Knowl Inf Syst (2015) 45:589–615
DOI 10.1007/s10115-014-0809-0

REGULAR PAPER

The Cloaked-Centroid protocol: location privacy protection for a group of users of location-based services

Maede Ashouri-Talouki · Ahmad Baraani-Dastjerdi · Ali Aydın Selçuk

Received: 2 June 2011 / Revised: 28 December 2011 / Accepted: 17 August 2012 /
Published online: 2 December 2014

Abstract Several techniques have been recently proposed to protect user location privacy while accessing location-based services (LBSs). However, applying these techniques to protect location privacy for a group of users would lead to user privacy leakage and query inefficiency. In this paper, we propose a two-phase protocol, we name Cloaked-Centroid, which is designed specifically to protect location privacy for a group of users. We identify location privacy issues for a group of users who may ask an LBS for a meeting place that is closest to the group centroid. Our protocol relies on spatial cloaking, an anonymous veto network and a conference key establishment protocol. In the first phase, member locations are cloaked into a single region based on their privacy profiles, and then, a single query is submitted to an LBS. In the second phase, a special secure multiparty computation extracts the meeting point result from the received answer set. Our protocol is resource aware, taking into account the LBS overhead and the communication cost, i.e., the number of nearest neighbor queries sent to a service provider and the number of returned points of interests. Regarding privacy, Cloaked-Centroid protects the location privacy of each group member from those in the group and from anyone outside the group, including the LBS. Moreover, our protocol provides *result-set anonymity*, which prevents LBS providers and other possible attackers from learning the meeting place location. Extensive experiments show that the proposed protocol is efficient in terms of computation and communication costs. A security analysis shows the resistance of the protocol against collusion, disruption and background knowledge attacks in a malicious model.

M. Ashouri-Talouki (✉)
Department of IT Engineering, Faculty of Computer Engineering, The University of Isfahan, Isfahan, Iran
e-mail: m.ashouri@eng.ui.ac.ir

A. Baraani-Dastjerdi
Department of Software Engineering, Faculty of Computer Engineering, The University of Isfahan, Isfahan, Iran

A. A. Selçuk
Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey

Keywords Location privacy · Group privacy · Location-based services · Secure multiparty computation

1 Introduction

Location-based services (LBSs) provide a wide range of capabilities to mobile users, such as traffic report services, transportation services, nearby friend or nearby store services, advertising and emergency control services [12]. These services deliver desired information based on the users' private information [26]. Mobile users can ask location-dependent queries of the spatial database [61] and receive information based on their locations at any time and from anywhere [61]. These services can be invoked by a single user or by a group of users [57]. For example, one user could ask "Where is the nearest restaurant to my location?" or a group of users could ask "Where is the nearest meeting place to the group centroid?".

Since LBSs offer their benefits based on the exact location of a user or a group of users, location privacy concerns are raised. 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 affiliation(s) [23,26]. To tackle such privacy concerns, current research efforts focus on proposing techniques that preserve user location privacy during the use of LBSs. Although there exists a large amount of the literature for preserving the location privacy of an individual user [3,10,13,15,19–21,23,24,27,34–38,55,58,61,62,64], supporting location privacy for a group of users has not been much explored.

Consider a scenario in which a military group of users wishes to have a critical meeting in a place that is closest to the group centroid. They can utilize a LBS provider that maintains a database (P) of points of interest (POIs) [47]. To get the desired POI, users of the group provide their current locations (called query points) to the LBS; then, the LBS returns the point(s) of P with the smallest distance(s) from the centroid of query point.

There are two major privacy concerns in this scenario:

(i) Preserving the location privacy of each group member and
(ii) Preserving the location privacy of the meeting place.

The first issue encompasses protection of user location information from other group members, as well as from the LBS and outside attackers. The second privacy issue deals with hiding the meeting point location from anyone outside the group, including the LBS and outside attackers.

Considering these two privacy issues, we can see the problem as an instance of a secure multiparty computation (SMC), in which group members jointly and securely compute a function of their private inputs (their locations) such that the function outcome is the meeting place location. Furthermore, not only users' private inputs but also the result of the computation (meeting place location) must be kept secret. In other words, the result of the computation can only be visible to the group members.

The focus of group location privacy is on protecting location privacy for all group members; individual location privacy aims to protect single-user location privacy. Further, preserving the location privacy of a requested place in a single-user scenario is straightforward, but this is more complicated in a group scenario. For these reasons, the techniques of the former cannot be directly applied to the latter; special solutions must be developed to achieve group location privacy.

To the best of our knowledge, Hashem's research [31] and the GLP protocol [2] are the only works addressing the location privacy problem for a group of users during the use of

LBSs. In Hashem's method, each member sends her imprecise location to the LBS; then, the LBS returns a set of candidate POIs with respect to the members' imprecise locations. To determine the actual answer, group members execute a private filtering algorithm that finds the exact result from the candidate answer set without violating members location privacy.

Although Hashem's work preserves the location privacy of group members, it is an expensive method in terms of communication cost because it requires each member to send her imprecise location (a cloaked region) to the LBS and the LBS to return a set of candidate POIs that must be jointly refined by the group members to determine the exact result.

In GLP protocol, group members jointly and securely compute the centroid point of their locations and send it to the LBS. Then, the LBS returns the nearest meeting point to the centroid. GLP protocol does not need any computation to determine the actual answer, because the answer set only contains the exact result. The drawback of this approach is that GLP protocol does not support the location privacy of the meeting place [2].

In this paper, we propose a resource-aware protocol we name Cloaked-Centroid that provides member location privacy and meeting place location privacy. The proposed protocol relies on spatial cloaking, an AV-net scheme and a conference key establishment protocol and is resistant against collusion attacks, disruption attacks and background knowledge attacks. Furthermore, the Cloaked-Centroid protocol offers a location cloaking process with personalized privacy requirements for each group member. Moreover, the Cloaked-Centroid protocol is completely independent of how the LBS evaluates the queries; thus, it can be seamlessly integrated with any existing privacy-aware query-processing algorithm [11,33,43].

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

1. We propose a location privacy protection technique (Cloaked-Centroid) for a group of users that meets the privacy requirements of group members and the meeting place. Specifically, our protocol supports the result-set anonymity property.
2. The proposed protocol provides a location cloaking process based on personalized user privacy requirements, specifically minimum area $A_{i,\min}$, i.e., user u_i would like to blur her exact location into a region with an area size of at least $A_{i,\min}$.
3. We provide the proof of correctness of Cloaked-Centroid protocol and analyze its privacy and security properties. In particular, we show that our protocol is secure against collusion attacks, disruption attacks and background knowledge attacks in a malicious model.
4. We evaluate the performance of the protocol through extensive experiments. The results show that Cloaked-Centroid protocol is efficient and scalable while preserving the privacy requirement of group members and meeting place.

The rest of the paper is organized as follows. The next section reviews the existing works in the field of location privacy. Section 3 delineates our system model and the assumption of our study. In Sect. 4, the preliminaries of our solution are explained. Section 5 presents the proposed protocol and its proof of correctness. In Sects. 6 and 7, we describe our privacy analysis and security analysis of the Cloaked-Centroid protocol, respectively. The experimental results are shown in Sect. 8, along with the comparison of the previous work, and finally, the paper is concluded in Sect. 9.

2 Related works

There is a wide literature on preserving user location privacy during the use of LBSs [11, 14,15,21,30–35,43,55,56]. A large portion of location privacy mechanisms are based on k-anonymity techniques, which are borrowed from databases [51] and privacy-preserving data mining field [17,53,59,60].

Generally, location privacy mechanisms are classified into two main categories [55]: (1) schemes that rely on trusted third parties (TTP-based) and (2) methods that are not based on TTPs (TTP-free).

The Casper framework [43] is a TTP-based method presented by Mokbel et al. that consists of two main components: the anonymizer and the privacy-aware query processor. The anonymizer uses a grid-based pyramid structure [43] and blurs a user location to a cloaked region that contains at least *k* users, including the initial user (*k* is a user-specified parameter defined in her privacy profile). The privacy-aware query processor is embedded in the LBS provider and processes location-based queries.

Proposed by Kalnis et al. [33], the nearest neighbor cloak and the Hilbert cloak are two other TTP-based methods that blur an exact location to a cloaked region containing *k* users. Moreover, the authors address the issue of privacy-aware query processing at the LBS and develop an algorithm for it. It is worth mentioning that our paper does not aim to propose another privacy-aware query processor; rather, it addresses the problem of protecting location privacy for a group of users when accessing an LBS. Thus, any existing privacy-aware query-processing algorithm embedded in the LBS provider can be employed [11,33,43].

Although TTP-based methods provide a good balance between efficiency, security and accuracy, there is problem with all of these methods: users must trust the TTP and disclose their exact location to it. To overcome these problems, TTP-free methods have been proposed [55]. Two important classes of methods of this category are as follows: (1) collaboration-based methods [14,32,56] and (2) obfuscation-based methods [1,19]. In a collaboration-based method, a mobile user blurs her exact location by forming a group of her peers. Obfuscation-based methods preserve location privacy by artificially perturbing location information [1].

In this paper, we only consider solutions that protect user location privacy through group formation because they are similar to the group location privacy paradigm. After discussing these solutions, we specifically focus on the approaches that support location privacy for a group of users [2,31].

Chow et al. [14] were the first to apply the group formation technique to cloak single users' locations. In Chow's method, the mobile user forms a group of her peers by contacting them via single-hop or multi-hop communication. Then, the mobile user can blur her exact location into a spatial cloaked region that covers the entire group of peers. In the group formation phase, a query requester broadcasts a FORM_GROUP request to the neighboring peers. Because her peers respond to the FORM_GROUP request with their IDs and locations, the requester learns the locations of her peers. This factor is a drawback to Chow's approach that is not addressed in his later work [15]. Another drawback of Chow's method is that the user tends to be close to the center of her special cloak. Although this bug is repaired in Chow's later work [15], the first problem still exists.

PRIVE [22] and MOBIHIDE [21] are two consecutive approaches presented by Ghinita et al. They proposed these two distributed methods to preserve the anonymity of a user issuing spatial queries to an LBS. Both methods are based on the Hilbert space-filling curve and assume that a user trusts her peers. In PRIVE, users are grouped into fixed hierarchical partitions (clusters) based on their Hilbert value. Each cluster head is responsible for determining the cloaked region of users in her cluster; therefore, the load of the head node in each cluster may be very high. In contrast, MOBIHIDE does not organizing users into fixed partitions, so it is more efficient. The mobile user will construct an index of other user location through a Chord-based distributed hash table and then anonymize her location by mapping the location to a random group of *k* consecutive users in the hash table.

Solanas et al. [55] proposed a cryptographic-based method to preserve single-user location privacy. A mobile user contacts the peers in her cover range to learn their locations; then, a centroid point is computed by the mobile user as her fake location. The locations are masked by adding Gaussian noise with zero mean to allow users to freely share their location without trusting their peers. However, if this procedure is applied several times with static users, their location will be disclosed due to the cancelation of Gaussian noise. To solve this drawback, Solanas [56] applied a public key privacy homomorphism; each user encrypts her masked location with an LBS public key and then shares the result with her peers.

Although applying privacy homomorphism solves this drawback, there is another problem with Solanas method: If the LBS were able to eavesdrop on users' internal communication, then in consecutive usages with static users, the LBS would be able to deduce their exact locations due to the noise cancelation.

More similar to our protocol, Hu's method [32] preserves individual user location privacy by forming a group without the user trusting her peers. In general, Hu's method consists of two phases. In Phase one, the mobile user identifies her k peers through proximity information; in phase two, the minimum bounding rectangle (MBR) of the set of users is constructed through a specialized secure multiparty protocol. Alleviating the need for peer trust, this is a solution for single-user location privacy, and, similar to other such solutions, does not need extra phases to determine the exact POI from the received set of POIs (such as the answer-refining phase of Hashem's protocol [31]).

It is worth mentioning that refining the answer set in all individual scenarios is done by the query requester or by the query anonymizer (a trusted third party that mediates communication and performs the cloaking and anonymizing processes [33,43]). In our proposed protocol, there is no anonymizer and users do not need to trust their peers; they refine the answer set to determine the exact result in a secure manner.

In our Cloaked-Centroid protocol, if the LBS eavesdrops on internal communication, it learns no information about users' exact locations, even with static users. As there is no need for an encryption scheme, Cloaked-Centroid is a lightweight method in terms of computation and communication costs.

As mentioned above, Hashem's [31] and GLP [2] are the sole works in the field of group location privacy. In Hashem's work, there are two phases, similar to our Cloaked-Centroid protocol. Hashem's first phase, which is responsible for location cloaking, blurs the exact location of each user based on her peers' local imprecise locations [30]. Afterward, each user submits her cloaked location along with a query ID to the LBS. (Query IDs are issued by a group coordinator, which is responsible for managing the group and submitting the parameters of nearest neighbor (NN) queries to the LBS [31]). Upon receiving all requests, the LBS provider evaluates the received query with respect to a set of cloaked regions and returns a set of candidate POIs, A, along with their total maximum and minimum distances from the users' cloaked regions.

Hashem' second phase, called the answer-refining phase, determines the exact POI without revealing the users' exact locations. Sequentially, each member updates the total maximum and minimum distances of each POI in A with her actual distance; then, the point with the minimum total distance is selected as the meeting place.

Although Hashem's work preserves location privacy for each group user, it does not support meeting place location privacy. In particular, although Hashem's work preserves result-set anonymity, the location of the meeting place can be learned by any outside attacker, including the LBS.

Furthermore, this method requires the group to send n distinct NN queries, which imposes a high communication cost. Moreover, computing an imprecise location requires each member to find her $k - 1$ peers and contact them to collect their local imprecise locations [30]. Thus, the cloaking process requires additional communication and computation costs. Additionally, the LBS overhead to evaluate a group of NN queries is much higher than for that of a single NN query because the LBS evaluates each POI against a set of regions, rather than against a single region.

The GLP protocol [2] contains only one phase that computes the centroid point of group members. In particular, each member publishes her masked location, and then, a specific member computes the encrypted centroid point of the published locations using Paillier encryption [46]. Afterward, the encrypted centroid is sent to the LBS; the LBS then decrypts it and returns the meeting place nearest to the centroid. Although this approach preserves members' location privacy, it does not protect the location privacy of the meeting place.

Our Cloaked-Centroid protocol submits a single NN query along with the Cloaked-Centroid region to the LBS and receives the answer set; then, it privately determines the exact result from the answer set in a distributed manner while ensuring exact result privacy. Further, it achieves its security and privacy goals with a lower computation and communication costs. Moreover, as Cloaked-Centroid is completely independent from how LBS providers process and evaluate location-based queries, any existing query-processing algorithm with respect to a cloaked region, e.g., [11,33,43] can be employed to evaluate location-based queries; our protocol can be seamlessly integrated with them.

3 System model

In this section, we present the assumptions made in our protocol and formally define the general problem of our study.

We assume that there is a group of users having wireless devices with location positioning modules, such as a GPS. These devices can establish Internet connections to external servers and point-to-point connections to neighboring devices.

We consider a malicious model as the protocol threat model and allow the existence of active adversaries. Generally, there are two types of threat models: (i) a semi-honest model and (ii) a malicious model. In a semi-honest model, each participant follows the protocol specification but tries to deduce some private information of the other participants; this model only allows for passive attackers. In a malicious model, the adversary is active and can behave arbitrarily.

We assume an authenticated public channel for each member of the group, which is an essential requirement for general secure multiparty computations [25,28]. This channel can be realized using physical means or a public bulletin board [36], where authentication can be done using digital signatures [36,52] or symmetric shared keys [41,49,52].

In addition, we assume a group membership key, which is a secret shared key known only to members and distributed by the group manager (the member who initiate the group). Notice that the group manager registers the group members and distributes the group membership key among them.

We assume Euclidean distance and a 2D point database server for Cloaked-Centroid protocol.

The proposed protocol assumes slow-moving users, but it is important to mention that, with caution, the Cloaked-Centroid can also be used for fast-moving users. In such a situation, distances change rapidly and thus also will the meeting point. We will give some general information about this situation in Sect. 9 but leave the details for a future work.

Based on the above assumptions, the general problem of the paper can be formally stated as follows:

Given a set of POIs P, a set of active attackers E and a set of users $U = \{u_1, u_2, \ldots, u_n\}$ with their precise locations $L = \{l_1, l_2, \ldots, l_n\}$, we want to design a protocol that outputs a data point $p \in P$ such that for any point $p' \in P$, dist $(p, c) \leq \text{dist}(p', c)$, where c is the centroid of U. The protocol should output p, while the precise location l_i of a user u_i is only visible to u_i, and the centroid c and meeting point p are only visible to U even in the presence of active attackers.

4 Preliminaries

In this section, we present the main building blocks used in designing the Cloaked-Centroid protocol: an AV-net scheme [29] and the Burmester–Desmedt conference key establishment protocol [6,7,49]. We use the AV-net scheme to mask users' locations such that the masks vanish upon aggregation. The Burmester–Desmedt conference key establishment protocol is used to hide the result of the protocol from anyone outside the group. Through these methods, Cloaked-Centroid provides member location and meeting point location privacy. In both building blocks, and consequently in our protocol, it is assumed that G is a finite cyclic group of prime order q in which the Decisional Diffie–Hellman (DDH) problem is intractable. The generator in G is g, and all computations take place in G. There are n members in the group as $\{u_1, u_2, \ldots, u_n\}$, and they agree on $(G; g)$.

4.1 AV-net protocol

AV-net [28] was developed by Hao in 2006 to solve the anonymous veto problem and consists of two rounds. In the first round, each member produces and broadcasts a random ephemeral public key g^{a_i}. Then, each member computes g^{a_i} by multiplying all the random ephemeral public keys before i and dividing all the random ephemeral public keys after i:

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

In the second round, each member broadcasts $g^{c_i b_i}$ or $g^{a_i b_i}$, depending on whether the user vetoes or not, respectively (c_i is a random number). Upon multiplying all messages, if no one vetoes, we have $\prod_i g^{a_i b_i} = 1$ because of the vanishing property of AV-net exponents $\left(\sum_i a_i b_i = 0\right)$ [29]; if one or more participants veto(es), we have $\prod_i g^{c_i b_i} \neq 1$, while the vetoing user(s) remain(s) anonymous [29].

4.2 Burmester–Desmedt protocol

The second building block of Cloaked-Centroid is the conference key establishment protocol. Many such protocols have been presented in the literature [6]; of those, we apply a broadcast version of the protocol proposed by Burmester and Desmedt [7], which we adequately integrate with the AV-net rounds. The Burmester–Desmedt protocol has two major phases. In the first phase [7], each member u_i computes and broadcasts a random number g^{e_i}. In the second phase, each u_i broadcasts $t_i = (g^{e_{i+1}} / g^{e_{i-1}})^{e_i}$, which is used to construct the conference key by the following equation:

$$k_i = \left(g^{e_{i-1}}\right)^{n.e_i} \cdot t_i^{n-1} \cdot t_{i+1}^{n-2} \cdots t_{i-2} \bmod p \tag{2}$$

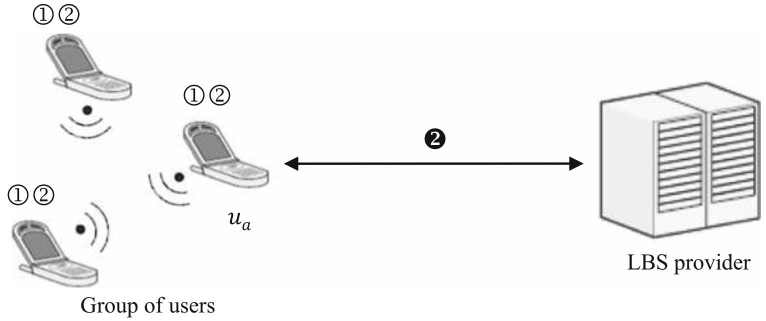


Fig. 1 System architecture, where ① denotes the first phase of the protocol, which computes the centroid cloaked region. ② denotes the second phase of the protocol, which securely computes the centroid. ❷ denotes the request-sending and result-receiving step, which can be run in parallel with phase ②

Note that k_i is the conference key constructed by u_i, is the same as other honest members' keys and is equal to Eq. (3):

$$k' = k_i = g^{e_1 e_2 + e_2 e_3 + \cdots + e_{n-1} e_n + e_n e_1} \bmod p \tag{3}$$

Considering the intractability of the Diffie–Hellman problem in G, k' (the established conference key) is only computable by group members; adversaries can find no information about it [7].

5 Cloaked-Centroid protocol

As shown in Fig. 1, the Cloaked-Centroid protocol has two major phases:

Phase 1: Location cloaking
Phase 2: Blind centroid computation.

In the first phase of the protocol (① in Fig. 1), group members jointly and securely compute a cloaked region as the group location, which includes the centroid point of their exact locations. To achieve this, each member cloaks her location based on her privacy profile and anonymously publishes her cloaked region to the public bulletin board through a pseudonym service [22].

After submitting her cloaked region, each member is able to compute the Cloaked-Centroid region by computing the average of the published cloaked regions' coordinates. The Cloaked-Centroid region contains the exact centroid point, which will be proved in the proof of correctness subsection.

Then, a representative member of the group (a randomly chosen member) u_a submits an NN query along with the Cloaked-Centroid region to the LBS, either using an onion router [52] or through a randomly selected peer [15] (❷ in Fig. 1). These techniques hide the sender's identity from the LBS provider. The LBS provider evaluates the received query and returns to u_a a set of candidate answers (A) that is guaranteed to contain the exact result (❷ in Fig. 1). We prove this fact in the next few paragraphs.

In the second phase of the protocol (② in Fig. 1), members of the group securely and collaboratively compute the centroid blindly to determine the actual answer. This phase must be conducted in a way that preserves the location privacy of all group members and protects the

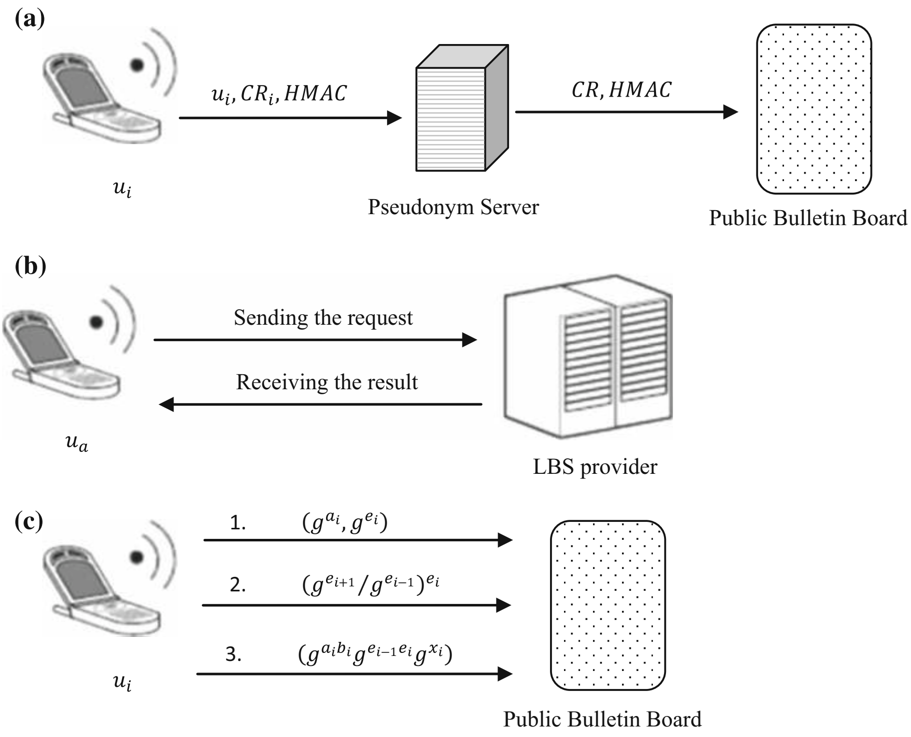


Fig. 2 Message flow of each phase of the Cloaked-Centroid protocol

location privacy of the centroid (and thus the meeting place) from possible outside attackers, including the LBS.

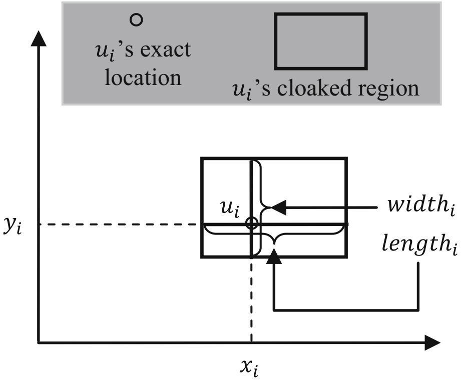
Thus, the blind centroid computation phase can be considered a special secure multiparty computation [25]; it protects users' private inputs and ensures that the computation results can only be learned by group members. Note that the computation results are the centroid coordinates, which are used to determine the exact answer. Here, we use the AV-net [29] and the Burmester–Desmedt conference key establishment protocol [7] to design a secure multiparty computation.

It is important to note that because of the parallel execution possibility, we use the same number (❷ and ②) for submitting the query and for the blind centroid computation step. We do not consider sending the query and receiving the result (step ❷) a separate phase; we consider this a subtask that can be done after step ① and in parallel with step ②. Figure 2 shows the message flow of each phase, and the following parts explain each phase in depth.

Phase 1: Location cloaking

Each user u_i determines her exact location ($l_i = [x_i, y_i]$) through a GPS-enabled device. Then, she blurs her exact location into a rectangle by generating two fixed length lines ($length_i$, parallel to the x-axis and $width_i$, parallel to the y-axis) that pass through her current location. Her cloaked region (CR_i) is then the top left and bottom right coordinates of a rectangle constructed by these two lines, as shown in Fig. 3. Note that the length of the lines is dependent on the user's policy and can change over the time and the environment, but should satisfy equation $A_{i,\min} \leq length_i * width_i$, where $A_{i,\min}$ is the minimum cloaked

Fig. 3 Location cloaking phase



area of u_i (defined in her privacy profile as her privacy requirement). Because u_i can pass the lines through her exact location at any point she wishes, this kind of cloaking ensures that all points in the cloaked region are equally likely to be the exact location of u_i.

Then, u_i anonymously publishes her cloaked region (CR_i) to the public bulletin board through a pseudonym service [22], which removes user identity such as an IP address to ensure the anonymity of the cloaked region, as shown in Fig. 2a. To prove the authenticity of the anonymous message, each member attaches an HMAC checksum to her message, which is a keyed hash of the message with a group membership key. Verification of the HMAC checksum is done by group members for each message through separately computing the HMAC checksum and comparing it with the received one. Including an HMAC checksum with the anonymous message prevents an attacker from sending fake messages because the checksum requires the attacker to know the group membership key.

When anonymity of a cloaked region is not necessary or the possibility of an attacker with background knowledge[1] is low, group members can publish their messages to the bulletin board without using a pseudonym server. In such cases, group members reveal their identities along with their blurred locations (cloaked regions). Because they do not reveal their exact locations, their location privacy is not violated; the LBS or possible outsider attackers only infer users' cloaked regions, not their exact locations. We will discuss attackers with background knowledge in Sect. 7.2.

Upon finishing this round, members compute the Cloaked-Centroid region (CR_c), which includes the exact centroid point. The coordinates of this region are computed by calculating the centroid points of the top left and bottom right coordinates of all cloaked regions, i.e., the top left coordinate of CR_i $\left(\left[x_{c,t}, y_{c,t}\right]\right)$ is computed by $\left[(1/n)\sum_{i=1}^{n} x_{i,t}, (1/n)\sum_{i=1}^{n} y_{i,t}\right]$; the same is true for the bottom right coordinate $\left(\left[x_{c,b}, y_{c,b}\right]\right)$.

Afterward, u_a (a representative member randomly chosen to communicate with the LBS) sends the NN query along with the Cloaked-Centroid region to the LBS (shown in Fig. 2b), either using onion routing [52] or through a randomly selected peer [14]. These techniques provide the anonymous usage of the LBS by concealing the sender's identity.

Phase 2: Blind centroid computation

Blind centroid computation computes the centroid of members' locations without endangering their location privacy or the centroid point privacy. We call this phase "blind" because it uses a blinding factor to hide the centroid from anyone outside the group. In this phase,

[1] An attacker with a prior knowledge about a user approximate location.

which begins in parallel with the submission of the query, group members start a special secure multiparty computation to compute the centroid point, such that users' private inputs (location coordinates) and the results of the computation (the centroid coordinates) are kept secret. To design this special secure computation, we apply and adapt the AV-net protocol [29] along with the Burmester–Desmedt conference key establishment protocol [7]. We apply a broadcast version of the Burmester–Desmedt conference key establishment protocol, which is adequately integrated with the AV-net rounds and set up during the blind centroid computation phase as follows:

As shown in Fig. 2c, each member u_i selects two random secret values $a_i, e_i \in_R Z_q$ and broadcasts (g^{a_i}, g^{e_i}) to the bulletin board. Then, she computes and publishes $t_i = (g^{e_{i+1}}/g^{e_{i-1}})^{e_i}$ to the bulletin board, which leads to the conference key computation. After finishing this step, u_i computes g^{b_i} (the AV-net value) and k' (the conference key) according to Eqs. (1) and (2), respectively.

In the third step, u_i publishes $w_i = g^{a_i b_i} g^{e_{i-1} e_i} g^{x_i}$ to the bulletin board. The structure of w_i contains $g^{a_i b_i}$ (u_i's AV-net mask) to ensure u_i's location privacy; $g^{e_{i-1} e_i}$ (u_i's portion of the conference key) to hide the result of the computation (the centroid); and x_i, which is the x-coordinate of u_i.

Multiplying all w_is results in canceling the AV-net masks and computing the conference key times the summation of x coordinates of all members, which is a discrete logarithm to the base g, $\left(k' g^{\sum_i x_i}\right)$. In particular, since a_i and b_i are AV-net values, we have $\sum_i a_i b_i = 0$ [28]; thus, we also have $\prod_i g^{a_i b_i} = g^{\sum_i a_i b_i} = 1$.

Moreover, aggregating the conference key part of all w_is results in computing the k' as follows:

$$\prod_i g^{e_{i-1} e_i} = g^{e_n e_1 + e_1 e_2 + e_2 e_3 + \cdots + e_{n-2} e_{n-1} + e_{n-1} e_n} = k'$$

Therefore, aggregating all w_is results in computing $k' g^{\sum_i x_i}$ as follows:

$$\begin{aligned}\prod_i w_i &= \prod_i g^{a_i b_i} g^{e_{i-1} e_i} g^{x_i} = \prod_i g^{a_i b_i} \prod_i g^{e_{i-1} e_i} \prod_i g^{x_i} \\ &= g^{\sum_i a_i b_i} k' g^{\sum_i x_i} = k' g^{\sum_i x_i}\end{aligned}$$

As mentioned previously, under the difficulty of Diffie–Hellman problem, k' is only computable by group members [7] and serves as a blinding factor to hide the centroid from anyone outside the group; thus, only participating users can divide the result by k' to get $g^{\sum_i^x}$.

Because $\sum x_i$ is normally a small number, group members can compute the discrete logarithm of $g^{\sum x_i}$ by applying an exhaustive search or the Pohlig–Hellman algorithm [50]. It is worth mentioning that the coordinate data are usually an integer of six- or seven-decimal digits that requires about 32 bits. Thus, $\sum x_i$ will be a small number and determining $\sum x_i$ from $g^{\sum x_i}$ will be done efficiently. Dividing the summation by n, results in computing the x coordinate of the centroid. The same is done to obtain the y coordinate of the centroid.

By receiving the candidate answer set A, each member can determine the exact result by finding the point $p \in A$ with the minimum distance to the centroid point; then, the protocol terminates. Figure 4 presents the summary of the proposed protocol.

It is worth mentioning that although applying an exhaustive search technique makes it possible to retrieve $\sum x_i$ from $g^{\sum_i^x}$, adversaries cannot benefit from this because the final result is kept hidden by the established blinding factor k', which is only known to the group members.

Cloaked-Centroid Protocol

Phase 1 (*Location Cloaking*):

i. u_i determines her exact location $\left(l_i = \left[x_i, y_i\right]\right)$ based on her GPS device.

ii. u_i cloaks her location into a rectangle by generating two fixed length line segments ($length_i$ parallel to x-axis and $width_i$ parallel to y-axis) that pass through her current location and satisfy $A_{i,min} \leq length_i * width_i$. Then her cloaked region (CR_i) is the top left and bottom right coordinates of a rectangle constructed by these two line segments.

iii. $u_i \rightarrow *\colon\ (CR_i), u_i$ anonymously broadcasts her cloaked region to the authenticated public channel.

Upon finishing Phase 1, the cloaked centroid region (CR_c) is computed by each group member as follows:

$$CR_c = \left(\left[x_{c,t}, y_{c,t}\right], \left[x_{c,b}, y_{c,b}\right]\right) =$$

$$\left(\left[(1/n)\sum_{i=1}^{n} x_{i,t}, (1/n)\sum_{i=1}^{n} y_{i,t}\right], \left[(1/n)\sum_{i=1}^{n} x_{i,b}, (1/n)\sum_{i=1}^{n} y_{i,b}\right]\right)$$

Then, u_a (a member randomly chosen to communicate with the LBS) sends the NN query along with the cloaked centroid region to the LBS, either by using an onion router [50] or through a randomly chosen peer [13].

Phase 2 (*Blind Centroid Computation*):

i. $u_i \rightarrow *\colon\ (g^{a_i}, g^{e_i})$, where $a_i, e_i \in_R Z_q$

ii. $u_i \rightarrow *\colon\ (t_i)$, where $t_i = (g^{e_{i+1}} / g^{e_{i-1}})^{e_i}$

Now u_i computes $g^{b_i} = \prod_{j=1}^{i-1} g^{a_j} / \prod_{j=i+1}^{n} g^{a_j}$ and $k' = \left(g^{e_{i-1}}\right)^{n.e_i}. t_i^{\,n-1}. t_{i+1}^{n-2} \ldots t_{i-2} \; mod\, p$.

iii. $u_i \rightarrow *\colon\ (w_i)$, where $w_i = g^{a_i b_i} g^{e_{i-1} e_i} g^{x_i}$

Upon finishing Phase 2, each u_i multiplies all the received w_is and gets the centroid x-coordinates times k'. The same is done for acquiring the centroid y-coordinates. Hence upon receiving the answer set (A), each u_i can determine the meeting point by choosing the $p \in A$ with the minimum distance to the centroid. Note that, after the LBS receives and evaluates the query, it returns the set of candidate answers A to u_a, which u_a broadcasts to the group.

Fig. 4 Cloaked-Centroid protocol

For security from malicious participants and active adversaries, we apply a zero-knowledge proof [16]. Each time a user publishes a value to the bulletin board, she must provide its zero-knowledge proof. In the case of any doubt, members can verify knowledge proofs and detect the malicious member(s). For this purpose, any zero-knowledge proof system can be applied. Because of simplicity and non-interactivity properties, we use Schnorr's signature [54], as Hao does [29]. In Schnorr's signature, 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^v is equal to $g^r g^{a_i h}$.

We apply Schnorr's signature to provide a single proof for all messages of blind centroid computation phase, namely g^{a_i}, g^{e_i}, t_i and w_i. Providing this single-knowledge proof proves the knowledge of a_i and e_i and proves that t_i and w_i are well-formed messages. To provide this proof, u_i proceeds as follows:

In step 3 of Phase 2, the user u_i publishes $\left\{g^v, g^{v'}, g_i^{v'}, g_{i,1}^v g_{i,2}^{v'} g^{v''}, r = v - a_i h, r' = v' - e_i h, r'' = v'' - x_i h\right\}$, where $g_i = g^{e_{i+1}} / g^{e_{i-1}}$, $g_{i,1} = g^{b_i}$, $g_{i,2} = g^{e_{i-1}}$, $v, v', v'' \in_R Z_q$ and $h = H\left(g, g_i, g_{i,1}, g_{i,2}, g^v, g_i^{v'}, g_{i,1}^v g_{i,2}^{v'} g^{v''}, g^{a_i}, g^{e_i}, t_i, w_i, i\right)$.

This proof can be verified by the following checks:

1. $g^v \overset{?}{=} g^r g^{a_i h}$
2. $g^{v'} \overset{?}{=} g^{r'} g^{e_i h}$
3. $g_i^{v'} \overset{?}{=} g_i^{r'} t_i^h$
4. $g_{i,1}^v g_{i,2}^{v'} g^{v''} \overset{?}{=} g_{i,1}^r g_{i,2}^{r'} g^{r''} w_i^h$

The first two checks ensure that u_i knows a_i and e_i; the next two checks ensure that u_i has constructed and published a well-formed t_i and w_i.

Proof of correctness

The Cloaked-Centroid protocol aims to retrieve the nearest POI to the group centroid; thus, to prove the correctness of the Cloaked-Centroid protocol, it suffices to prove that the sent cloaked region to the LBS contains the centroid point of the group. In other words, if the sent cloaked region contains the centroid point, then because the LBS provider evaluates the nearest POI of all points in the cloaked region, it also evaluates the nearest POI to the centroid and includes that point in the answer set. That point will thus be determined as the exact result by the group members. Proof of correctness of the Cloaked-Centroid protocol follows through Lemma 1.

Lemma 1 *The sent cloaked region to the LBS contains the centroid point of the group.*

Proof As stated earlier, the centroid coordinates are computed as the average of the x coordinates and y coordinates of all members. Assume $c = (x_c, y_c)$ as the centroid point and $CR_c = \left(\left[x_{c,t}, y_{c,t}\right], \left[x_{c,b}, y_{c,b}\right]\right)$ as the sent cloaked region to the LBS. For each member u_i, the exact location coordinates are denoted by (x_i, y_i) and her cloaked region is denoted by $CR_i = \left(\left[x_{i,t}, y_{i,t}\right], \left[x_{i,b}, y_{i,b}\right]\right)$. Without loss of generality, consider just the x coordinate. It is obvious that for each member u_i, $x_{i,t} \leq x_i \leq x_{i,b}$, so this should be true for the average function of these values over all members; thus, we have $(1/n) \sum_{i=1}^n x_{i,t} \leq (1/n) \sum_{i=1}^n x_i \leq (1/n) \sum_{i=1}^n x_{i,b}$, which means that the x coordinate of the centroid is between the lower and upper bounds of the sent cloaked region $\left(x_{c,t} \leq x_c \leq x_{c,b}\right)$. The y coordinate can be derived in the same way, and we have that $(1/n) \sum_{i=1}^n y_{i,t} \leq (1/n) \sum_{i=1}^n y_i \leq (1/n) \sum_{i=1}^n y_{i,b}$. Based on these two inequalities for x_c and y_c, it is obvious that the centroid is somewhere inside the sent cloaked region, and the proof is complete. □

6 Privacy analysis

As mentioned before, the Cloaked-Centroid protocol should satisfy the following privacy requirements:

(i) Preserving the location privacy of all group members and
(ii) Preserving the location privacy of the meeting place.

To analyze these two requirements, we investigate each phase of the protocol separately and discuss privacy requirements.

6.1 General requirements of location cloaking phase

As stated in [42,44], a location anonymization process should satisfy four general requirements: accuracy, privacy, efficiency and flexibility which are discussed in the following:

Accuracy With respect to accuracy, the anonymization process should satisfy user privacy requirements, i.e., the resulting cloaked region should be as close as possible to the user privacy requirements (defined in her privacy profile). Location cloaking in the Cloaked-Centroid protocol is done by the users themselves. Each user cloaks her location based on her privacy profile by computing a cloaked region with an area size of at least $A_{i,\min}$. Thus, the accuracy property is achieved in the Cloaked-Centroid protocol.

Privacy Regarding privacy, an adversary should not be able to infer any information about the user's exact location from the published cloaked region. Because the reported cloaked area in Cloaked-Centroid is formed by passing two fixed length lines from a user's exact location, all points in the line and consequently in the cloaked region are equally likely to be the user's exact location, so an adversary cannot infer a user's actual location. In addition, using a pseudonym server causes background knowledge attacks to fail. We will explain background knowledge attack in more detail in the next few paragraphs (Sect. 7.2).

Efficiency This property means that the cloaked area must be computed in an efficient and scalable manner. Calculating the cloaked region in the Cloaked-Centroid protocol requires only a few simple mathematical operation; therefore, it is an efficient process. The cloaking process needs no cooperation from the user's peers; hence, it is scalable and can be applied to large groups.

Flexibility Finally, in terms of flexibility, each user should be able to change her privacy profile at any time. In the Cloaked-Centroid protocol, a user can change her privacy profile (specifically $A_{i,\min}$) whenever she wishes. The proposed protocol is also flexible in that it guarantees that the user will achieve her desired privacy level.

6.2 General requirements of the blind centroid computation phase

The blind centroid computation phase determines the centroid point by running an SMC protocol. Therefore, Phase 2 should satisfy the central requirements of a general SMC protocol, which are privacy and correctness [4,39].

Regarding privacy, no information except what can be inferred from the output should be learned. More exactly, a user's private inputs must be kept hidden from other users.

Regarding correctness, each party should receive the correct output and an adversary should not be able to cause the result of the computation to deviate from its desired function [39].

In addition to these two properties, the blind centroid computation phase must satisfy an additional property known as centroid privacy: It must keep the result (the centroid) hidden from all except group members. The following paragraphs state these three properties.

Property 1 *The blind centroid computation phase preserves the location privacy of individual users.*

The blind centroid computation phase is composed of two well-known building blocks (the AV-net and Burmester–Desmedt protocols); thus, its privacy property relies on the security of these two schemes. Learning the location of a particular user (u_i) requires an attacker to learn u_i's AV-net mask and u_i's portion of the conference key.

In the case of no collusion, an attacker fails to learn the required knowledge, because doing so requires her to solve an instance of the Decisional Diffie–Hellman (DDH) problem [29], which she cannot. Specifically, finding the AV-net mask and the conference key portion requires the attacker to compute $g^{a_i b_i}$ from g^{a_i} and g^{b_i}, and compute $g^{e_{i-1} e_i}$ from $g^{e_{i-1}}$ and g^{e_i}, respectively (notice that a_is, b_is and e_is are unknown to the attacker [29]). Under the difficulty of the DDH problem [29], the attacker cannot do this and consequently fails to learn the user's location.

In the case of partial collusion against u_i, if u_{i-1} participates in the attack, then computing the conference key portion ($g^{e_{i-1} e_i}$) is straightforward because u_{i-1} knows e_{i-1}. To find the location of u_i, attackers must learn the AV-net mask, but this is not possible in a partial collusion attack. Specifically, based on the security of the AV-net scheme [29], b_i is a secret random value to colluding members in a partial collusion attack; thus, colluding members cannot cancel the mask and no useful information can be learned. Moreover, the only information that can be obtained from the zero-knowledge proofs is that the sender knows the discrete logarithms [29] and that the sender publishes the well-formed messages.

Because of the above factors, the parties' published ciphertexts do not leak any useful information and the location privacy of individual users is guaranteed; no members learn other users' locations.

Property 2 *The blind centroid computation phase of the Cloaked-Centroid protocol preserves correctness in a malicious model.*

To distort the result (centroid), malicious member may attempt to send fake values or change the sent messages of honest members; however, they will not be able to do this because of the zero-knowledge proof. Including the knowledge proof in the protocol design requires the attackers to publish a consistent zero-knowledge proof for the fake value. To rectify the attack, the honest parties exclude the malicious ones and restart the blind centroid computation phase for obtaining the correct output and their privacy remains intact. It is worth mentioning that fake values of outside attackers cannot be published to the bulletin board, because the bulletin board is an authenticated channel that only publishes authenticated messages (messages belong to the group members) and discards others.

The zero-knowledge proof is essential in the design of blind centroid computation phase. Without it, several misbehaviors resulting in outcome incorrectness would be possible. For example, if there were no knowledge proof, a participant u_i could cause the protocol outcome to be incorrect by publishing $w_i' = g^{c_i b_i} g^{e_{i-1} e_i} g^{x_i}$ or $w_i' = g^{c_i b_i} g^{e_{i-1} c_i'} g^{x_i}$, where c_i and c_i' are random values chosen by u_i. Hence, the zero-knowledge proof ensures that the protocol is self-enforcing and correct.

Property 3 *The blind centroid computation phase preserves centroid privacy against possible outside attackers, including the LBS.*

As discussed in Property 1, the blind centroid computation phase preserves user location privacy even if partial collusion occurs. Here, we explain that this phase preserves centroid privacy as well. In the last round of Phase 2, when members' broadcast values are multiplied, the result obtained is the conference key multiplied by the summation of the x coordinates (or the y coordinates). Learning the centroid requires an outside attacker to learn the conference key.

An outside attacker cannot learn the conference key, because it requires her to solve an instance of Diffie–Hellman problem according to Theorem 1 of [7]; therefore, the centroid privacy is preserved. Moreover, an attacker fails to learn useful information from zero-knowledge proofs [29]; thus, she cannot learn the centroid.

Since knowing the centroid is enough to find the meeting point, preserving the location privacy of the meeting place implies that nobody except the group members learns the centroid. As explained in Property 3, applying the conference key protocol makes this phase secure; hence, the Cloaked-Centroid protocol preserves the meeting point location privacy.

Furthermore, because the result of the LBS is a set of candidate POIs, A, with cardinality k (assuming k as the cardinality of A), the result-set anonymity property is provided with the degree k. More exactly, neither the LBS nor an attacker could deduce the location of the meeting place with a probability larger than $1/k$.

7 Security analysis

In this section, through informal analysis (such as [23,45,56,63]), we investigate the Cloaked-Centroid behavior in the case of malicious members (known as insider attackers) with background knowledge attack.

7.1 Insider attacks

Two main attacks caused by an insider are collusion attacks and disruption attacks. A malicious member may collude with other malicious parties to disclose honest members' locations. She may send fake values to prevent the protocol from achieving its goal and to cause a disruption attack, i.e., she may broadcast incorrect values for her AV-net mask or she may publish an incorrect value for t_i or w_i, or in the worst case, she may alter her location coordinates. Also, a malicious member may abort the protocol execution at any time, i.e., she may refuse to send data. Here, we study these misbehaviors and analyze how the protocol can overcome them.

7.1.1 Collusion attacks

In a collusion attack, active attackers may collude to discover the location(s) of some honest member(s) of the group. There are two types of collusion attacks: (i) full collusion and (ii) partial collusion. In a full collusion attack, all participants collude against one user in the network. The Cloaked-Centroid protocol does not preserve user location privacy in the case of a full collusion because the AV-net mask would be canceled [28]. However, it is unlikely that all participants would collude against just one [9]; thus, we consider only *partial collusion*, which involves some participants, but not all.

In the worst case, only participant u_k does not participate in a partial collusion against participant u_i. In the location cloaking phase, this partial collusion may reveal the cloaked region of u_i with probability $1/2$, since the cloaked regions of only two participants would remain anonymous. Although revealing the identified cloaked region would not be considered a threat in itself, it is a limitation of the Cloaked-Centroid protocol.

Partial collusion in the blind centroid computation phase would not reveal any useful information. Assume all group members except u_k collude against u_i to discover u_i's location. The colluding members ($n-2$ members) aim to compute x_i from $g^{a_i b_i} g^{e_{i-1} e_i} g^{x_i}$. Computing x_i requires the colluders to find $g^{e_{i-1} e_i}$ (u_i's portion of the conference key) and $g^{a_i b_i}$ (u_i's AV-net mask). Finding the value of $g^{e_{i-1} e_i}$ requires u_{i-1} to participate in the collusion; otherwise, it will fail. Assuming this participation, the colluders must find u_i's AV-net mask to disclose her coordinates. To reveal the mask, it is enough for the attackers to find b_i, but the AV-net structure (Lemma 2 of [29]) guarantees that "b_i is a secret random value to attackers in partial collusion against participant u_i" [29]. Therefore, colluding parties fail to learn b_i, and consequently, fail to discover u_i's location coordinates.

According to Yang et al. [60], a protocol is called t-private "if no collusion containing at most t parties can get any additional information from its execution". Based on the above discussion, Cloaked-Centroid protocol will be an $(n - 2)$-private protocol.

7.1.2 Disruption attacks

Broadcasting fake values for the AV-net mask can prevent a protocol from fulfilling its task; hence, it is considered a disruption attack. In this attack, a malicious party must use a fake b_i value. Due to the zero-knowledge proof, however, the malicious member would fail in her attack [29] because she would not be able to demonstrate a consistent knowledge proof for the fake value. Upon attempting to verify the zero-knowledge proof, honest parties would realize an attack had occurred because the verification would fail. They could then expel the attacker and restart the protocol without violating their location privacy.

Publishing an incorrect value for t_i may cause honest parties to come up with an incorrect k' (except the party who is immediately next to the malicious member because she constructs her key without considering the t_i of the malicious member). However, due to the zero-knowledge proof, the malicious member would fail at her attack because she would not be able to provide a consistent knowledge proof for the fake t_i. Specifically, providing any knowledge proof other than the correct one would lead to the failure of knowledge proof verification similar to the AV-net [28]; thus, the honest parties would realize the attack and then exclude the malicious member and restart the step without endangering their location privacy.

The situation is the same for a malicious member who publishes an incorrect value for w_i. Generally, the zero-knowledge proof ensures that participants follow the protocol faithfully; thus, the protocol achieves its goal.

In all multiparty computation protocol, a malicious member can always alter its input [39]. Although altering the input by a malicious member in the Cloaked-Centroid brings no benefit to the attacker, it may cause a disruption attack if the attacker sends a meaningless value for her coordinates, i.e., a large value out of the range of the location coordinates. Preventing this attack is hard, but there is a technique that ensures members use meaningful values for their coordinates.

As mentioned earlier, location coordinates are small numbers that are at most 32 bits long; to cause a disruption attack, a malicious member alters her x coordinate to a value larger than 2^{32}. To overcome this attack, although the Cloaked-Centroid protocol cannot ensure that members provide their real location data, it can ask them to prove that their inputs lie in the valid range by applying range proof protocols [5,40,48]. A range proof protocol proves that a committed secret number (the location coordinates in the case of Cloaked-Centroid) lies in a specified interval without disclosing the secret [5].

The Centroid-Cloak protocol asks members to provide a range proof for their input location coordinates when the computed coordinates for the centroid are meaningless, i.e., there is no point on the map with these coordinates. With this condition, members can start a range proof protocol to prove that their input location coordinates lie in the predefined range and also to detect the malicious member(s). Some well-known range proof protocols (that can be seamlessly integrated with the Cloaked-Centroid protocol) include the classical range proof [40] or the batch range proof [48] (see "Appendix").

Aborting the protocol execution in the first phase does not cause any harm, so other members can enter the protocol and get the desired results. A refusal to participate during Phase 2 or between the steps of Phase 2 can easily be rectified: at this point, the honest parties can identify and exclude the malicious member through the zero-knowledge verification and restart the protocol at the corresponding step.

7.2 Background knowledge attacks

In the context of location privacy, a background knowledge attack might take place when the adversary applies her prior knowledge to infer a user's identity or true location [18].

Since the blind centroid computation phase is entirely cryptographic, the adversary cannot gain any advantage from a background knowledge attack. In the location cloaking phase, group members publish their anonymous cloaked regions. Depending on the adversary's prior knowledge, one of the following situations may occur:

1. If the adversary has no background knowledge, she would learn some anonymous cloaked regions, but no knowledge about their owners. This is not a location privacy threat because the adversary would not learn the identity of group members. Hence, location privacy remains intact.
2. Assuming the adversary knows members' identity and also has some knowledge about the approximate location (AL) of a typical user u_i; by running the location cloaking phase, she may or may not obtain more accurate knowledge about u_i's location. The adversary first uses her prior knowledge $\left(AL_{u_i}\right)$ to find a correct map between u_i and u_i's anonymous published cloaked region. In finding the most probable map, the adversary has determined the cloaked location (CR_i) that most probably belongs to u_i. Assume the adversary finds the correct map, and CR_i is the actual cloaked region of u_i. If the area of CR_i is greater than that of AL_{u_i}, then the adversary gains no advantage; if the area of CR_i is smaller than that of AL_{u_i}, the adversary obtains more knowledge (CR_i) about only the approximate location of u_i. In this case, although a background knowledge attack has taken place, location privacy has not been violated because the adversary only knows the cloaked region of u_i, not her true location [18].
3. If the adversary knows the exact location of a particular user, then there is no location privacy and the location cloaking phase does not help the adversary (the adversary already knows the user's true location). This implies that no additional knowledge can be gained in the presence of this type of attacker.

8 Experiments

In this section, we evaluate the performance of Cloaked-Centroid protocol through extensive experiments. We use Sequoia[2] dataset which contains 62,556 points of interest in California and normalize it in a square of 10,000 × 10,000 units (Fig. 5). Table 1 summarizes the values used for each parameter in our experiments.

We consider the value of minimum area rectangle for each user as 0.001–0.01 % of the total space. We use group size of 16, 24, 265 and 1024. The number of required data points for NN query is set to one value in the range $\{2, 4, 8, 16, 32\}$. The size of the area that encloses the set of group users varies between 2 and 10 % of the total space. We then randomly generate 1024 location points that are uniformly distributed in the considered areas. The size of module p

[2] www.rtreeportal.com.

Fig. 5 Sequoia dataset

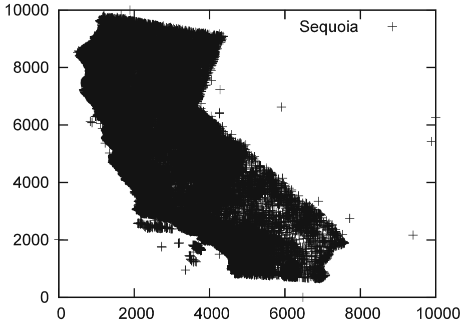


Table 1 System parameters and their values according to Hashem's work [31]

System parameter	Values	Default value
K (required data point)	2, 4, 8, 16, 32	2
Group size	16, 64, 256, 1,024	256
User query rectangle area	0.001–0.01 %	0.005
Group area size	2–10 %	2 %

for the cryptographic operation is set to 128 bits. The experiments are run on an Intel P3 2.01 GHz desktop with 1 GB of RAM.

We evaluate the performance of Cloaked-Centroid by measuring the following metrics: in terms of computation cost, we measured the CPU time and query response time; in terms of communication cost, we measure the number of returned objects by LBS (size of LBS message) and also the size of intra-group messages.

For varying group sizes, we first compare the area size of the cloaked region sent by the Cloaked-Centroid protocol versus by Hashem's method. Figure 6 shows that the area size of the Cloaked-Centroid protocol is much lower (nearly a constant value) than that of Hashem's; this is because we use the Cloaked-Centroid region as the group location rather than the MBR that encloses all user-cloaked areas, as Hashem does.

We evaluate the query response time (the time taken by each phase plus the LBS evaluation time) required by Cloaked-Centroid compared to Hashem's protocol for different group sizes and show the result in Fig. 7a. As shown in the figure, Hashem's method provides a higher query response time than Cloaked-Centroid, especially as the group size grows larger. The Cloaked-Centroid protocol is more efficient due to a lower LBS overhead and the parallel nature of Phase 2. In particular, the LBS overhead to retrieve the nearest POI for the large cloaked area is higher than that of a small one, and as we observe in Fig. 6, the area size of the sent cloaked region in Cloaked-Centroid protocol is, on average, 1,000 orders of magnitude smaller than that of Hashem's. Figure 7b shows the time required for the LBS to evaluate a query and retrieve the candidate POIs.

Further, in Phase 2 of Cloaked-Centroid, users can do their work in parallel; in Hashem's method, they must do it sequentially. In other words, the blind centroid computation phase of Cloaked-Centroid requires only three sequential steps versus n sequential steps in Hashem's method. Hence, although each user in Phase 2 of Cloaked-Centroid must perform a time-consuming task (cryptographic operations), the overall required time to complete the phase is lower than that of Hashem's. Figure 7a presents the query response time without considering

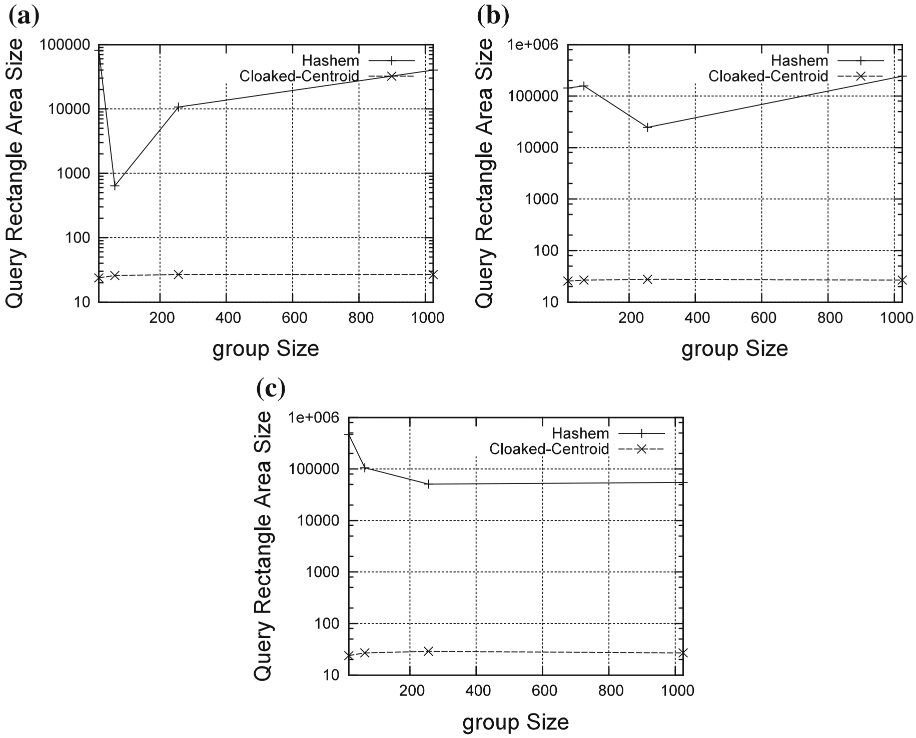


Fig. 6 Area size percentage of the sent cloaked region to the total area in logarithmic scale

the zero-knowledge operations. Figure 7c shows the execution times of Cloaked-Centroid, including the required time for generating and verifying zero-knowledge proofs. Although securing the protocol against malicious adversaries requires more computations, parallelizing the operation of Phase 2 with the LBS operations reduces the total execution time.

In Fig. 8, the result of the communication cost is presented. Since the area size of the sent cloaked region in the Cloaked-Centroid protocol is smaller than the MBR that encloses all user-cloaked regions in Hashem's method, the Cloaked-Centroid protocol has a smaller answer set (about 0.014 orders of magnitude). Therefore, the proposed protocol not only decreases bandwidth consumption, and it prevents the LBS from excessive disclosure. It is worth mentioning that the LBS message in Hashem's method consists of the candidate POIs along with the maximum and minimum total distances values for each POI, so the size of the LBS message is larger than that of Cloaked-Centroid's.

As mentioned before, the Cloaked-Centroid protocol is a resource-aware method. This property is verified by the experimental evaluation, since it saves the bandwidth by sending only one request and by receiving the smaller answer set size, as well.

To compare the intra-group communication cost, we consider the total communication costs of Phase 1 and Phase 2. We measure this cost by summing the size of all messages exchanged in both phases. In Phase 1 of Cloaked-Centroid, each user sends her imprecise location to the bulletin board, while in Hashem's method, each user collaborates with her neighbors to find her imprecise location. If the number of neighbors of each user is set to m, then she will receive m messages containing her neighbors' local cloaked regions. In Phase

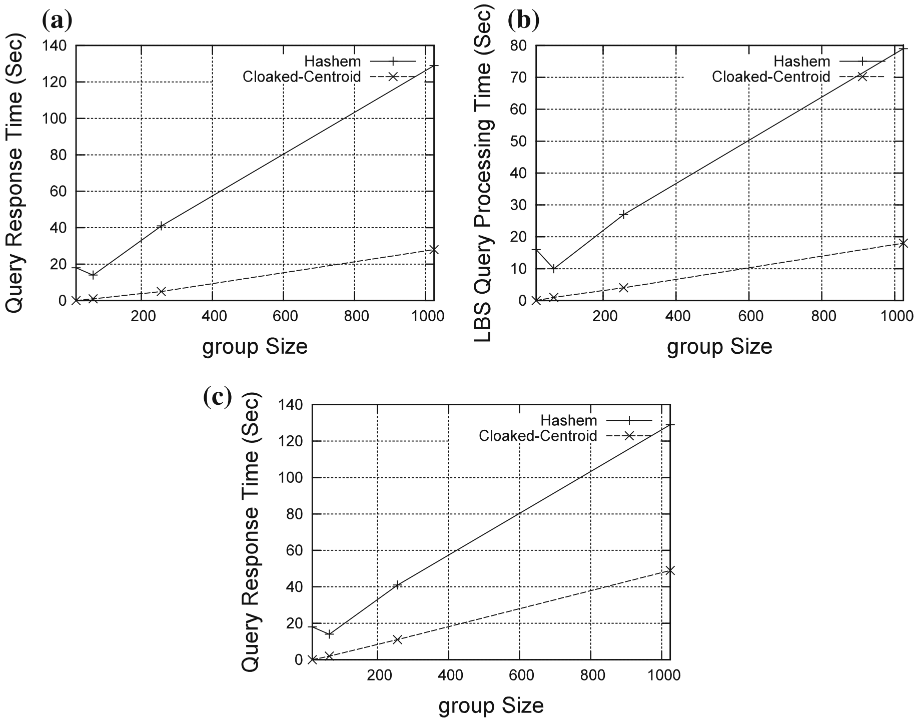


Fig. 7 Query response time and the LBS overhead for different group sizes

2, we experimentally count the number of messages and their sizes and then add the values. As a result, we conclude that the intra-group message size of Cloaked-Centroid would be more than 100 orders of magnitude smaller than that of Hashem's.

To sum up, the Cloaked-Centroid protocol preserves location privacy of group members and meeting place privacy. The proposed protocol is resistant to collusion attacks, disruption attacks and background knowledge attacks. Cloaked-Centroid is also a resource-aware protocol as it only sends one NN query to the LBS provider, which leads to a communication complexity of $O(1)$. Moreover, the communication complexity of its intra-group messages is of $O(n)$; in Hashem's protocol, it is $O(nm)$, where m is the number of response messages received by each participant from her peers [30].

It is worth mentioning that Cloaked-Centroid is independent of how LBS providers evaluate the queries; any existing privacy-preserving query-processing algorithm [11,33,43] can be used.

9 Conclusion

In this paper, we addressed the problem of supporting location privacy for a group of users while accessing location-based services. We considered a group of users that wants to benefit from an LBS and meet at a point with the smallest distance from their centroid. We identified the privacy issues of this scenario (location privacy for all group members and location privacy for the meeting place) and proposed the Cloaked-Centroid protocol to satisfy those

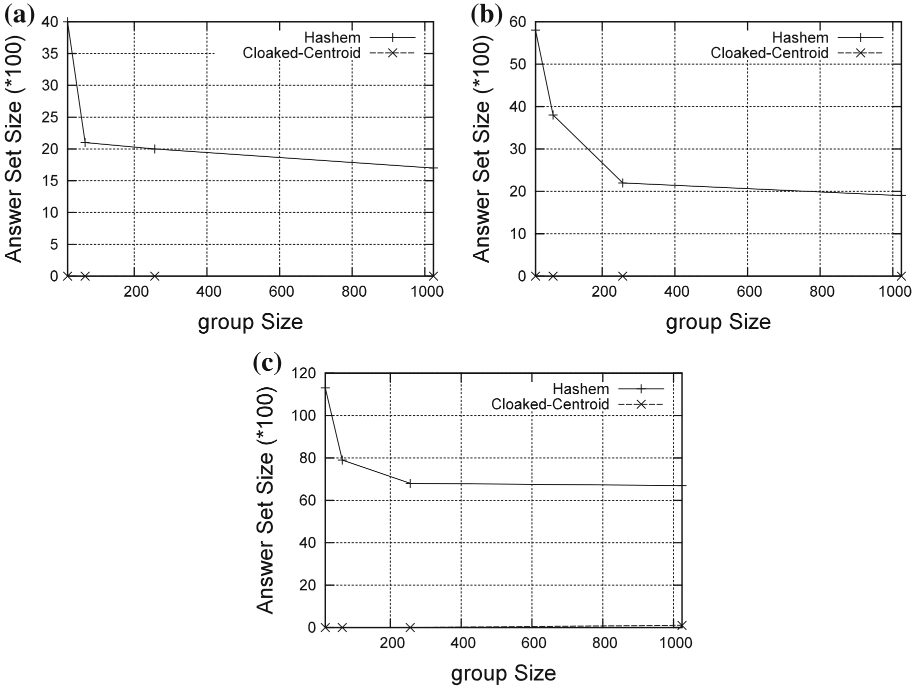


Fig. 8 Answer set size for different group sizes

issues. Our protocol provides result-set anonymity, preventing the LBS and other possible attackers from learning the location of the meeting place.

Furthermore, Cloaked-Centroid is a resource-aware solution; in sending only one query to the LBS, the overhead to evaluate the query and the size of the LBS result are significantly decreased. Moreover, as the Cloaked-Centroid protocol is independent of the query-processing algorithm of the LBS, any existing privacy-preserving query-processing algorithm can be applied.

As stated in the paper, with some caution, Cloaked-Centroid can be used for fast-moving users. We briefly discuss this option below, but leave the details for a future work. Under the fast-moving condition, users' locations change rapidly and thus also will be the meeting point. Hence, a user must consider her speed and direction in both phases of the Cloaked-Centroid protocol. In particular, the user can either blur her location with respect to her speed and direction in such a way that covers her during the protocol run while she is moving fast or she can predict her future location based on her current location, speed and direction. In the latter case, she can publish a cloaked region of her future location in the location cloaking phase and use her future location in the blind centroid computation phase. As an illustration, in Fig. 9, u_1 can use CR_1' instead of CR_1 in the location cloaking phase and $\left[x', y'\right]$ instead of $[x, y]$ in the blind centroid computation phase. The idea behind this recommendation is that the user should determine which option will better reflect her future location.

Extensive analysis shows that the Cloaked-Centroid protocol is more secure and efficient with respect to privacy preservation and bandwidth consumption than the previous technique. In addition, the proposed protocol is resistant against collusion attacks, disruption attacks and background knowledge attacks in a malicious model.

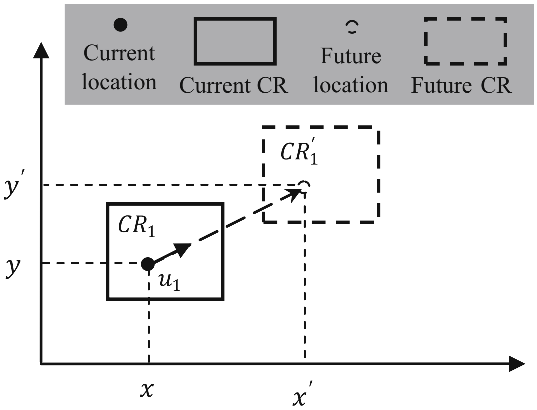


Fig. 9 Fast-moving user predicts her future location $[x', y']$ based on her current location $[x, y]$ and her speed and her direction and uses it in the blind centroid computation. She also uses her future location cloaked region CR' in the location cloaking phase

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

Appendix: Range proofs for the Cloaked-Centroid protocol

To prove $x_i, y_i \in [a, b]$ (location coordinates) in the Cloaked-Centroid protocol, the classical range proof [40] can be applied. In this proof that is based on the zero-knowledge proof of a discrete logarithm [54], the prover encodes her secret to its binary representation and then proves that each digit in this representation is either 0 or 1, using a proof of knowledge of 1 out of 2 discrete logarithms [16]. Adapting the classical range proof to the Cloaked-Centroid protocol proceeds as follows:

Assume the parameters of the range proof are the same as the Cloaked-Centroid protocol.

1. The prover generates $V = g^{x_i} h^r \bmod p$ as a commitment to x_i where h is the generator of G and r is a random integer in Z_q.
2. The prover computes $V' = V/g^a = g^{x_i - a} h^r \bmod p$; then, the proof that $x_i \in [a, b]$ is reduced to the proof that $x_i - a \in [0, b - a]$.
3. Let $x_i - a = x_0 2^0 + x_1 2^1 + \cdots + x_m 2^m$ be the binary representation of $x_i - a$, where $x_j \in \{0, 1\}$ and $j = 0, 1, \ldots, m$ where $m = 32$.
4. The prover chooses $u_0, u_1, \ldots, u_m \in_R Z_q$, and computes $u = u_0 2^0 + u_1 2^1 + \cdots + u_m 2^m \bmod q$. Then, she computes $u' = u - r$ and $E_i = E\left(x_j, u_j\right) = g^{x_j} h^{u_j} \bmod p$ for $j = 0, 1, \ldots, m$.
5. The prover sends E_j and u' to the verifier.
6. The verifier checks whether $V' h^{u'}$ is equal to $\prod_{j=0}^{m} E_j^{2^j} \bmod p$.
7. For each $E_j (j = 0, 1, \ldots, m)$, the prover and the verifier run a sub-protocol to prove that the x_j value is either 0 or 1. This can be done by applying the zero-knowledge proof of knowledge of 1 out of 2 discrete logarithms [16].

Note that before running the range proof protocol, the prover should prove that $V = g^{x_i} h^r \bmod p$ and $w_i = g^{a_i b_i} g^{a_{i-1} a_i} g^{x_i} \bmod p$ hides the same secret x_i by applying a proof of equality of two discrete logarithms [8]. Also, the verification can either be done centrally by a chosen member in the group or distributedly by all members.

The batch range proof of Peng et al. [48] is similar to the classical range proof and can also be applied. In a batch range proof, the prover represents her secret in a base-k system where k can be any integer greater than 1. Then, the prover proves $\log_k(b-a)$ instances of the proof that each digit of the base-k representation of $x_i - a$ is in Z_k. This is done using a batch proof in which the $\log_k(b-a)$ instances of proof of knowledge of 1 out of k are batched into a single proof [48]. Assuming $k = 2$, the batch proof for m instances of knowledge of 1 out of 2 discrete logarithms is as follows:

Batch proof of m instances of knowledge of 1 out of 2 discrete logarithms

Goal: the prover proves the knowledge of $b_j \in \{0,1\}$, x_{j,b_j} s.t. $y_{j,b_j} = g^{x_{j,b_j}}$, for $j = 1,2,\dots,m$.

- The prover selects $r, v, c_{j,\overline{b_j}} \in_R Z_q$ and computes

$$t_0 = g^r \prod_{\{j|b_j=1\}} {y_{j,0}}^{c_{j,0}}$$

$$t_1 = g^v \prod_{\{j|b_j=0\}} {y_{j,1}}^{c_{j,1}}$$

$$c_j = H(CI \,||\, c_{j-1} \,||c_{j-1,0})$$

$$c_{j,b_j} = c_j - c_{j,\overline{b_j}} \; mod \; q$$

$$z_0 = r - \sum_{\{j|b_j=0\}} c_{j,0} x_{j,0}$$

$$z_1 = v - \sum_{\{j|b_j=1\}} c_{j,1} x_{j,1}$$

where CI is a bit string known to the prover and the verifier, $c_0 = t_0$, $c_{0,0} = t_1$. The prover sends $(z_0, z_1, c_1, c_{1,0}, \dots, c_{m,0})$ to the verifier.

- The verifier computes:

$$c_{j,1} = c_j - c_{j,0} \;\; mod \; q$$

$$c_{j+1} = H(CI \,||\, c_j \,||c_{j,0})$$

- and verifies

$$c_1 =? \, H(CI \,||\, g^{z_0} \prod_{j=1}^{m} {y_{j,0}}^{c_{j,0}} \;||\, g^{z_1} \prod_{j=1}^{m} {y_{j,1}}^{c_{j,1}})$$

Assuming $k = 2$, adapting the batch range proof to the Cloaked-Centroid protocol proceeds as follows:

8. Steps 1 to 6 are exactly the same as for the classical range proof.
9. The prover and the verifier run a batch proof of knowledge of 1 out of 2 (or 1 out of k) discrete logarithms to prove that for each $E_j (j = 0, 1, \dots, m)$, the value of $x_j \in \{0, 1\}$ using the above batch proof.

References

1. Ardagna CA, Cremonini M, De Capitani di Vimercati S et al (2011) An obfuscation-based approach for protecting location privacy. IEEE Trans Dependable Secur Comput (TDSC) 8:13–27
2. Ashouri-Talouki M, Baraani-Dastjerdi A, Selçuk AA (2012) GLP: a cryptographic approach for group location privacy. Comput Commun 35:1527–1533
3. Bamba B, Liu L, Pesti P et al (2008) Supporting anonymous location queries in mobile environments with PrivacyGrid. In: Proceedings of world wide web conference (WWW '08), pp 237–246
4. Bickson D, Reinman T, Dolev D et al (2009) Peer-to-peer secure multi-party numerical computation facing malicious adversaries. Peer-to-Peer Netw Appl J 3:129–144
5. Boudot F (2000) Efficient proofs that a committed number lies in an interval. In: Proceedings of advances in cryptology (EUROCRYPT'00), pp 431–444
6. Boyd C, Mathuria A (2003) Protocols for authentication and key establishment. Springer, Berlin, ISBN 978-3-540-43107-7
7. Burmester M, Desmedt Y (1994) A secure and efficient conference key distribution system. In: Proceedings of advances in cryptology (EUROCRYPT'94), pp 275–286
8. Camenisch J, Michels M (1999) Proving in zero-knowledge that a number is the product of two safe primes. In: Proceedings of advances in cryptology (EUROCRYPT'99), LNCS, vol 1592, pp 106–121

9. Chaum D (1988) The dining cryptographers problem: unconditional sender and recipient untraceability. J Cryptol 1:65–67
10. Chen K, Liu L (2011) Geometric data perturbation for privacy preserving outsourced data mining. Knowl Inf Syst 29:657–695
11. Chow CY, Mokbel MF, Aref WG (2009) Casper*: query processing for location services without compromising privacy. ACM Trans Database Syst 34:1–48
12. Chow CY, Mokbel MF, Bao J et al (2011) Query-aware location anonymization for road networks. GeoInformatica 15(3):571–607
13. Chow CY, Mokbel MF (2007) Enabling private continuous queries for revealed user locations. In: Proceedings of international conference on Advances in spatial and temporal databases (SSTD'07), pp 258–273
14. Chow CY, Mokbel MF, Liu X (2006) A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In: Proceedings of the ACM symposium on advances in geographic information systems (GIS'06), pp 171–178
15. Chow CY, Mokbel MF, Liu X (2011) Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica 15:351–380
16. Cramer R, Franklin MK, Schoenmakers B et al (1996) Multi-authority secret-ballot elections with linear work. In: Proceedings of advanced in cryptology (EUROCRYPT'69), pp 72–83
17. Das K, Bhaduri K, Kargupta H (2010) A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. Knowl Inf Syst 24:341–367
18. Dewri R (2011) Location privacy and attacker knowledge: who are we fighting against? In: Proceeding of 7th international ICST conference on security and privacy in communication networks, SecureComm, London, UK
19. Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: Proceedings of international conference on pervasive computing (Pervasive'05), pp 152–170
20. Gedik B, Liu L (2008) Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans Mob Comput TMC 7:1–18
21. Ghinita G, Kalnis P, Skiadopoulos S (2007) MobiHide: a mobile peer-to-peer system for anonymous location-based queries. In: Proceedings of international symposium on advances in spatial and temporal databases (SSTD'07), pp 221–238
22. Ghinita G, Kalnis P, Skiadopoulos S (2007) PRIVÉ: anonymous location-based queries in distributed mobile systems. In: Proceedings of international conference on world wide web (WWW'07), pp 371–389
23. Ghinita G, Kalnis P, Kantarcioglu M et al (2009) A hybrid technique for private location-based queries with database protection. In: Proceedings of international symposium on advances in spatial and temporal databases (SSTD'09). LNCS, vol 5644, pp 98–116
24. Ghinita G, Kalnis P, Khoshgozaran A et al (2008) Private queries in location based services: Anonymizers are not necessary. In: Proceedings of the ACM international conference on management of data (SIGMOD'08), pp 121–132
25. Goldreich O, Micali S, Wigderson A (1987) 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 (STOC'87), pp 218–229
26. Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of MobiSys, pp 31–42
27. Gruteser M, Schelle G, Jain A et al (2003) Privacy-aware location sensor networks. In: Proceedings of USENIX workshop on hot topics in operating systems (HOTOS'03)
28. Hao F, Zielinski P (2006) A 2-round anonymous veto protocol. In: Proceedings of the 14th international workshop on security protocols, Cambridge. LNCS, vol 5087, pp 202–211
29. Hao F, Zielinski P (2009) The power of anonymous veto in public discussion. Trans Comput Sci IV 5430:41–52
30. Hashem T and Kulik L (2007) Safeguarding location privacy in wireless ad-hoc networks. In: Proceedings of international conference on ubiquitous computing (Ubicomp'07), pp 372–390
31. Hashem T, Kulik L, Zhang R (2010) Privacy preserving group nearest neighbor queries. In: Proceedings of international conference on extending database technology (EDBT'10), pp 489–500
32. Hu H, Xu J (2009) Non-exposure location anonymity. In: Proceedings of IEEE international conference on data engineering (ICDE'09), pp 1120–1131
33. Kalnis P, Ghinita G, Mouratidis K et al (2007) Preventing location-based identity inference in anonymous spatial queries. IEEE Trans Knowl Data Eng (IEEE TKDE) 19:1719–1733
34. Khoshgozaran A, Shahabi C, Shirani-Mehr H (2011) Location privacy: going beyond K-anonymity, cloaking and anonymizers. Knowl Inf Syst 26:435–465

35. Khoshgozaran A, Shahabi C (2007) Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Proceedings of international conference on advances in spatial and temporal databases (SSTD'07), pp 239–257
36. Kiayias A, Yung M (2003) Non-interactive zero-sharing with applications to private distributed decision making. In: Proceedings of financial cryptography. LNCS, vol 2742, pp 303–320
37. Langheinrich M (2002) A privacy awareness system for ubiquitous computing environments. In: Proceedings of the 4th international conference on ubiquitous computing (UbiComp'02), pp 237–245
38. Lee B, Oh J, Yu H et al. (2011) Protecting location privacy using location semantics. In: Proceedings of ACM international conference on knowledge discovery and data mining (KDD'11), pp 1289–1297
39. Lindell Y, Pinkas B (2002) Privacy preserving data mining. J Cryptol 15(3):177–206
40. Mao W (1998) Guaranteed correct sharing of integer factorization with off-line shareholders. In: Proceedings of public key cryptography (PKC'98), pp 27–42
41. Menezes AJ, Van Oorschot PC, Vanstone SA (1997) Handbook of applied cryptography. CRC Press, Boca Raton
42. Mokbel MF (2008) Privacy-preserving location services. In: Proceedings of IEEE international conference on data engineering (ICDM'08), Pisa, Italy (3-hours tutorial)
43. Mokbel MF, Chow CY, Aref WG (2006) The new casper: query processing for location services without compromising privacy. In: Proceedings of the 32nd international conference on very large data bases (VLDB'06), pp 763–774
44. Mokbel MF (2007) Privacy in location-based services: state-of-the-art and research directions. In: IEEE international conference on mobile data management, MDM 2007, Mannheim, Germany (3-hours tutorial)
45. Olumofin F, Tysowski PK, Goldberg I et al (2010) Achieving efficient query privacy for location based services. In: Proceedings of the 10th international conference on privacy enhancing technologies (PETS'10), pp 93–110
46. Paillier P, Pointcheval D (1999) Efficient public-key cryptosystems provably secure against active adversaries. In: Advances in cryptology (ASIACRYPT'99), pp 165–179
47. Papadias D, Tao Y, Mouratidis K et al (2005) Aggregate nearest neighbor queries in spatial databases. ACM Trans Database Syst (TODS) 30:529–576
48. Peng K, Bao F (2010) Batch range proof for practical small ranges. In: Proceedings of the AFRICACRYPT. LNCS, vol 6055, pp 114–130
49. Pieprzyk J, Hardjono T, Seberry J (2003) Fundamentals of computer security. Springer, Berlin, ISBN 978-3-540-43101-5
50. Pohlig S, Hellman M (1978) An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Trans Inf Theory 24:106–110
51. Ramakrishnan R, Gehrke J (2009) Database Manag Syst, 3rd edn. WCB/McGraw-Hill, New York
52. Reed MG, Syverson PF, Goldschlag DM (1998) Anonymous connections and onion routing. IEEE J Sel Areas Commun 16:482–494
53. Sakuma J, Kobayashi S (2010) Large-scale k-means clustering with user-centric privacy-preservation. Knowl Inf Syst 25:253–279
54. Schnorr CP (1991) Efficient signature generation by smart cards. J Cryptol 4:161–174
55. Solanas A, Domingo-Ferrer J, Martínez-Ballesté A (2008) Location privacy in location-based services: beyond TTP-based schemes. In: Proceeding of 1st international workshop on privacy in location-based applications (PILBA) within 13th European symposium on research in computer security (ESORICS), pp 12–23
56. Solanas A, Martínez-Ballesté A (2008) A TTP-free protocol for location privacy in location-based services. Comput Commun 31:1181–1191
57. Strassman M, Collier C (2004) Case study: the development of the find friends application. In: Schiller JH, Voisard A (eds) Location-based services. Morgan Kaufmann, Los Altos, pp 27–40
58. Tai CH, Yu PS, Yang DN et al (2011) Privacy-preserving social network publication against friendship attacks. In: Proceedings of ACM international conference on knowledge discovery and data mining (KDD'11), pp 1262–1270
59. Yakut I, Polat H (2012) Privacy-preserving hybrid collaborative filtering on cross distributed data. Knowl Inf Syst 30:405–433. doi:10.1007/s10115-011-0395-3
60. Yang B, Nakagawa B, Sato I, Sakuma J (2010) Collusion-resistant privacy-preserving data mining. In: Proceedings of the ACM international conference on knowledge discovery and data mining (KDD'10), pp 483–492
61. Yiu ML, Jensen C, Huang X et al (2008) SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: Proceedings of IEEE international conference on data engineering (ICDE'08), pp 366–375

62. Zhong G, Goldberg I, Hengartner U (2007) Louis, lester and pierre: three protocols for location privacy. In: Proceedings of privacy enhancing technologies (PET'07), pp 62–76
63. Zhong G, Hengartner U (2009) A distributed k-anonymity protocol for location privacy. In: Proceedings of IEEE international conference on pervasive computing and communications (PerCom'09), pp 253–262
64. Zhou B, Pei J (2011) The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowl Inf Syst 28:47–77. doi:10.1007/s10115-010-0311-2

Maede Ashouri-Talouki is an Assistant Professor of IT Engineering department of the University of Isfahan (Iran). She received her B.S. degree and M.S. degree in Computer Engineering from the University of Isfahan (Iran) in 2004 and 2007, respectively. In 2012, she received her Ph.D. degree at University of Isfahan in computer engineering. In 2013, she joined the University of Isfahan (Iran). Her research interests include mobile networks security, user privacy and anonymity, cryptographic protocols, distributed cryptography protocols and network security.

Ahmad Baraani-Dastjerdi is an Associate Professor of the Software Engineering department of the University of Isfahan (Iran). He received his B.S. degree from the Ferdowsi University (Iran) in Statistics and Computing in 1977 and his M.S. degree in 1979 from the George Washington University (USA) in Computer Science. In 1996, he received his Ph.D. degree at University of Wollongong (Australia) in Computer Science. In 1996, he joined the University of Isfahan (Iran). His research interests include data security, cryptography, security in computing and security in e-commerce.

Ali Aydın Selçuk is an Associate Professor of the Computer Engineering department of TOBB University of Economics and Technology (Turkey). He received his B.S. degree from Middle East Technical University (Turkey) in 1993 and M.S. from Bilkent University in 1995, in Industrial Engineering, and his Ph.D. degree in Computer Science from University of Maryland, Baltimore County, in 2001. His previous work experience includes RSA Data Security, Novell, and the Network Systems Laboratory of Purdue University. He joined TOBB University in 2013. His research interests include cryptanalysis of block and stream ciphers, secure multiparty computation, group and multicast key management.