b_i}\right)$

iii. $U_\alpha \rightarrow LBS : \left(Enc_{pk}\left(\sum x\right), Enc_{pk}\left(\sum y\right), n\right) =$

$\left(g_s^{\sum x} r_1^N \bmod N^2, g_s^{\sum y} r_2^N \bmod N^2, n\right)$

$\rightarrow *$ indicates message broadcasting in the group

Fig. 1. GLP protocol.

members know the LBS public key (g_s,N) in the Paillier cryptosystem and agree on g as the AV-net generator.

The GLP protocol has the following two phases as shown in Fig. 1:

- *Phase 1*: AV-net values computation.
- *Phase 2*: Secure centroid computation.

The first phase computes the AV-net values, which will later be used as a mask to hide users' exact locations. To compute the values, each participant U_i selects at random a secret value $a_i \in_R Z_{N^2}$ and broadcasts $g^{a_i} \bmod N^2$. After finishing this phase, each party U_i computes g^{b_i} as follows:

$$g^{b_i} = \prod_{j=1}^{i-1} g^{a_j} \Big/ \prod_{j=i+1}^{n} g^{a_j} \bmod N^2.$$

In the second phase, group members collaboratively compute the centroid point (c) of their location through their corresponding AV-net masks. Specifically, each member U_i except the group representative U_α (a randomly chosen member to communicate with the LBS) masks her location through the AV-net mask and publishes $(g_s^{x_i} g^{a_i b_i}, g_s^{y_i} g^{a_i b_i})$; U_α publishes the value $(g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N, g_s^{y_\alpha} g^{a_\alpha b_\alpha} r_2^N)$.

Note that U_α is a representative group member selected randomly at the beginning of the protocol and can be different for every usage of the protocol; it is not a predetermined user.

The user U_α performs the Paillier encryption, which, upon multiplying all values, results in encrypting the summations of the x and y coordinates of all members ($g_s^{\sum x_i} r_1^N$ and $g_s^{\sum y_i} r_2^N$), as follows:

$$\left(\prod_{all\ U_i\ but\ U_\alpha} g_s^{x_i} g^{a_i b_i} \right) g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N = r_1^N \prod_i g_s^{x_i} \prod_i g^{a_i b_i} = g_s^{\sum x_i} g^{\sum a_i b_i} r_1^N$$
$$= g_s^{\sum x_i} r_1^N \bmod N^2$$

and

$$\left(\prod_{all\ U_i\ but\ U_\alpha} g_s^{y_i} g^{a_i b_i} \right) g_s^{y_\alpha} g^{a_\alpha b_\alpha} r_2^N = g_s^{\sum y_i} r_2^N \bmod N^2.$$

Since a_i and b_i are AV-net values, then $\sum a_i b_i = 0$ [15], and the results of the above equations are the summations of the x and y coordinates, which are encrypted by the LBS public key in the Paillier cryptosystem [16]. Finally, U_α sends a single NN query containing the encrypted summation coordinates and the number of group members (n) to the LBS.

Upon receiving the NN query, the LBS decrypts it using the private key, applying Eq. (1).

$$Dec(w) = m = \frac{L(w^\lambda \bmod N^2)}{L(g_s^\lambda \bmod N^2)} \bmod N, \tag{1}$$

where w is the ciphertext and m is the corresponding plaintext. Then, the LBS divides the result by n to get the centroid coordinates and executes a conventional NN query-processing algorithm to obtain the point of interest with the smallest distance from the centroid. The LBS sends the result back to U_α, who broadcasts it within the group and the protocol terminates.

The GLP protocol uses zero knowledge proofs for security from malicious participants and active adversaries. Each time a user publishes a value to the bulletin board, she must provide its zero knowledge proof. Thus, if there is any doubt about a member's honesty, members can verify the knowledge proofs and detect the malicious member(s). For this purpose, we apply Schnorr's signature [20] because it is a non-interactive zero knowledge proof; to prove the knowledge of the exponent a_i in g^{a_i}, 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^r is equal to $g^v \cdot g^{a_i h}$.

In the GLP protocol, it suffices to provide a knowledge proof only for the last message of the users. Hence, each U_i but not U_α goes through the following three steps:

1. Selects at random v, v', $v'' \in Z_{N^2}$.
2. Computes $h = H(g, t, g_s, g^v, g_s^{v'} t^v, g_s^{v''} t^v, g_s^{x_i} g^{a_i b_i}, g_s^{y_i} g^{a_i b_i}, i)$ where $t = g^{b_i}$.
3. Sends $(g^v, t^v \cdot g_s^{v'}, t^v \cdot g_s^{v''}, r = v + a_i h, r' = v' + x_i h, r'' = v'' + y_i h)$.

Verification of the proof can be done through the following three checks:

1. $g^r \stackrel{?}{=} (g^{a_i})^h g^v$
2. $g_s^{r'} t^r \stackrel{?}{=} \left(g_s^{x_i} g^{a_i b_i}\right)^h g_s^{v'} t^v$
3. $g_s^{r''} t^r \stackrel{?}{=} \left(g_s^{y_i} g^{a_i b_i}\right)^h g_s^{v''} t^v$

The first expression verifies the knowledge proof of a_i, while the next two expressions verify the knowledge proofs of x_i and y_i, respectively.

U_α publishes a different value, so her zero knowledge proof will be different; she must demonstrate the knowledge proof of exponents a_α, x_α and y_α, and the N'th root of r_1^N and r_2^N. To do this, we combine Schnorr's signature [20] with Damgard and Jurik's knowledge of the N 'th power protocol [21]. In their protocol, to prove that r^N is an N 'th power, the prover sends $\{v^N, z = v r^h\}$, where $v \in_R Z_{N^2}$ and $h = H(N, r^N, v^N)$. To verify this proof, one can check whether z^N is equal to $v^N r^{Nh}$.

Providing zero knowledge proofs for values a_i and $a_i \in_R Z_{n^2}$, U_α does the following:

1. Selects at random v, v', v'', v_1, $v_2 \in Z_{N^2}$.
2. Computes $h = H(g, N, t, g_s, g_s^{v'}, t^v v_2^N, g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N, g_s^{y_\alpha} g^{a_\alpha b_\alpha} r_2^N, \alpha)$ where $t = g^{b_i}$.
3. Sends $(g^v, g_s^{v'} t^v v_1^N, g_s^{v''} t^v v_2^N,\ r = v + a_\alpha h,\ r' = v' + x_\alpha h,\ r'' = v'' + y_\alpha h,\ z_1 = v_1 r_1^h,\ z_2 = v_2 r_2^h)$.

This proof can be verified by the following three checks:

1. $g^r \stackrel{?}{=} (g^{a_\alpha})^h g^v$
2. $g_s^{r'} t^r z_1^N \stackrel{?}{=} \left(g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N\right)^h g_s^{v'} t^v v_1^N$
3. $g_s^{r''} t^r z_2^N \stackrel{?}{=} \left(g_s^{y_\alpha} g^{a_\alpha b_\alpha}\right)^h g_s^{v''} t^v v_2^N$

As before, the first expression verifies the knowledge proof of a_α and the second (third) expression verifies the knowledge proof of $x_\alpha(y_\alpha)$ and the proof of the N 'th power of $r_1^N(r_2^N)$. The GLP protocol only requires a proof of knowledge for the second phase; as the proof of exponent a_i is already included, there is no need to provide a separate proof for the first phase.

4. Security and efficiency analysis

As mentioned in Section 3.1, the GLP protocol must satisfy three security requirements:

1. Preserving intragroup location privacy
2. Preserving the first issue of intergroup location privacy
3. Preserving privacy in the case of active adversaries

In this section, we first analyze the protocol behavior against active adversaries (third issue) and then in Section 4.2 we investigate

the intergroup and intragroup privacy properties of the GLP. Finally, we discuss the efficiency analysis of the protocol in Section 4.3.

4.1. Resistance against active adversaries

A malicious member can try to cause a disruption attack, which prevents the protocol from fulfilling its task. She can also try to send fake values, i.e., she can modify her AV-net masks to prevent the protocol from achieving its goal. Moreover, malicious parties may collude to violate honest members' privacy. We consider these misbehaviors and analyze how the protocol can overcome them.

A disruption attack can be done by publishing a fake value or by modifying the AV-net masks, i.e., a malicious member may use a fake value of b_i instead of the correct one as $\sum a_i b_i \neq 0$. With our use of the zero knowledge proof, no one can do this; the attackers must know the discrete logarithm of the fake value to provide a valid proof [15]. Even if the malicious party uses a known value such as g^{c_i} and publishes a zero knowledge proof for c_i, the verification will fail because each party knows value g^{b_i} and uses that value in the zero knowledge verification. The attacker will thus be detected during verification and will be excluded; the protocol then restarts.

In a collusion attack, active attackers may collude to discover the location of a group member. There are two types of collusions: *(i)* full collusion and *(ii)* partial collusion. In a full collusion attack, all participants collude against one user; in a partial collusion some participants, but not all, collude against one user.

In the case of full collusion on the GLP, the attackers are able to remove the mask [15] and obtain the location coordinates. However, the GLP protocol is resistant against partial collusion. The private value of each party in the GLP protocol is her location coordinates (x_i, y_i). To reveal the location coordinate of U_i, the attackers (or the LBS) must remove the AV-net mask from the message ($g^{x_i} g^{a_i b_i}$). To remove the mask, the attackers need to learn b_i, but the AV-net protocol guarantees the secrecy of b_i if partial collusion occurs [15]. Since the colluding parties (including LBS) can determine no information about b_i, they fail to discover the location coordinates of U_i.

4.2. Privacy properties

The GLP protocol has two privacy goals: (*i*) intragroup location privacy and (*ii*) the first issue of intergroup location privacy. Recall that the intragroup location privacy supports user location privacy inside the group; the first issue of the intergroup location privacy preserves user location privacy from anyone outside the group. The proposed protocol preserves intragroup location privacy in a partial collusion attack. As already mentioned, if a malicious member tries to discover the location coordinate of another member, she must cancel the AV-net mask, but with a partial collusion, the attackers cannot obtain b_i; consequently, intragroup location privacy will be totally preserved. Moreover, due to the randomness of the AV-net masks, the GLP protocol protects user location privacy in consecutive usages of the protocol with static users.

Regarding intergroup location privacy, GLP provides user location privacy from anyone outside the group, including the LBS; the LBS only learns the centroid coordinates. Moreover, even if the LBS is able to eavesdrop on the group's internal communication, it cannot obtain useful information about members' locations. Discovering the location coordinates requires the LBS to cancel the AV-net masks, and as explained earlier, the LBS cannot obtain b_i in the case of a partial collusion. The situation is the same or even harder for outside attackers, because in addition to not knowing the AV-net values, they have no information about the LBS's private key.

Further, the secrecy of the centroid coordinate is provided for the GLP protocol because the outcome of the group computation is the encryption of the centroid with the LBS public key. Therefore, eavesdropping on internal communication by an outside attacker would not reveal the centroid location.

Because the LBS returns the exact result of the NN query, the LBS and possible outside attackers will learn the location of the meeting place; hence, meeting place location privacy is not provided for GLP. To fulfill this need, one could use location cloaking techniques [4]. In particular, before sending the query, U_α would employ a location cloaking technique to cloak the centroid location into a region and then send a single NN query along with the centroid cloaked region. In this case, the service provider would not be able to identify the exact location of the centroid and would thus return a set of candidate answers without knowing the exact result (the meeting place).

4.3. Efficiency analysis

In terms of computation cost, we count the number of exponentiation operations. The cost of multiplication operations is ignored because they are negligible compared to the exponentiation cost. In Phase 1 of GLP, each party performs a single exponentiation to encrypt her random secret; this can be done offline. Computing g^{b_i} incurs a negligible cost, as explained in [22]. In phase 2 of GLP, each party must perform one exponentiation to compute $g^{a_i b_i}$. The computation cost of $g_s^{x_i}$ (and $g_s^{y_i}$) is negligible, since the exponent is a geographical coordinate, usually an integer between six- or seven-decimal digits and at most 32 bits long. As a result, based on the simultaneous multiple exponentiation algorithm [23], the computation of $g_s^{x_\alpha} g^{a_\alpha b_\alpha} r_1^N$ requires one exponentiation, and in total, GLP costs two exponentiations per user.

The GLP protocol uses a zero knowledge proof of the discrete logarithms in the case of a dispute. We use an efficient knowledge proof [20] system to decrease the cost of the zero knowledge operations. It is important to note that any SMC protocol needs a zero knowledge proof system to be secure against active adversaries [15]; thus, its cost is unavoidable.

There are several factors that affect communication cost: the number of intragroup messages exchanged, the number of queries sent to the LBS and the size of the LBS result.

In each phase of the GLP protocol, each participant broadcasts one message to the group containing a full-length integer; this results in each user sending two full-length integers in total. If we count the total number of broadcasting messages, we get $n + n = 2n$, where n is the number of participants. Thus, the number of intragroup messages exchanged is of complexity $O(n)$. Moreover, the GLP protocol only sends a single NN query to the LBS and receives a minimal answer (a single POI) from the LBS. Thus, the total communication complexity of the GLP is $O(n)$.

It is worth mentioning that U_α, as the group representative, is not a single point of failure because she is not a member with any special capability, and every member in the group can act as the group representative. Moreover, if U_α fails during a protocol run, group members expel U_α and restart the protocol (choosing another representative member randomly to communicate with the LBS) without violating their privacy.

5. Comparison

This section compares the GLP protocol with previous work. To the best of our knowledge, Hashem et al. [7] have conducted the only other work in the field of group location privacy. We compare GLP with Hashem et al.'s protocol and summarize the result in Table 1.

Table 1
Comparison of GLP with Hashem et al. method.

	Intragroup privacy	Intergroup privacy	Number of intragroup messages	Number of sending query points	Size of the answer set	Resistance against partial collusion attack
Hashem, 2010	Yes	Yes	$O(nm)$	n	$O(l)$	No
GLP protocol	Yes	Yes	$O(n)$	1	$O(1)$	Yes

Unlike GLP, Hashem et al.'s protocol is vulnerable to a partial collusion attack. We describe this attack below. Recall from Section 2 that the second phase of Hashem et al.'s protocol can be conducted through two scenarios: with or without the coordinator. In the first scenario, for each user U_i, the coordinator knows the value of $d_{\max}(p_j)$ before U_i updates it (we denote it as $d_{\max}{}^{(i)}(p_j)$) and after the update (we denote that as $d'_{\max}{}^{(i)}(p_j)$), for each $p_j \in A$. In the second scenario, without the coordinator, the immediate predecessor and immediate successor of each user know these values. The LBS knows the set of the users' anonymous cloaked regions and the maximum distance ($MaxDist(R_i,p_j)$) of each cloaked region R_i from each $p_j \in A$. Here we show that a collusion of the LBS and the coordinator in the first scenario reveals all members' exact locations. The colluding parties perform the following steps to find the exact location of a user U_i:

1. They choose a cloaked region R among the set of cloaked regions, and assume it is U_i 's cloaked region.
2. For each $p_j \in A$, the coordinator computes $\Delta_{ij} = d^{(i)}_{\max}(p_j) - d'_{\max}{}^{(i)}(p_j)$ and, with the help of the LBS who knows R, computes $dist(l_i, p_j) = MaxDist(R,p_j) - \Delta_{ij}$.
3. For each $p_j \in A$, they draw a circle with center p_j and radius $dist(l_i,p_j)$.
4. If the circles drawn around p_j for all $p_j \in A$ do not intersect at a single point in R, then R cannot be the actual cloaked region of U_i. Otherwise, most probably it is the actual cloaked region of U_i, and the intersection point is most probably the exact location of U_i, given that there are a relatively large number of POIs returned by the LBS. Hence, the actual cloaked region of U_i, and consequently her exact location, can easily be determined.

Fig. 2 illustrates the attack for a typical user U_i and three POIs when the attackers assign the right cloaked region to U_i.

This attack is also executable in the scenario without the coordinator: A collusion of the LBS, U_{i-1} and U_{i+1} reveals U_i 's exact location. In this scenario, for each user U_i,U_{i-1}(U_i 's predecessor) knows the value of $d^{(i)}_{\max}(p_j)$ and U_{i+1} (U_i 's successor) knows the value of $d'_{\max}{}^{(i)}(p_j)$. Then, U_{i-1} and U_{i+1} act as the coordinator in the above mentioned attack, so with the help of the LBS, U_i 's exact location will be revealed.

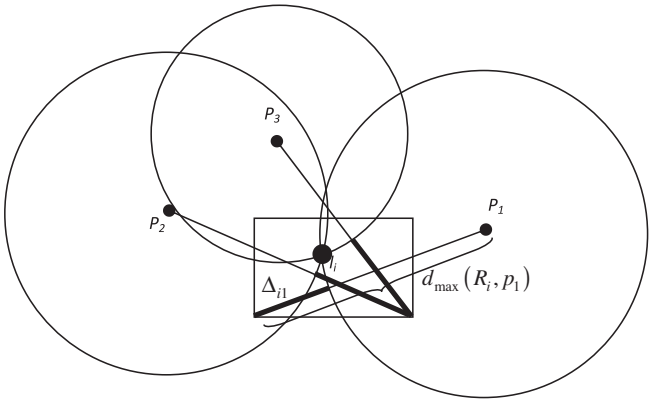

Fig. 2. Illustration of a partial collusion attack with three POIs in Hashem et al.'s protocol; thick lines show Δ_{ij}.

In a targeted attack, two colluding members can frame a particular user U_i: in this attack, colluding members can become the immediate predecessor (U_{i-1}) and immediate successor (U_{i+1}) of U_i through tampering with the list of unvisited users. Specifically, when one of the colluding parties receives the answer set and the list of unvisited users, she can cheat by modifying the list of unvisited users such that only U_i and her partner in the attack (who will become U_{i+1}) remain unvisited and then she passes the answer set to U_i. In this situation, U_i, not suspecting anything, forwards her updated answer set to the next attacker; the attackers obtain the required knowledge and the attack takes place.

Based on the above discussion, we can say that Hashem et al.'s protocol does not protect user location privacy in a partial collusion attack.

Regarding the communication cost, the number of messages exchanged between peers in Hashem et al.'s protocol is of complexity $O(nm)$, where m is the number of response messages that have been received by each participant from her peers.

It is important to note that in the first phase of Hashem et al.'s protocol [7], each user blurs her exact location based on her peers' imprecise locations [12]. To achieve this, each user sends a request to her peers and receives their imprecise locations; she then computes her cloaked region as the MBR that includes her exact location and her peers' imprecise locations. Thus, the total number of exchanged messages would be $O(nm)$. In contrast, the GLP protocol does not require users to compute their cloaked locations; users privately compute the group location descriptor (centroid), which needs $O(n)$ messages to exchange.

The GLP protocol is resource-aware as it only sends a single NN query to the LBS; Hashem et al.'s protocol sends n distinct queries. As a result, not only the communication cost but also the LBS overhead to evaluate the query is significantly decreased in GLP. Specifically, processing a group of NN queries requires the LBS to evaluate each POI against a set of cloaked regions, which results in increasing the overhead of the LBS.

The final factor that affects the communication cost is the size of the LBS result. In Hashem et al.'s protocol, the LBS returns a set of candidate POIs along with the minimum and maximum aggregate distances. Consequently, the size of the LBS result would be $O(l)$, versus $O(1)$ in GLP, where l is the number of POIs in the set of candidate POIs. Hence, considering each factor described above, Hashem et al.'s protocol results in a higher communication cost than GLP.

Having a small-sized result set protects the LBS from excessive disclosure and supports LBS content privacy. The GLP protocol supports this property because it provides the minimum possible size for the result set; a large number of POIs is disclosed in Hashem et al.'s method.

In summary, the GLP protocol outperforms the previous work in terms of efficiency and security. It preserves group location privacy with a lower communication cost and a higher level of security in a malicious model. It also prevents the disclosure of users' locations in a partial collusion attack.

6. Conclusion

In this paper we have considered the problem of supporting location privacy for a group of users who wants to benefit from

an LBS and find the nearest meeting place that minimizes their aggregate distance. We have identified the privacy issues for this group scenario and proposed a resource-aware solution – the GLP protocol – to address them. The proposed protocol, which is based on the AV-net protocol and homomorphic encryption, protects the location of all group members from an attacker inside and outside the group. The GLP protocol decreases the number of group queries and the size of the LBS result and also protects the LBS from excessive disclosure of its content. The performance and security analysis of the GLP protocol have been investigated and it has been shown that GLP is secure against partial collusion attacks in a malicious model.

Acknowledgments

This work is partly supported by the Ministry of Information and Communications Technology (1615-500-T) of the Islamic Republic of Iran. The authors would like to thank Emre Yilmaz for his feedback.

References

[1] H. Hu, J. Xu, Non-exposure location anonymity, in: Proceedings of IEEE 25th International Conference on Data Engineering (ICDE), 2009.
[2] M.F. Mokbel, C.Y. Chow, W.G. Aref, The new casper: query processing for location services without compromising privacy, in: Proceedings of VLDB, 2006.
[3] P. Kalnis, G. Ghinita, K. Mouratidis, D. Papadias, Preserving location-based identity inference in anonymous spatial queries, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 1719–1733.
[4] A. Solanas, J. Domingo-Ferrer, A. Martínez-Ballesté, Location privacy in location-based services: beyond TTP-based schemes, in: 1st International Workshop on Privacy in Location-Based Applications (PILBA 2008) within 13th European Symposium on Research in Computer Security (ESORICS), 2008.
[5] M. Strassman, C. Collier, Case study: The development of the find friends application, in: Location-based Services, 2004, pp. 27–40.
[6] D. Papadias, Y. Tao Y, K. Mouratidis K, C.K. Hui, Aggregate nearest neighbor queries in spatial databases, ACM Transactions on Database Systems 30 (2) (2005) 529–576.
[7] T. Hashem, L. Kulik, R. Zhang, Privacy preserving group nearest neighbor queries, in: ACM International Conference Proceeding Series, vol. 426, 13th International Conference on Extending Database Technology (EDBT), 2010.
[8] C.Y. Chow, M.F. Mokbel, X. Liu, A peer-to-peer spatial cloaking algorithm for anonymous location-based services, in: GIS '06: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, 2006.
[9] G. Ghinita, P. Kalnis, S. Skiadopoulos, PRIVÉ: anonymous location-based queries in distributed mobile systems, in: WWW, 2007.
[10] G. Ghinita, P. Kalnis, S. Skiadopoulos. MobiHide: a mobile peer-to-peer system for anonymous location-based queries, in: Proceedings of the International Symposium on Spatial and Temporal Databases, 2007.
[11] A. Solanas, A. Martínez-Ballesté, A TTP-free protocol for location privacy in location-based services, Computer Communications Journal 31 (2008) 1181–1191.
[12] T. Hashem, l. Kulik, Safeguarding location privacy in wireless ad-hoc networks, in: Ubicomp, 2007.
[13] A. Yao, How to generate and exchange secrets, in: Proceedings of the Twenty-seventh Annual IEEE Symposium on Foundations of Computer Science, 1986.
[14] O. Goldreich, S. Micali, A. Wigderson, How to play any mental game or a completeness theorem for protocols with honest majority, in: Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing, 1987.
[15] F. Hao, P. Zielinski, The power of anonymous veto in public discussion, in: Transactions on Computational Science IV, LNCS, vol. 5430, 2009, pp. 41–52.
[16] P. Paillier, D. Pointcheval, Efficient public-key cryptosystems provably secure against active adversaries, Advances in Cryptology (ASIACRYPT) (1999).
[17] F. Hao, P. Zielinski, A 2-round anonymous veto protocol, in: Proceedings of the 14th International Workshop on Security Protocols, 2006.
[18] F. Hao, P.Y.A. Ryan, Password authenticated key exchange by juggling, in: Proceedings of the 16th International Workshop on Security Protocols, 2008.
[19] E. Bresson, D. Catalano, D. Pointcheval, A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and its Applications, in: Proceedings of ASIACRYPT, LNCS, vol. 2894, 2003, pp. 37–54.
[20] C.P. Schnorr, Efficient signature generation by smart cards, Journal of Cryptology 4 (3) (1991) 161–174.
[21] I. Damgard, M. Jurik, A generalisation, a simplification and some applications of Paillier's probabilistic public-key system, in: Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography (PKC), 2001.
[22] F. Hao, M.N. Kreeger, Every vote counts: ensuring integrity in DRE-based voting system, Technical Report No. 1268, Newcastle University, August 2011. Available online <www.cs.ncl.ac.uk/publications/trs/papers/1268.pdf>.
[23] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1996. <www.cacr.math.uwaterloo.ca/hac>.