Hindawi
Mobile Information Systems
Volume 2017, Article ID 2189646, 11 pages
https://doi.org/10.1155/2017/2189646

Research Article

A Hybrid Location Privacy Solution for Mobile LBS

Ruchika Gupta and Udai Pratap Rao

Department of Computer Engineering, National Institute of Technology, Surat, Gujarat 395007, India

Correspondence should be addressed to Ruchika Gupta; rgupt009@gmail.com

Received 9 December 2016; Revised 2 March 2017; Accepted 8 March 2017; Published 18 June 2017

Academic Editor: Jaegeol Yim

The prevalent usage of location based services, where getting any service is solely based on the user's current location, has raised an extreme concern over location privacy of the user. Generalized approaches dealing with location privacy, referred to as cloaking and obfuscation, are mainly based on a trusted third party, in which all the data remain available at a central server and thus complete knowledge of the query exists at the central node. This is the major limitation of such approaches; on the other hand, in trusted third-party-free framework clients collaborate with each other and freely communicate with the service provider without any third-party involvement. Measuring and evaluating trust among peers is a crucial aspect in trusted third-party-free framework. This paper exploits the merits and mitigating the shortcomings of both of these approaches. We propose a hybrid solution, HYB, to achieve location privacy for the mobile users who use location services frequently. The proposed HYB scheme is based on the collaborative preprocessing of location data and utilizes the benefits of homomorphic encryption technique. Location privacy is achieved at two levels, namely, at the *proximity* level and at *distant* level. The proposed HYB solution preserves the user's location privacy effectively under specific, pull-based, sporadic query scenario.

1. Introduction

The intense development of location detection empowered devices and escalated availability of wireless interconnections almost everywhere results in emerging location based applications. In Location Based Services (LBS), we incline to use positioning technology to register mobile location movement. There are quite a lot of abstract approaches and real implementations of systems to resolve the place of a cell phone. The most outstanding example of such a positioning system is the GPS [1, 2]. Although LBS offer major openings for a large variety of markets and remarkable convenience to the end user, it also presents subtle privacy attacks at the same time. Privacy of the system is threatened due to the requirement of the current location of the user in order to provide related services.

As per the connotation, LBS (i.e., services based on location) needs user's exact location coordinates to supply accurate service support to the user. Centralized architecture and decentralized architecture, also referred to as trusted third party (TTP) based and TTP-free architectures, respectively, are two basic frameworks existing to preserve location privacy of the user in LBS. An adversary with the adequate accessibility to user's data may use the location information for a particular motive and may also keep it to perform the linkages with publicly available data for detailed profiling of the user [3]. LBS may also use such data for business promotions through advertising. The series of submitted location with query from a specific place can disclose too much about a person. The scenario can become extremely unpleasant if the adversary gets access to the user's sequence of location data with attached timestamps. For example, first visit of Alice to an attorney's office speaks less about her but few days later, her subsequent visit to the court reveals altogether a different story. Location revelation by Alice to LBS provider discloses some extremely private affairs of her life through inference attacks which were not apparent otherwise [4].

The query "Find my nearest attorney's office" by Alice can directly be answered by a location server such as Google

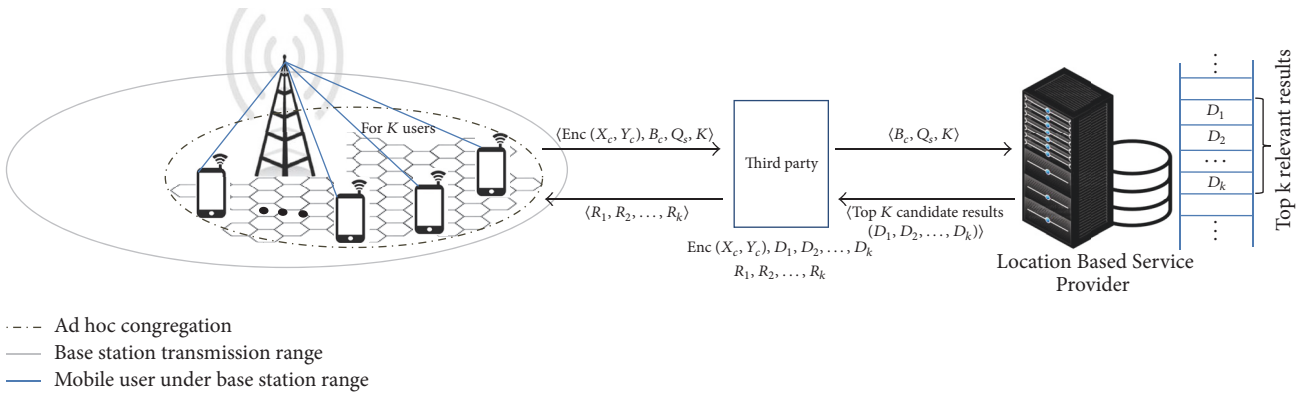

Figure 1: System model.

maps, bing maps, and map quest but the connection to these servers are not trusted. Therefore, instead in order to protect privacy, Alice sends her query via a TTP (also called anonymizer) that strips off her identification information, generates the blurred location data, and mediates the communication between her and LBS provider [5, 6]. However, the query submitted by Alice to TTP still has her actual location coordinates; hence malicious user having control over TTP can have complete information about the user. Thus it is always risky to use TTP based framework to connect to the LBS server. Trusting the third party is the prime downside of the TTP based mechanisms. If a user can trust a third party for small functionality then why not the service provider for bigger benefits, can always be argued. In distributed peer approach, mobile clients are equipped to connect with other mobile users as and when required. The development of distributed wireless communication technologies, such as WLAN IEEE 802.11, Bluetooth IEEE 802.15.1, and ZigBee (for low energy devices) IEEE 802.15.4-based specifications, combined with the propelled computing potential and memory capacity of today's mobile devices become useful to bring privacy preserving benefits to the user. This way the need to rely solely on the connection to the server is eliminated. In TTP-free architecture, all functions are supposed to be carried out at the user's handheld and thus make the communication heavier and more time consuming. Efficiency of decentralized architecture also depends upon the computing capability of used mobile device. However, peers' trust measure and evaluation is another big concern.

Figure 1 presents the proposed architecture of hybrid model. Here, it is presumed that there are a substantial number of mobile users carrying handheld devices such as cell phones, PDAs, or the like which are equipped with positioning capabilities and use location services frequently. The handhelds have computation power, processing potential, memory, and required access to the wireless network. All the users are in the transmission range of the base station (or beacon node).

In the proposed hybrid model, we suggest that the mobile user querying LBS first forms an ad hoc congregation with other users exploiting the well-established principle of $\mathcal{K}$-anonymity. Once the congregation is formed, centroid is calculated in such a way that participating users' locations are not revealed. The centroid coordinates are then secured using encryption and sent to the third party (TP). Query ($\mathcal{Q}$) includes secured location coordinates, nearest base station information, anonymity parameter $\mathcal{K}$, and the query string. TP strips off the encrypted data and without performing any changes forwards the rest of the query to the service provider. Service provider sends top $\mathcal{K}$ most relevant candidate result set (with reference to the beacon node) back to TP. TP then processes the inputs, performs homomorphic operation, and sends the result back to the congregation. The proposed HYB solution works well for specific queries in which queries are more personalized to the user specific needs.

Location queries can be categorized as generalized or specific queries. A generalized query can also be viewed as a general public query that fulfills the mass requirement, whereas specific query is the one that satisfies individual's need. "Find my nearest retail banking branch of SBI Bank" is the example of specific query, while "Find my nearest bank" is the example of generalized query. In our work it is assumed that user uses the location services to retrieve specific information. The novelty of the proposed hybrid solution is that it exploits the merits of TP and peer group formation without trusting TP as coordinates are kept private by securing them using encryption. Neither query issuer nor TP is aware about the exact locations of the members involved yet it communicates the required results. The rest of the paper is organized as follows: Section 2 highlights the related work. Sections 3 and 4 exhibit the proposed congregation model and homomorphic encryption technique, respectively. The proposed HYB solution is described in Section 5. Section 6 presents performance metrics of HYB solution. Finally, Section 7 concludes the paper.

2. Related Work

A survey of literature in the field of location privacy pertaining to LBS has brought forth several frameworks, architectures, algorithms, and techniques given by numerous researchers and practitioners. Broadly, existing defense mechanisms are based on either of the two architectures: (1)

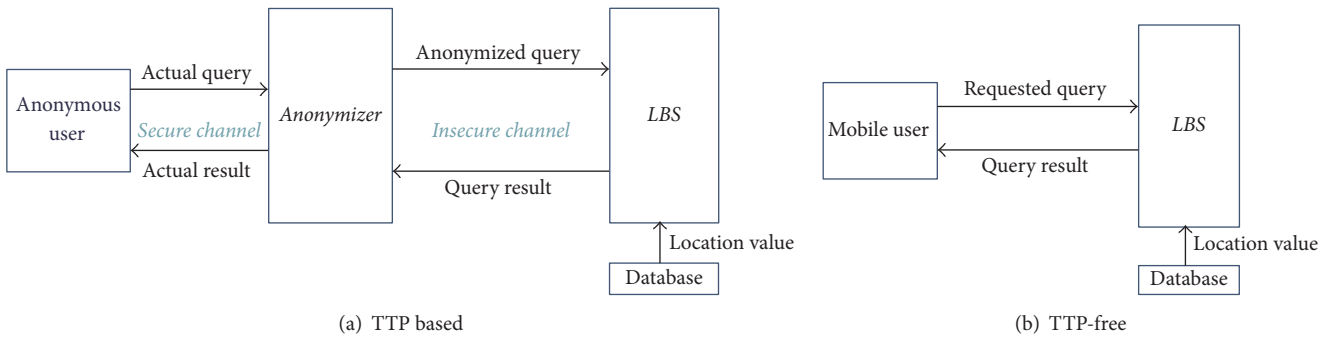

Figure 2: Existing frameworks.

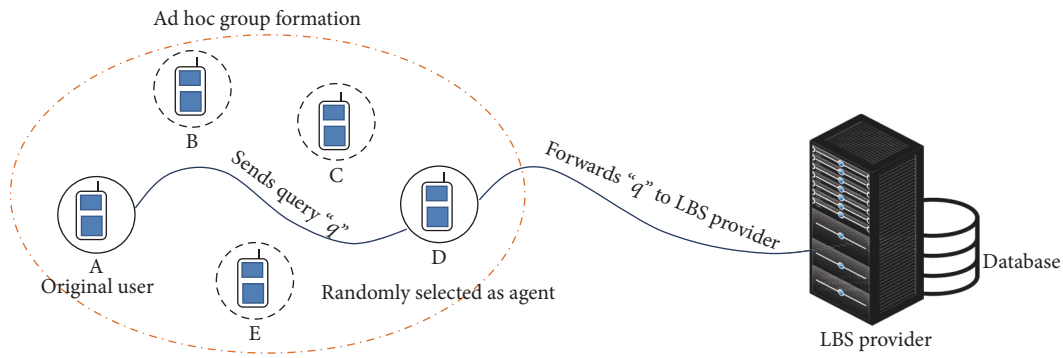

Figure 3: An instance of peer-to-peer spatial cloaking.

centralized architecture or (2) *decentralized architecture*. The setup of these architectures is shown in Figure 2.

In centralized architecture TTP acts as a proxy for service requests and responses between the user and service provider. The greater part of the previous work relies on TTP that mediates user and LBS server [6, 7]. Location anonymity is vastly discussed by [8, 9] in the TTP based architecture. The technique is based on hiding the position data before passing them to the LBS provider. $\mathcal{K}$-anonymity operates by hiding the position of the end user within a set of $\mathcal{K}$ members. Anonymizer includes additional $\mathcal{K} - 1$ users and forwards the anonymized query to LBS provider. It is now difficult for the LBS provider to distinguish the correct user from a set of $\mathcal{K}$ anonymous users. Following are few major constraints due to which TTP based methodologies are losing their ubiquity: (a) The centralized trusted third party can be the system bottleneck, (b) single point of failure is present, (c) a serious privacy threat can occur if the third party is attacked by an adversary, and (d) trusting TP is an absolute vulnerability to the user privacy. Existing cloaking mechanisms are unable to successfully ensure the user's location privacy in a continuous location query scenario (e.g., on the fly route assistance) and can deduce the real location of the client by performing trajectory attacks and dummy continual queries attack [10, 11]. Authors in [12–15] suggest diverse new ideas of using mix zones to mitigate trajectory inference and other attacks. However, it is acceptable but not sufficient to use only technical solutions.

Decentralized architectures, on the other hand, do not consider any intermediate party between users and service provider [16]. The first very basic method proposed to preserve location privacy is through the use of privacy policies [17]. Due the presence of hidden clauses and unsaid policies, this method could not serve the objective of user privacy efficiently for long and as LBS users grew drastically over the years there was a need to have a better and foolproof mechanism. Authors in [18, 19] propose the idea of distributed peer-to-peer communication among mobile users that can freely talk to each other. In this framework, dependence on the third party is eliminated and mobile users are allowed to form an ad hoc network out of which one mobile client is randomly selected as the agent to carry out the communication between querier and LBS server [16]. First, in the query issuer, let user A (refer to Figure 3) glance around and discover the rest of the collaborators to collaborate as a group. The four group members are the mobile users B, C, D, and E; out of them D is randomly chosen as an agent to mediate the communication. Trust among peers plays a profound role in such mechanisms. Evaluation and quantification of trust is another big challenge.

Another TTP-free approach given by [20] proposes a technique to preserve privacy using the concept of geoindistinguishability by adding Laplace noise to the user's Cartesian coordinates. The main objective is to protect issuer's location information while forwarding the aggregate data about the user's area. Differential privacy works on the principle that

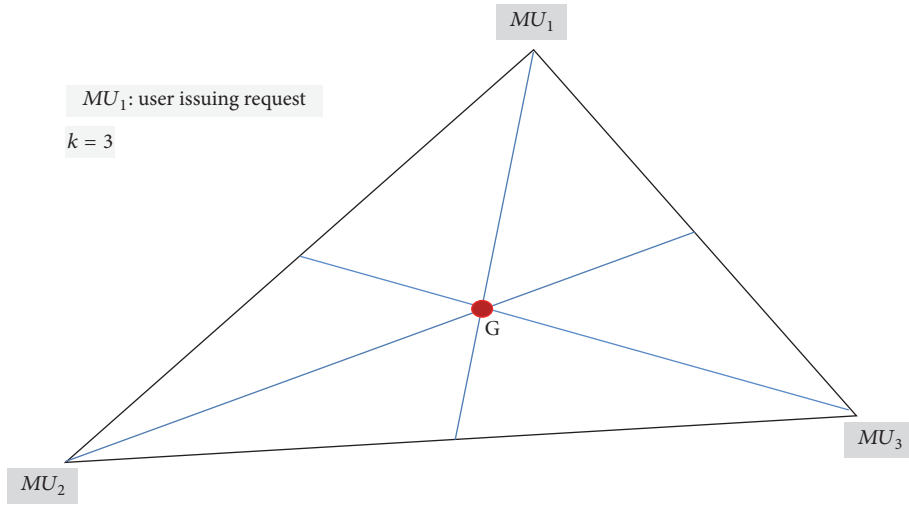

Figure 4: Microaggregation illustration.

modifying one record should have a negligible impact on the outcome of the query. The basic privacy enhancing techniques are first discussed in [21] which protects user's privacy by reducing personal identifiable information without any compromise in system's functionality. Client side obfuscation is also used in which location is repositioned by a random distance and angle of rotation at user's end [22]. The prime shortcoming with such approaches is that different users have different privacy requirement and utility thresholds. Private Information Retrieval (PIR) techniques are also proposed to safeguard the sensitive information like location of the user [23, 24]. These solutions have always been very expensive in terms of operations' computation time, communication cost, and resources needed [25]. Author in [26] first proposed the distributed concept for achieving location privacy in LBS. In this microaggregation based scheme, the major standard of the methodology is to find out the centroid of at least $\mathcal{K}$ perturbed user locations by including zero-mean Gaussian noise and send directly to the LBS database server as shown in Figure 4. The principle issue with [26] is that the centroid of locations with zero-mean Gaussian noise perturbation can be used to deduce the real location if the centroid procedure is repeated several times with the locations of static users. To prevent this problem, authors [27] use a protocol based on privacy homomorphism to ensure that centroid is computed without any knowledge of the real location of the user. Later the similar concept of public key privacy homomorphism is proposed by [28] to achieve location privacy. This is a TTP-free approach in which locations are encrypted under LBS public key and LBS later decrypts them and divides the outcome by the number of users involved to compute centroid. Location decryption by LBS makes this scheme weak and vulnerable to attacks.

The proposed HYB model is dissimilar to these approaches in a way that our solution exploits the merits of both the approaches (TTP based and TTP-free) without disclosing real location of the user anywhere throughout the communication. As of our knowledge the proposed HYB model is the first of its kind that preserves the user's location privacy at two levels, namely, at *proximity* level, while forming congregation, and at *distant* level, while sending encrypted locations to TP and TP performs computation over encrypted input values thereafter.

3. Congregation Model

The model suggests that the query issuer congregates with other $\mathcal{K} - 1$ users as a group and computes the aggregate without knowing the exact locations of the peers. The mobile user mu first broadcasts a *congregate* message to neighboring nodes and shows the intent to use location service. Upon receiving the *congregate* message, willing neighboring nodes send acknowledgment and an ad hoc congregation is formed.

Figure 5 presents the congregation model used in our system model. The mobile user $\mathcal{A}$ considers to be the query issuer node, the one who wants to use location related services. In order to keep the actual location coordinates unknown to others, locations are perturbed by adding a random split to the actual locations. Whole protocol goes as follows.

Protocol 1 (collaborative congregation).

(1) The mobile user mu (the query issuer) adds the random noise to her actual location coordinate (x, y) and generates a tweaked version of the real location, given as

$$\left(x', y'\right) = \left(x + \delta_x, y + \delta_y\right). \quad (1)$$

(2) mu broadcasts a *congregate* message to all neighboring nodes using her tweaked location coordinates to form an ad hoc congregation $\mathcal{C}$.

(3) Willing nodes acknowledge and mu selects $\mathcal{K}$ neighbors to form $\mathcal{C}$. If lesser than $\mathcal{K}$ neighbors acknowledge, step (2) is repeated until required $\mathcal{C}$ is formed

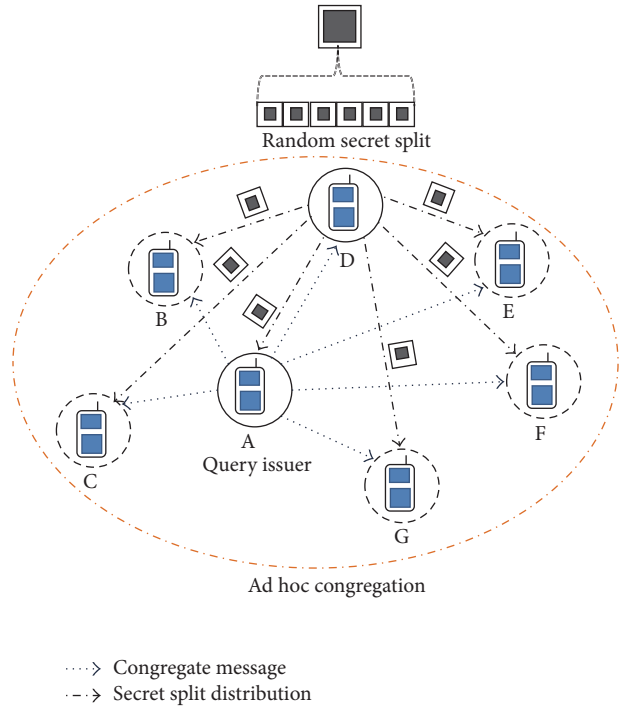

Figure 5: An instance of ad hoc congregation.

which satisfies $\mathscr{K}$. If $\mathscr{K}$ requirement is not fulfilled within a period of Δt, abort and reinitiate the process after $\mathscr{T}$ time interval.

The paucity of enough $\mathscr{K}$ users may introduce unnecessary delay in the query. Therefore, it becomes critical to choose an appropriate value of $\mathscr{K}$. For instance, why would a user feel protected for $\mathscr{K} = 10$ but not the same when $\mathscr{K} = 9$? In many cases $\mathscr{K}$ is demographic dependent, as specifying a larger $\mathscr{K}$ is acceptable for highly populated area, but choosing the same $\mathscr{K}$ value in a deserted area can cause delay in the requested service.

(4) mu randomly selects a node as congregation executor, $E_{\mathbb{C}}$. The responsibility of $E_{\mathbb{C}_i}$ is to facilitate the communication for a congregation $\mathbb{C}_i$.

(5) Now, $E_{\mathbb{C}}$ chooses and splits two sufficiently large random shares $\mathcal{S}_x$ and $\mathcal{S}_y$ such that

$$\begin{aligned} \mathcal{S}_x &= \mathcal{S}_{1,x} + \mathcal{S}_{2,x} + \cdots + \mathcal{S}_{\mathscr{K},x} \\ \mathcal{S}_y &= \mathcal{S}_{1,y} + \mathcal{S}_{2,y} + \cdots + \mathcal{S}_{\mathscr{K},y}. \end{aligned} \tag{2}$$

Splits are generated in such a way that

$$\begin{aligned} \sum_{i\in\mathbb{C}, i=1 \text{ to } \mathscr{K}} \mathcal{S}_{i,x} &= 0, \\ \sum_{i\in\mathbb{C}, i=1 \text{ to } \mathscr{K}} \mathcal{S}_{i,y} &= 0. \end{aligned} \tag{3}$$

(6) $E_{\mathbb{C}}$ sends splits to all the members of $\mathbb{C}$.

(7) Upon receiving the split, each neighbor (including mu) computes a new location (x_p, y_p) by adding the received split value to their actual location coordinates and send them back to $E_{\mathbb{C}}$.

$$\left(x_p, y_p\right) = \left(x_i + \mathcal{S}_{i,x}, y_i + \mathcal{S}_{i,y}\right). \tag{4}$$

(8) $E_{\mathbb{C}}$ computes the centroid of $\mathbb{C}$ defined as

$$\begin{aligned} X_{\mathbb{C}} &= \sum_{i=1 \text{ to } \mathscr{K}} \frac{x_{p,i}}{\mathscr{K}}, \\ Y_{\mathbb{C}} &= \sum_{i=1 \text{ to } \mathscr{K}} \frac{y_{p,i}}{\mathscr{K}}. \end{aligned} \tag{5}$$

(9) $E_{\mathbb{C}}$ passes the centroid $(X_{\mathbb{C}}, Y_{\mathbb{C}})$ to mu and leaves $\mathbb{C}$.

In Figure 5 node $\mathscr{A}$ is the query issuer, while nodes $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$, $\mathscr{E}$, $\mathscr{F}$, and $\mathscr{G}$ are the peer members of $\mathbb{C}$. Node $\mathscr{D}$ is randomly selected as $E_{\mathbb{C}}$ and $\mathscr{K} = 6$ is assumed.

Protocol 2 ($\mathbb{C}$ to TP communication).

(a) mu encrypts $(X_{\mathbb{C}}, Y_{\mathbb{C}})$ by her own public key (pk) and gets the encrypted value $\mathscr{E}(X_{\mathbb{C}}, Y_{\mathbb{C}})$.

(b) mu generates the query describes as

$$\mathcal{Q}\text{: } \left\langle \mathscr{E}\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right), \mathscr{BS}_{\mathbb{C}}, \mathscr{K}, \text{"specific search string"} \right\rangle, \tag{6}$$

where $\mathscr{BS}_{\mathbb{C}}$ is the identifier of the base station under which umbrella $\mathbb{C}$ is formed and $\mathscr{K}$ is the anonymity parameter specified by mu.

4. Homomorphic Encryption

An efficient and straightforward remedy to preserve user privacy in location (or any cloud based) services is to encrypt the information before sending to the service provider. Nonetheless, this straightforward arrangement has a critical downside in that if the information is scrambled utilizing a routine encryption method, the service provider (or cloud) can not process the information without decrypting it first. Obviously, sharing the secret decryption key with service provider again puts the same problem of privacy at stake.

In order to eliminate the mentioned problem of user privacy, a homomorphic encryption technique is used that permits some calculation to be performed specifically on encrypted information without any decryption [29].

Broadly, homomorphic encryption can be defined as follows: Suppose $\mathscr{P}$ represents the plain texts set, $\mathscr{C}$ represents corresponding set of cipher texts, and $\mathscr{ENC}$ denotes given encryption function; the cryptosystem is said to be *homomorphic* if it satisfies

$$\mathscr{ENC}\left(p_1 \odot_{\mathscr{P}} p_2\right) \longleftarrow \mathscr{ENC}\left(p_1\right) \odot_{\mathscr{C}} \mathscr{ENC}\left(p_2\right), \quad \forall p_1, p_2 \in \mathscr{P}, \tag{7}$$

where $\odot_{\mathscr{P}}$ in $\mathscr{P}$ and $\odot_{\mathscr{C}}$ in $\mathscr{C}$ are some operators. We call such disposition an *additive homomorphism* if we use addition operators and a *multiplicative homomorphism* if we use multiplication operators.

Homomorphism supports both types of encryption scheme: a symmetric key encryption and an asymmetric key encryption. There are three key elements required to specify a public key (or asymmetric) cryptosystem: an encryption algorithm $\mathscr{ENC}_{\mathrm{pk}}$, a decryption algorithm $\mathscr{DEC}_{\mathrm{sk}}$, and a key-pair generator algorithm that produces the public key and secret key (or private key) pair. The $\mathscr{ENC}_{\mathrm{pk}}$ algorithm takes the plain text and produces the encrypted text using public key pk. The output of $\mathscr{ENC}_{\mathrm{pk}}$ becomes input for $\mathscr{DEC}_{\mathrm{sk}}$ algorithm and encrypted text decrypts using the secret key sk. Homomorphic encryption permits calculations to be done on encrypted data (or cipher text). The computations are done in such a way that result when decrypted (using sk) matches the results of operations performed on the plain text.

Our proposed hybrid model takes the advantage of the homomorphic encryption property which allows the operations to be performed over encrypted data without decrypting it. Unlike existing addition and multiplication operations over encrypted data, we suggest difference (or subtraction) operation over encrypted data. However, existing cryptosystem that supports additive homomorphism [30, 31] is used to perform the proposed operation.

5. Proposed Hybrid Model

Hybrid model is built upon the concept of collaborative congregation and use of third party to mediate the results in a more effective way. The hybrid scheme appears to be centralized (due to TP) yet decentralized as no user locations are disclosed even to TP during entire communication. TP is used to provide computational support that makes the overall communication faster and efficient.

Following are the phases of our proposed scheme.

Phase 1 (ad hoc congregation $\mathbb{C}$). Mobile user mu, who wants to avail the location service, first broadcasts a *congregate* message to neighbors until required $\mathscr{K}$ users respond. This phase ends with a formation of $\mathbb{C}$ and a computed pair of $(X_{\mathbb{C}}, Y_{\mathbb{C}})$ at mu as per Protocol 1 of Section 3. mu encrypts the centroid coordinates $(X_{\mathbb{C}}, Y_{\mathbb{C}})$ with her own public key (pk) and forwards the query $\mathscr{Q}$ to TP as per Protocol 2 of Section 3.

Phase 2 (communication from TP to LBS and back). Once TP receives $\mathscr{Q}$, it strips off $\mathscr{E}(X_{\mathbb{C}}, Y_{\mathbb{C}})$ and forwards remaining $\mathscr{Q}$ to LBS provider. According to $BS_{\mathbb{C}}$ relevance, LBS look into the assisted database and returns top $\mathscr{K}$ candidate results to the TP given as

$$\mathrm{CR}: \left\langle \left(x_1, y_1\right), \left(x_2, y_2\right), \ldots, \left(x_k, y_k\right) \right\rangle, \tag{8}$$

where CR represents the candidate result.

Phase 3 (TP computation). TP preprocesses the data by multiplying all the items of candidate result set by a constant (-1) and encrypts this modified CR by mu's public key.

$$\mathscr{E}(\mathrm{CR}): \left\langle \mathscr{E}_{\mathrm{pk}}\left(x_1, y_1\right), \mathscr{E}_{\mathrm{pk}}\left(x_2, y_2\right), \ldots, \mathscr{E}_{\mathrm{pk}}\left(x_k, y_k\right) \right\rangle. \tag{9}$$

TP now has encrypted centroid coordinates $\mathscr{E}(X_{\mathbb{C}}, Y_{\mathbb{C}})$, and encrypted set of candidate results $\mathscr{E}(\mathrm{CR})$. The motive is to find the distance between the target point (centroid here) and the relevant points sent by the LBS provider so that the proximity of two can be measured. An additive homomorphic encryption is then applied to $(X_{\mathbb{C}}, Y_{\mathbb{C}})$ and each item of encrypted candidate result set separately given as

$$\begin{aligned} \mathscr{E}\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) \cdot \mathscr{E}_{\mathrm{pk}}\left(x_1, y_1\right) &= \mathscr{E}\left(\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) + \left(x_1, y_1\right)\right) \\ \mathscr{E}\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) \cdot \mathscr{E}_{\mathrm{pk}}\left(x_2, y_2\right) &= \mathscr{E}\left(\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) + \left(x_2, y_2\right)\right) \\ &\vdots \\ \mathscr{E}\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) \cdot \mathscr{E}_{\mathrm{pk}}\left(x_k, y_k\right) &= \mathscr{E}\left(\left(X_{\mathbb{C}}, Y_{\mathbb{C}}\right) + \left(x_k, y_k\right)\right). \end{aligned} \tag{10}$$

TP forwards the encrypted results and CR (in plain text) to mu. The purpose of having TP between mu and LBS is to perform certain computation such that the information retrieval becomes faster and relevant that too without losing any location privacy.

Phase 4 (decryption at mu). The mu has $\mathscr{K}$ encrypted values that can be viewed as the distances between the encrypted coordinates sent by $\mathbb{C}$ and the candidate result points sent by the LBS provider. mu deciphers them using her own secret key (sk). Let decryption gives the set of distances $\mathscr{D}$. Clearly, the minimum, $\min(\mathscr{D})$, among all distance values is the most relevant result. mu keeps the corresponding location

(1) Function: Communication using Hybrid System Model
(2) //*Phase 1: Ad hoc Congregation* $\mathbb{C}$
(3) *Let mobile user "mu" starts the query and* $\mathcal{K}$ *represents the number of users required to form* $\mathbb{C}$
(4) *Let* $\mathcal{S}$ *be the set to count numbers of neighbors responded*
(5) *Initially,* $\mathbb{C}(\mathcal{K}) = \emptyset$, $\mathcal{S} = 0$, $i = 0$
(6) *Let mu's actual location coordinates* $= (x, y)$
(7) $(x', y') = (x + \delta_x, y + \delta_y)$
(8) **while** $(i \leq \mathcal{K})$ **do**
(9) *mu broadcasts a CONGREGATE message to neighbors*
(10) *Let* $\mathcal{S}$ *users acknowledge mu*
(11) $i = |\mathcal{S}|$
(12) $\mathbb{C}(\mathcal{K}) = \mathbb{C}(\mathcal{K}) \cup$ *nodes in S*
(13) $i = i + |\mathcal{S}|$
(14) **return** $\mathbb{C}(\mathcal{K})$ //congregation formed
(15) **end**
(16) mu chooses a random node as congregation executor $E_{\mathbb{C}}$, $E_{\mathbb{C}} \in \mathbb{C}(\mathcal{K})$
(17) CALL Secret_Split_Function;
(18) *Let set* $X_p(\mathcal{K})$ *and* $Y_p(\mathcal{K})$ *holds the perturbed locations received after secret splitting*
(19) CALL Centroid_Function;
(20) $E_{\mathbb{C}}$ forwards $X_{\mathbb{C}}, Y_{\mathbb{C}}$ to mu and leaves $\mathbb{C}$;
(21) mu generates (pk, sk) pair
(22) $\mathscr{E}_{\text{pk}}(X_{\mathbb{C}}, Y_{\mathbb{C}})$ //encrypted points
(23) //*Phase 3: Computation performed at TP*
(24) (X_L, Y_L) = CALL TP_Computation-I;
(25) $\mathscr{E}_{\text{pk}}(X_L, Y_L)$ //Encryption using mu's pk
(26) CALL TP_Computation-II;
(27) //Phase 4: Decryption at mu
(28) $\mathscr{D}_{\text{sk}}((X_{\mathbb{C}} + X_L), (Y_{\mathbb{C}} + Y_L))$ //Decryption using mu's sk
(29) *Let* X_D, Y_D *be the set of distance difference received on decryption*
(30) CALL Min_dist (X_D, Y_D)
(31) *Broadcast Results to all members of* $\mathbb{C}$

Algorithm 1: HYB solution.

coordinate against min($\mathscr{D}$) and sends remaining results to all the members of $\mathbb{C}$.

Considerations and Assumptions

(a) The utilized mobile devices are Location Based Services enabled and have the ability to determine their approximate location.

(b) The TP possess required computation power and processing potential.

(c) Location queries are sporadic, pull-based, and specific in nature.

(d) Generation of ⟨Public-Private⟩ key pair at mu is implicit.

Algorithm Description. The algorithm, HYB solution, gives pseudocode for the overall communication of our proposed hybrid system model. A congregation is formed (lines (7)–(15) in Algorithm 1), a pair of coordinates are computed (lines (16)–(19) in Algorithm 1), and the encryption is performed (lines (21)–(23) in Algorithm 1) over computed coordinates during Phase 1 of HYB solution. Phase 2 fetches the candidate result from LBS to TP. In Phase 3, candidate result is first modified (line (24) in Algorithm 1) and then encrypted (line (25) in Algorithm 1) before applying homomorphic operation (line (26) in Algorithm 1) over encrypted inputs. Decryption is performed in Phase 4 (line (28) in Algorithm 1) and the minimum is calculated (line (30) in Algorithm 1) to get optimum result. Algorithms 2, 3, 4, 5, and 6 give the pseudocodes for the suboperations: splitting the random secret, centroid computation, input preprocessing, homomorphic encryption, and finding minimum value from the result set, respectively.

6. Empirical Evaluation

We develop the simulation scenario and implemented the same in Java. We run it on an Intel Core 3.20 GHz machine with 4 GB of RAM running Linux OS. We experimented the

(1) Function: Secret Splitting Sharing
(2) $E_{\mathbb{C}}$ *chooses and split sufficiently large two random shares* $\mathcal{S}_x$ *and* $\mathcal{S}_y$ *s.t.*
$\sum_{i\in\mathbb{C}, i=1 \text{ to } \mathcal{K}} \mathcal{S}_{i,x} = 0, \quad \sum_{i\in\mathbb{C}, i=1 \text{ to } \mathcal{K}} \mathcal{S}_{i,y} = 0$
(3) $E_{\mathbb{C}}$ *sends separate split values to every node* $\in \mathbb{C}(\mathcal{K})$
(4) **foreach** node $\in \mathbb{C}(\mathcal{K})$ **do**
(5) **for** $i = 1; i_{\mathbb{i}}=\mathcal{K}; i++$ **do**
(6) $(x_p, y_p) = (x_i + \mathcal{S}_{i,x}, y_i + \mathcal{S}_{i,y})$
(7) **end**
(8) **return** (x_p, y_p)
(9) **end**

Algorithm 2: Secret_Split_Function.

(1) Function: Centroid Computation
(2) **foreach** $x \in X_p(\mathcal{K})$ *and* $y \in Y_p(\mathcal{K})$ **do**
(3) $i = 1, Temp_x = Temp_y = 0$
(4) **while** $i <= \mathcal{K}$ **do**
(5) $Temp_x = Temp_x + x_i;$
(6) $Temp_y = Temp_y + y_i;$
(7) $i++;$
(8) **end**
(9) $X_{\mathbb{C}} = \frac{Temp_x}{\mathcal{K}}, Y_{\mathbb{C}} = \frac{Temp_y}{\mathcal{K}}$
(10) **return** $(X_{\mathbb{C}}, Y_{\mathbb{C}})$
(11) **end**

Algorithm 3: Centroid_Function.

(1) Function: Coordinate Pre-processing
(2) *Let set* $X_l(\mathcal{K})$ *and* $Y_l(\mathcal{K})$ *be the points provided by LBS*
(3) *Let set* X_l' *and* Y_l' *be the points modified by TP*
(4) *Initially*, $i = 0, X_l' = Y_l' = 0$
(5) **foreach** $x \in X_l(\mathcal{K})$ *and* $y \in Y_l(\mathcal{K})$ **do**
(6) **while** $i \leq \mathcal{K}$ **do**
(7) $x_i' = (-1) * x_i, y_i' = (-1) * y_i;$
(8) $i++;$
(9) **end**
(10) $X_l' = X_l' \cup x_i', Y_l' = Y_l' \cup y_i'$
(11) **end**
(12) return (X_l', Y_l')

Algorithm 4: TP_Computation-I.

(1) Function: Computing point differnce
(2) *input*: $X_{\mathbb{C}}, Y_{\mathbb{C}}$ *and* X_L, Y_L
(3) Apply *Paillier* Homomorphic Encryption
(4) **return** $((X_{\mathbb{C}} + X_L), (Y_{\mathbb{C}} + Y_L))$

Algorithm 5: TP_Computation-II.

(1) Function: Finding location with minimum distance
(2) *Let* MIN *represents the minimum element of the list,* $i = 2$ **foreach** *element of* X_D, Y_D **do**
(3) (X_1, Y_1) = MIN **while** $i <= \mathcal{K}$ **do**
(4) **if** (X_i, Y_i) < MIN **then**
(5) MIN = (X_i, Y_i)
(6) $i++;$
(7) **end**
(8) **else**
(9) $i++;$
(10) **end**
(11) **end**
(12) **return** MIN;
(13) **end**

Algorithm 6: Min_Dist.

Table 1: Parameters used with description.

Parameter	Description	Values used
$\mathcal{K}$	Anonymity parameter	1, 5, 10, 20, 40, 100, 150, 200
$\mathcal{N}$	Key size (in bits)	512, 1024, 2048, 4096
Review period	Time interval between two consecutive runs of the algorithm	90 s
Total run count	Number of times the algorithm runs for a particular combination of parameters used	100
Input size	Size of an input item	4 KB

performance with different variations in anonymity parameter and key size. Performance metrics is measured in average computation time taken by the processes.

6.1. Parameters Description. Results are evaluated for different values of parameters. Table 1 highlights the brief description of the parameters used.

6.2. Anonymity Parameter and Key Size Impact over TP Computation-II. The first experiment explores the impact of anonymity parameter with different key sizes over the performance of the system in terms of the computation time. The algorithm TP Computation-II computes the homomorphic encryption.

Analysis. Figure 6 shows the average time taken by TP to perform operations over encrypted data. It can be seen that time taken is very less (less than a second) for those combinations where key size ($\mathcal{N}$) and $\mathcal{K}$ are low. As we move left to right through x-axis in the graph, the time increases beyond acceptable threshold and makes the framework costly in terms of time for higher values of $\mathcal{N}$ and $\mathcal{K}$.

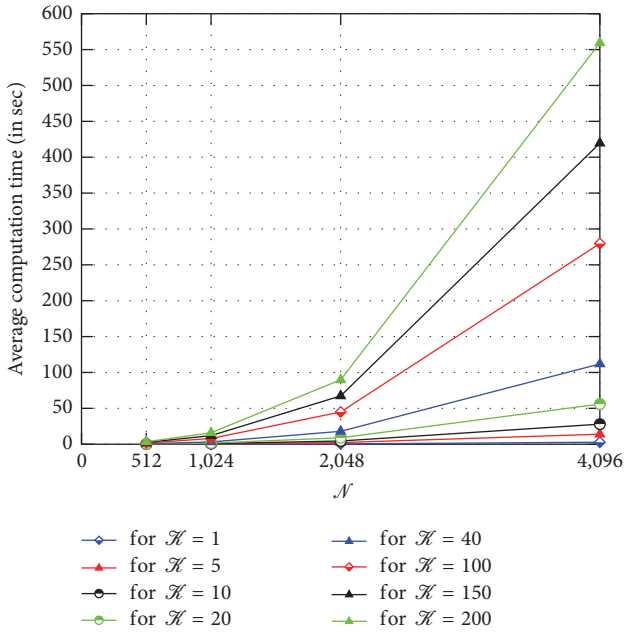

Figure 6: Anonymity parameter and key size impact over TP Computation-I.

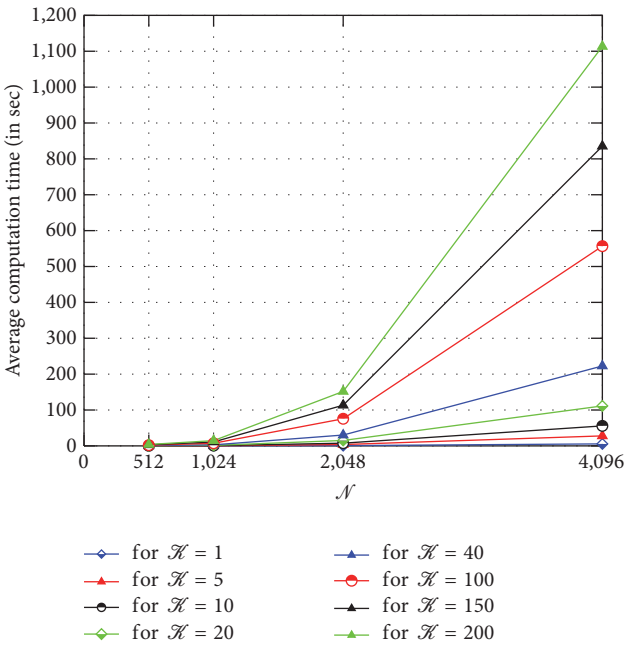

Figure 7: Anonymity parameter and key size impact over decryption.

6.3. Anonymity Parameter and Key Size Impact over Decryption Computation at mu. This evaluation shows the time taken to decrypt the encrypted results. Decryption is performed using mu's secret key which is secure and not shared with any other party.

Analysis. Figure 7 shows the average computation time for decryption. The effect of $\mathcal{K}$ and $\mathcal{N}$ is more or less similar as in the case discussed before. It is clear that computation time is lesser for smaller $\mathcal{K}$ and $\mathcal{N}$ values; on the other hand, computation cost becomes exorbitantly expensive for higher $\mathcal{K}$, $\mathcal{N}$ values combination.

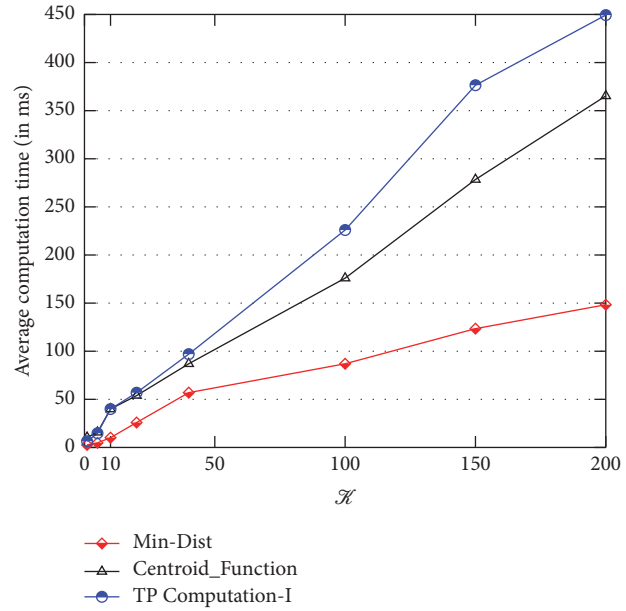

Figure 8: Miscellaneous computation time dependence over anonymity parameter $\mathcal{K}$.

6.4. Effect of Size of ζ over Miscellaneous Computation. Min-Dist is used to calculate the minimum among all the values received after decryption. TP Computation-I preprocesses the input and Centroid Function computes the centroid of locations. These processes also contribute to the overall time of HYB solution.

Analysis. Figure 8 shows that, for lower $\mathcal{K}$ values, the computation time is lower. However, time taken for higher $\mathcal{K}$ (150 and 200) is much lesser compared to the time taken by TP Computation-II and becomes less significant when added to the overall computation cost.

The value of $\mathcal{K}$ specified by the mobile user mu and the key size used for encryption impacts the overall computation time to a large extent. The balanced combination of these two parameters produces the optimum results. Moreover, the public key encryption enabled the secure communication as no key distribution is now needed. As the location data is encrypted under mu's public key and decryption takes place at mu with the secret key she has, it makes the overall solution secure and reliable.

7. Conclusion

This paper first addressed the issues in TTP based and TTP-free frameworks and presented a hybrid solution that makes effective use of the advantages both the approaches possess, to preserve location privacy of the user through congregation and homomorphic encryption. The novelty of

the proposed HYB solution lies in the fact that involvement of third party is introduced to perform computations only and TP has no knowledge of the user's real location. A congregation scheme is also suggested that helps the mobile user to compute centroid of all the users involved, that too without knowing anyone's actual location. Homomorphic encryption technique is used with a modified input data in order to take most out of it. We have analyzed the performance of our model for various key sizes and for different values of anonymity parameter. Our scheme works well when key size and anonymity parameter are in a certain range. The proposed HYB model preserves the user's location privacy at two levels, namely, at *proximity* level, while forming congregation, and at *distant* level, while sending encrypted location to TP.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] D. Wells, N. Beck, A. Kleusberg et al., *Guide to GPS Positioning*, Canadian GPS Associates, New Brunswick, Canada, 1987.

[2] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS vulnerability to spoofing threats and a review of antispoofing techniques," *International Journal of Navigation and Observation*, vol. 2012, Article ID 127072, 2012.

[3] M. F. Mokbel, "Privacy in location-based services: state-of-the-art and research directions," in *Proceedings of the 8th International Conference on Mobile Data Management (MDM '07)*, p. 228, IEEE, Mannheim, Germany, May 2007.

[4] J. Krumm, "Inference attacks on location tracks," in *Pervasive Computing*, pp. 127–143, Springer, Berlin, Germany, 2007.

[5] D. Song and K. Park, "A privacy-preserving location-based system for continuous spatial queries," *Mobile Information Systems*, vol. 2016, Article ID 6182769, 9 pages, 2016.

[6] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[7] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, pp. 763–774, VLDB Endowment, 2006.

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

[9] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Mobihide: a mobilea peerto-peer system for anonymous location-based queries," in *Advances in Spatial and Temporal Databases*, pp. 221–238, Springer, Berlin, Germany, 2007.

[10] E. K. Wang and Y. Ye, "A new privacy-preserving scheme for continuous query in location-based social networking services," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 979201, 2014.

[11] M. Zhou, X. Li, and L. Liao, "On preventing location attacks for urban vehicular networks," *Mobile Information Systems*, vol. 2016, Article ID 5850670, 13 pages, 2016.

[12] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: robust location privacy scheme for VANET," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1569–1589, 2007.

[13] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux, "Mix-zones for location privacy in vehicular networks," in *Proceedings of the ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS '07)*, Vancouver, Canada, 2007.

[14] F. Kargl and J. Petit, "Security and privacy in vehicular networks," in *Vehicular Communications and Networks: Architectures, Protocols, Operation and Deployment*, pp. 171–189, 2015.

[15] Y. Gai, J. Lin, and B. Krishnamachari, "Security and privacy in vehicular networks," *Cognitive Vehicular Networks*, pp. 151–166, 2016.

[16] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS '06)*, pp. 171–178, ACM, November 2006.

[17] M. Langheinrich, "Privacy by designprinciples of privacy-aware ubiquitous systems," in *Ubicomp 2001: Ubiquitous Computing*, pp. 273–291, Springer, Berlin, Germany, 2001.

[18] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "Prive: anonymous locationbased queries in distributed mobile systems," in *Proceedings of the 16th International Conference on World Wide Web*, pp. 371–380, ACM, 2007.

[19] H. Zhangwei and X. Mingjun, "A distributed spatial cloaking protocol for location privacy," in *Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC '10)*, vol. 2, pp. 468–471, April 2010.

[20] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, pp. 901–914, ACM, Berlin, Germany, November 2013.

[21] R. Koorn, H. van Gils, J. ter Hart, P. Overbeek, R. Tellegen, and J. Borking, *Privacy Enhancing Technologies, White Paper for Decision Makers*, 2004.

[22] C. A. Ardagna, M. Cremonini, S. De Capitani Di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13–27, 2011.

[23] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 121–132, ACM, June 2008.

[24] A. Khoshgozaran, H. Shirani-Mehr, and C. Shahabi, "SPIRAL: a scalable private information retrieval approach to location privacy," in *Proceedings of the 9th International Conference on Mobile Data Management Workshops (MDMW '08)*, pp. 55–62, April 2008.

[25] A. Khoshgozaran and C. Shahabi, "Private information retrieval techniques for enabling location privacy in location-based services," in *Privacy in Location-Based Applications*, pp. 59–83, Springer, 2009.

[26] J. Domingo-Ferrer, "Microaggregation for database and location privacy," in *Next Generation Information Technologies and Systems*, pp. 106–116, Springer, 2006.

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

[28] A. Solanas and A. Martínez-Ballesté, "A TTP-free protocol for location privacy in location-based services," *Computer Communications*, vol. 31, no. 6, pp. 1181–1191, 2008.

[29] R. Rothblum, "Homomorphic encryption: from private-key to publickey," in *Proceedings of the Theory of Cryptography Conference*, pp. 219–234, Springer, Providence, RI, USA, March 2011.

[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '99)*, pp. 223–238, Springer, Prague, Czech Republic, 1999.

[31] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," in *International Workshop on Public Key Cryptography*, pp. 119–136, Springer, 2001.