International Journal of Geo-Information

MDPI

*Article*

# A Symbiotic Relationship Based Leader Approach for Privacy Protection in Location Based Services

**Hosam Alrahhal [1,2], Mohamad Shady Alrahhal [3,*], Razan Jamous [2] and Kamal Jambi [3]**

1 Communication and Computer Engineering Department, Faculty of Engineering, NAHDA University, Beni Suef 62511, Egypt; hosamrahhal@gmail.com
2 Faculty of Engineering and Applied Science, University of Regina, Regina, SK S4S 0A2, Canada; rjw836@uregina.com.ca
3 Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; kjambi@kau.edu.sa
* Correspondence: shady.rahal1986@gmail.com

Received: 27 May 2020; Accepted: 23 June 2020; Published: 26 June 2020


**Abstract:** Location-based services (LBS) form the main part of the Internet of Things (IoT) and have received a significant amount of attention from the research community as well as application users due to the popularity of wireless devices and the daily growth in users. However, there are several risks associated with the use of LBS-enabled applications, as users are forced to send their queries based on their real-time and actual location. Attacks could be applied by the LBS server itself or by its maintainer, which consequently may lead to more serious issues such as the theft of sensitive and personal information about LBS users. Due to this fact, complete privacy protection (location and query privacy protection) is a critical problem. Collaborative (cache-based) approaches are used to prevent the LBS application users from connecting to the LBS server (malicious parties). However, no robust trust approaches have been provided to design a trusted third party (TTP), which prevents LBS users from acting as an attacker. This paper proposed a symbiotic relationship-based leader approach to guarantee complete privacy protection for users of LBS-enabled applications. Specifically, it introduced the mutual benefit underlying the symbiotic relationship, dummies, and caching concepts to avoid dealing with untrusted LBS servers and achieve complete privacy protection. In addition, the paper proposed a new privacy metric to predict the closeness of the attacker to the moment of her actual attack launch. Compared to three well-known approaches, namely enhanced dummy location selection (enhanced-DLS), hiding in a mobile crowd, and caching-aware dummy selection algorithm (enhanced-CaDSA), our experimental results showed better performance in terms of communication cost, resistance against inferences attacks, and cache hit ratio.

**Keywords:** cache; attack launch; dummies; leader; privacy protection; reputation

## 1. Introduction

Under swift and mesmerizing developments in the world of technology and Internet networking, specifically the commercial success of mobile devices, lives of people have become easier and more enjoyable. Location-based services (LBS) form a main part of the Internet of Things (IoT) [1–6], where a wide spectrum of IoT applications relies on LBS, including smart cars, wearable devices (smart watches, sleep tracker bracelets, clothes, etc.), and reward-based LBS applications [7–9]. Moreover, in the e-Health field, LBS plays a significant role in monitoring the patient's health conditions (pulse rate and blood pressure level) and avoiding disasters [10,11]. A further advantage of LBS is enabling people to search for points of interests (POI) such as nearby restaurants, hotels, hospitals, and sport clubs.

### 1.1. Statement of Problem

Using LBS requires sending queries based on the real geographical locations of LBS users, where LBS users obtain their real locations through GPS. After manipulating these queries from the service provider, the results are returned to the users as shown in Figure 1.

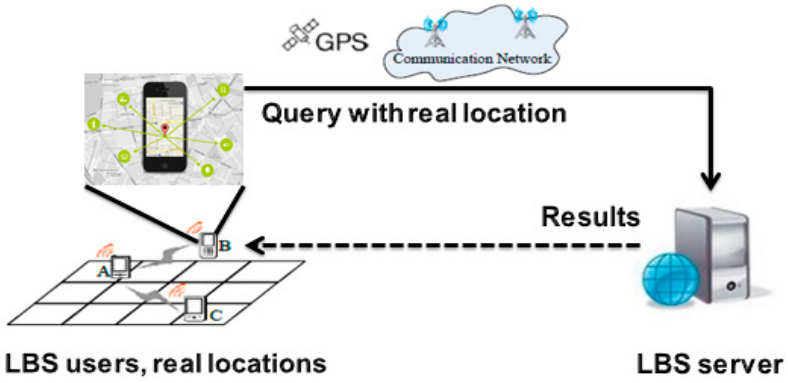


**Figure 1.** The classical scenario of LBS applications.

This simple and traditional scenario includes risks, even as users are carried away by the advantages of LBS. The underlying reason behind these risks is that the services the LBS users are wanting to use, and the places they are most likely visiting or trying to find, reflect important aspects that are directly related to their personal lives (such as their customs, habits, or religious persuasion). Furthermore, in light of existing advanced methods that could be used to track users, such as [12,13], gathering private information has become more serious. In the work [12], the authors presented a survey on indoor wireless tracking of mobile nodes from a signal processing perspective. In addition, they stated that it will not be surprising if we witness a widespread use of indoor tracking technologies to complement and empower pedestrian and vehicular systems in the fields of intelligent transportation systems, automated vehicles, robotics, and location-based services. Zhang, et al. [13] developed Montage for real-time multi-user formation tracking and localization. Montage achieves high tracking accuracy by integrating temporal and spatial constraints from user movement vector estimation and distance measuring. Beyond tracking, the authors in [14] showed that such information on these sensitive aspects could be obtained, as attackers could track the locations of users or analyze their queries. After gathering sensitive data about the victim, the attacker can establish and trigger an actual attack in several forms, such as burglary, blackmail, or mugging. In the worst case, if the LBS server or the LBS server maintainer himself is the attacker, the danger will have more of a negative impact on privacy since all information related to the activities of the LBS users is accessible. Thus, privacy protection is a problem of great importance, and the need for a revolution in privacy protection methods is pressing.

### 1.2. Motivation

To address this problem and to protect the privacy of LBS users, researchers have proposed several approaches. The solutions were addressed from different perspectives, namely the server-side, user-side, and the interactive cooperation between both server-and user-sides. Figure 2 is a classification of LBS privacy protection approaches, where each category has its drawbacks.

Regarding the server side, the consideration taken into account is that dealing with a server is inevitable when it comes to attaining the benefits from high computational capabilities and huge storage. Different approaches to providing privacy protection on the server level have been presented [15–23]. Xu, et al. [24] proposed a new way to protect the privacy of LBS users by applying temporal-spatial masks for user locations, where the server acts as an anonymizer. However, according to the sensitivity of the application the LBS user uses, the server is considered a malicious party (i.e., an attacker) that has

the ability to track the motion trajectories of the LBS user and compromise privacy. Consequently, there are no satisfied guarantees regarding the total and absolute reliance on the LBS server. Beyond that, the LBS server could be considered a valuable data center in the eyes of the attacker, as all information related to the motion trajectories of the LBS users or those that describe the POI they prefer are stored in it. In other words, attracting the attention of the attackers enables them to exert less effort and minimizes time needed to initiate the attack on the victim.

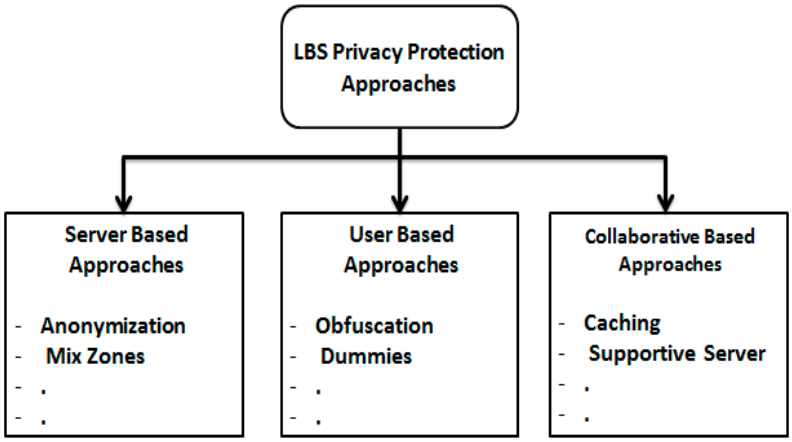


**Figure 2.** Classification of LBS privacy protection approaches.

From the user perspective side, to handle such critical issues, some researchers have changed their incentives and have focused on the user side (avoiding dealing with a trusted third party (TTP)). From their point of view, the consideration that must be taken into account is that the LBS user himself can determine the privacy level or even have complete control over it; the user has more awareness about where and when he will utilize a high privacy level and ask for a POI. Many proposed approaches have been provided in this aspect in [25–31]. In general, even these approaches avoid dealing with TTPs, but they suffer from many issues related to mobile device capacity storage limitations, low computational capabilities, and short battery life. In particular, the approaches contained in [32–35] have another major problem related to dummies generation, which is considered an open problem according to [36]. In the context of LBS privacy protection, dummy is a term that refers to a set of queries built on false locations. Since the responsibility of dummy generation is assigned to the LBS users, producing weak dummies will make them easy victims, as the attacker can easily filter weak dummies, determining the identities of the LBS users. These problems changed the direction of the research.

These new tactics depend on the principle of cooperative interaction between both LBS users, and the LBS server is proposed in [36,37]. In this category, the LBS user will take responsibility for privacy management with help provided from the LBS server aspect. Although, the LBS server helps either by arranging the data portable within the transmitted channel or supplying the LBS user with proactive information about the degree to which LBS user privacy is broken, these approaches still depend on the LBS server and the mission assigned to it, which refers to the drawbacks related to server-based category.

One of the most important cooperative ways in which LBS users can avoid dealing with TTPs is minimizing the connecting number with the LBS server [38–40]. The key idea relies on the cache, where the Responses of the Queries (RoQ) stored previously are exploited to answer future queries. Therefore, the LBS user tries to find the answer to his/her query in the cache, and if he/she finds his/her query answer in the cache, it is considered acceptable. Otherwise, the LBS user is forced to connect to the LBS server. However, the choice of connecting to the LBS server is still standing. This in turn means that the LBS user will be in a critical situation in case of tracking for a long time by the LBS server. Moreover, no strong trust basis can prevent users from turning to an attacker. Furthermore, the quality of the RoQ may be weak, leading to a poor system response performance.

### 1.3. Contribution

Focusing on cache-based approaches, a leader (that acts as a TTP) can decrease the connecting numbers to the LBS server and optimize the quality of the RoQ stored in the cache at the same time. Through building trust between LBS users and a Leader, all LBS users can be prevented from connecting to the LBS server. Implementing a leader (TTP), through which the queries of the LBS users are sent to the LBS server, means that the leader can protect his/her privacy without the need for dummies generation; real queries will be exploited as dummies on the leader side. Moreover, compared to the RoQ built on dummies, the responses of these exploited queries will be the actual results for what the LBS users are searching for. As a result, the cache will be filled with valuable information that contributes to increasing the probability of answering future queries. This in turn optimizes the response time of the system since it shortens the time of query manipulation due to the locality concept.

In this paper, a solution to optimize the privacy protection of LBS users is presented. The proposed solution is inspired by nature and depends on the symbiotic relationship exploiting the mutual benefit that could occur among animals (birds that search for food inside the opened jaws of a crocodile, for example). The projection of mutual benefit phenomenon will lead to great trust between the members of a cluster and the leader. The cluster members will be able to avoid connecting to the LBS server (a malicious party). At the same time, the leader will exploit the real queries with real positions as dummies to gain full privacy protection at his/her side. To know how the privacy of the leader is broken, we proposed a new privacy metric that could be considered as a standard metric.

In general, the contributions of this work are as follows:

- The paper proposes a leader approach to completely prevent LBS users (members of a cluster) from connecting to the untrusted party (LBS server). A symbiotic relationship is used to form the trust base between the cluster members and their leader. Consequently, the leader is considered a strong TTP.
- The paper introduces a solution to the dummy generation problem, which is considered as an expensive and open problem for achieving comprehensive privacy protection (i.e., location and query privacy protection).
- Depending on location entropy, a novel privacy metric is provided. It is used to measure the closeness of the attacker to the moment of his/her attack launch.
- To show the robustness of the proposed approach in terms of communication cost, resistance against inference attacks, and cache hit ratio, three well-known approaches, namely enhanced-DLS [34], hiding in a mobile crowd [38], and enhanced-CaDSA [40] are studied and compared.

The remainder of the paper is organized as follows: Section 2 contains a literature review. In Section 3, we present the proposed solution followed by the evaluation metrics in Section 4. Section 5 collects our experimental results with the evaluations. Finally, we conclude the paper in Section 6.

## 2. Related Work

Under the threat that LBS users would make complaints related to sacrifices to their privacy, researchers responded by building defenses against attackers. These defenses were expressed through various proposed approaches in the domain of LBS privacy protection. Many efforts were made to classify the proposed privacy protection approaches, and these classifications were taken from different points of view according to their objectives [15], topologies of location [41], or structure features [14]. For example, the classification provided in [15] limited privacy protection in the protection of user identity, spatial information, and temporal information.

In general, the literature review provided in this paper classifies privacy protection techniques into three main categories, where it basically depends on the amount of collaboration between the two major aspects involved in any LBS privacy protection system (i.e., LBS users and LBS server). Before starting, it should be mentioned that most of the proposed techniques intersect in one root, which is

applying the k-anonymity concept in different ways. The basic essence of k-anonymity is to confuse the attacker about determining the identity of the query issuer among other (k–1) users.

### 2.1. First Group: Most of The Load on The Server Side

In this group, the LBS server is mainly responsible for protection approach execution, while the user mission is only to send his/her query. This group assumes that the LBS server must be reliable.

The authors in [16] based their work on a signature for privacy protection, where they proposed a message signing algorithm to achieve their goal. The main feature of the message signing algorithm was enhancing the efficiency of authenticity verification on a large number of messages exchanged among users. Moreover, a semi-trusted LBS server architecture was proposed in [17], which is based on a location clocking algorithm (LCA) to protect the real location of the user. The major contribution of LCR algorithm was to minimize the anonymizing spatial region (ASR). Thus, the number of real users is kept close to the k – anonymity level even if there are a few users within the anonymized region. Gedlik and liu presented a personalized k-anonymity approach called CliqueCloak [18], where the LBS server acts as an anonymizer with respect to user demands. To protect privacy, spatial-temporal masks (cliques) are applied on the positions of the user by providing a controlled k-anonymity level. The attractive feature of the CliqueCloak approach is allowing LBS users to have individual and maximum limits on the spatial-temporal properties of the masks (i.e., an individual level of tolerance). Similar to [18,19] provided spatial-temporal masks for users located in a given region. However, the difference was that this approach played on the resolution of these masks by modifying the spatial-temporal dimensions of the masks to meet certain conditions and to achieve a k – anonymity concept at the same time. Thus, it ignored the level of tolerance, focusing on the resolution. Similarly, the authors of [20] suggested another personalized approach called Casper. Compared to CliqueCloak, the conditions of privacy protection were driven from the profile of the LBS user. One of the most remarkable techniques used in this group was proposed in [21] and is called mix zones. The users located in a given area are grouped within many spatial regions (zones), and each zone is assigned to one pseudonym. Then, the zones are mixed to guarantee both conditions, which means no location updates inside a mix zone during the moving of objects and the user must utilize the pseudonym of the new group when leaving one zone for another.

Another approach, presented in [22], targeted the confusion of attackers and minimized his ability to gather historical information related to the trajectories of the users during their motion. This approach used a perturbation algorithm exploiting the path intersections or those close to each other. Meyerowitz et al. achieved real-time location privacy protection by using the CacheClock approach as presented in [23]. The CacheClock approach is based on the intersection of paths to predict new paths, where the real position of the LBS user will be located in one of these predicted paths. So, the real position is masked by other paths.

The motivation of [24] is built upon the idea that privacy is about feeling, and it is awkward for LBS users to scale their feeling using a number (i.e., deciding a high k value for the k-anonymity concept). So, instead of deciding on a k value, the LBS user decides on the popularity of the area where he resides. The popularity of an area, shopping malls as an example, ensures privacy protection, where we have a lot of visitors or LBS users, as the attacker needs specific personal information about the LBS user, not public information.

### 2.2. Second Group: Most of The Load on The User Side

In this group, the privacy technique is executed on the mobile devices of the LBS users, and privacy management is controlled by the users themselves, where the LBS server is considered a malicious component.

Because untrusted LBS servers were avoided, the authors in [25] faced two main issues. First, user privacy could be attacked based on the inferred data from the issued query. Second is the user privacy level, which trades off between privacy protection and LBS response accuracy. The storing geographic

map along with perturbation-based protection are proposed to solve the privacy issue on the user side, and then various-grid-length Hilbert curve (VHC) mapping is adopted to convert the two dimensions of the stored map into one dimension depending on the context of the map homogeneity as a solution to robust privacy protection and as a way to maintain LBS response accuracy.

The Framework called MobiMimosa was provided in [26], which helps to protect the sensitive data stored in smartphones. Since the cost of the encryption and decryption process is considered a high computational cost, MobiMimosa tried to minimize this cost by providing a plausibly deniable encryption. Another mobile-device-based approach called SMILE was suggested to solve the untrusting issue of the LBS server from a purely social point of view [27]. It applied k-anonymity to measure and configure the users' privacy level for the use of encounter-based LBSs. This approach protects user privacy by selecting the prefix length of the location hash value to avoid revealing encounter involvements with untrusted servers. A k-anonymous cloaking boxes approach was proposed in [28] based on blurring the coordinates of whole active users' real positions to build such clocking boxes. This is unacceptable under untrusting terms. The strength of received Wifi signals is employed to build the clocking boxes instead of disclosing the real coordinates of the users. The authors of [29] relied on minimizing the attacker ability to infer private information from the motion patterns of the users, proposing the M-unobservability term. It coats the user position with noise, which in turn limits the recognition of the POIs that the users mostly visit. Ardagna et al. [30], based on a spatial obfuscation technique, proposed a technique to make the real and clear position of the user a coarse one mathematically. Location privacy protection was achieved by sending a circular aria instead of the accurate position of the LBS user. The mathematical operations included radius enlarging, center shifting, radius increasing, or applying double obfuscation (i.e., mixing center shifting with any remainders). Similarly, the authors of [31] developed the coordinates level approach that depends on coordinate transforms. The resultant coordinates will form the new user position instead of his real one. Then, inverse transforms could be easily applied to obtain the original positions on the LBS user side.

Kido, et al. [32] provided the dummies idea to protect the privacy of the LBS user. The key idea was that the user creates many false positions (dummies), building instances of the current query using both the dummies and the true position of the user, and then sends all of the copies to the LBS server asking for the same POI. Randomizing the real position among dummies will ensure privacy protection, where the LBS server cannot recognize the real position among dummies. Pingley et al. discussed query privacy protection against inference attacks that could be applied to the queries sent to the LBS server [33]. The region where the user is located was exploited to create dummy queries by modifying the features of the real query. To create strong dummies, historical query logs related to other users were used to import new features to contribute to dummies building. So, the difference from [32] was saving the same real location of the LBS user and changing the features of the current query itself. The authors of [34] provided an enhanced dummy location selection algorithm called enhanced-DLS to generate dummy locations. The difference was in two points: 1) It took into account the probability of exploiting the (side information) on the attacker side, and 2) choosing dummies carefully to obtain an optimal degree of *k-anonymity*. Among a candidate set of locations, choosing dummies in suitable places depended on an entropy metric. Hara, et al. [35] proposed another approach related to the dummies idea, taking into account the physical constraints of the real world. The feature that distinguished this work was that the trajectories of the generated dummies cross the trajectories of the actual movement of the LBS user. To protect the privacy of the LBS user, the crossing process acts under two conditions: i) If there is no dummy ahead of the user, there is no change in the dummies' destinations (i.e., no crossing is performed). ii) If some dummies move ahead of the user, the crossing process is performed.

### 2.3. Third Group: Load Balancing

In this group, the user either makes her privacy protection decision based on the help provided by the LBS server side or connects with the LBS server in the case where no query answer is found in the cache. In this group, the LBS server is considered an attacker also.

Authors of [36] adopted point-to-point access along with building an air index (NPI) list within the connecting channel. The server indexed the data segments before broadcasting. The indexing data carry information about the cells of the region the users are located within. The periodic transmission of indexed data ensures privacy protection. On the user side, the NPI-based algorithm is applied to answer the queries. A privacy-supportive LBS server structure was proposed in [37], helping the user to make his own privacy decision. The key idea depends on building an LBS server structure that provides an immediate trade-off between the privacy and usage. In this structure, the LBS server provides auxiliary information to the user to support his privacy decision so that he will be aware of risks about his achieved privacy level. This helped the user to create his queries carefully.

Shokri et al. [38] proposed the idea of collaboration among LBS users to avoid dealing with the LBS server. Privacy protection is achieved by answering queries within the mobile crowd. Their idea is based on storing the query responses in the cache of each mobile device of each user. If a user wants to query about a POI, it tries to obtain the answer by connecting with other users. The user will be forced to connect to the LBS server in case no answer is kept by the other peers. The drawbacks of ref. [38] are solved by the enhanced dummy selection algorithm (enhanced-DSA) proposed in [39]. The mobile devices cache stores the queries' answers so that the interaction among users will prevent them from dealing with untrusted LBS servers. The cache usage combined with dummies to achieve two main goals. The first was achieving k-anonymity level via dummies to protect the privacy of the LBS users. The second was minimizing the probability of connecting to the untrusted LBS server by selecting the dummies' locations that have more contributions in the caches of mobile devices. Instead of using the caches of mobile devices, the authors of [40] used the Access Point (AP) to represent the cache. So, they proposed a caching-aware dummy selection algorithm (CaDSA) integrated with the cache. The idea of CaDSA algorithm was directly inspired by the ideas of both enhanced-DLS algorithm presented in [34] and the enhanced-DSA algorithm presented in [39]. Compared to the enhanced-DLS and enhanced-DSA algorithms, the two main features of CaDSA algorithm are: 1) it used normalized distance to ensure that the selected dummies were optimal, and 2) it used a data freshness term to keep the most important queries' responses for a long time, which in turn enhanced answering future queries.

## 3. Proposed Privacy Protection Approach

This section expresses the scenario of the proposed approach where the criteria of the leader election is highlighted, and provides the answer to the following question: What if the elected leader behaves as an attacker?

### 3.1. Proposed Approach (Leader)

For a given region divided into $(n \times n)$ cells, a number of users are distributed over this region so that $(g)$ users are located in each cell, and each cell includes $(p)$ POIs. The general scenario, which cache-based approaches follow to minimize the connecting number to the untrusted LBS server is illustrated by Figure 3.

In Figure 3, the traditional way (i.e., a user who is not concerned about privacy) is to send real queries with real positions to the LBS server with a direct privacy threat, denoted by the dotted line (asking nearest hotels for an example). To protect location privacy, the LBS user deliberately issues many queries with dummy positions asking for the same POI (i.e., the nearest hotels) denoted by continuous lines. LBS server responses are cached to obtain benefits from answering incoming queries with the progress of time (i.e., future queries). In cases where no answer is found in the cache, the LBS

user is forced to connect with the untrusted LBS provider denoted by the dashed lines. However, query privacy protection is not assured where query analysis-based attacks can be applied. To have complete prevention for all LBS users, except the leader, from connecting to the untrusted LBS server, the leader approach is proposed. The essence of our idea depends on the mutual benefit between the leader and the other LBS users. However, the users that are located in each cell will be grouped in one cluster. From there, a leader will be elected for each cluster. For a query issuer, in case a query answer is not found in the cache, the issuer will send the query to the leader (instead of sending the query to the LBS server directly). Then, the leader in turn sends it to the untrusted LBS server. After manipulating the query on the LBS server side, the leader receives the answer and then returns the received answer to the wanted user (i.e., the query issuer). In other words, the leader exploits the real queries built on real positions (and sent by the cluster members) as dummies on his side. Then, the Leader will send his own real query with a real position without any need to produce dummies either at the location or query level. Considering three LBS users searching for three different POIs (nearest hotels, restaurants, and hospitals), Figure 4 provides a comprehensive look at the proposed system model, where the responses of queries will be stored in the cache to be used later.

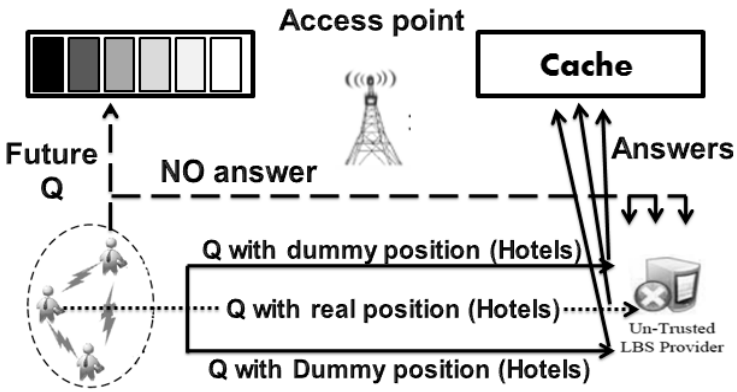


**Figure 3.** General scenario of cache-based approach.

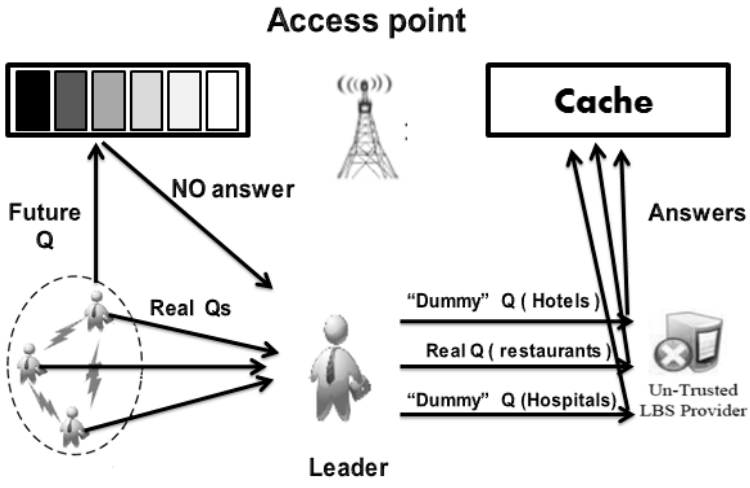


**Figure 4.** Proposed system model scenario.

According to Figure 4, LBS users will protect their privacy (both location and query privacy) because there is no need to directly connect to the LBS server even if they do not find answers to queries within the cache. Thus, they will achieve full privacy protection under any privacy metric.

Without the need to create false locations (i.e., dummy locations) or query feature tampering to generate dummy queries, the privacy of the leader will be protected since the Leader exploits queries of the cluster member as dummies. In addition, the leader has no need to use normalized distance

mentioned in [40] because LBS users will be mostly located in the cells that contain POIs users likely used to search for. However, in all cases, the attacker (LBS server) will be confused about determining the query issuer (the Leader). Moreover, even if the attacker applied analysis to those queries sent by the leader to infer some auxiliary information to launch his attack, his efforts will be a waste since this inferred information will not be related to the leader himself, but to the cluster members. For privacy metrics related to the leader, a metric that depends on a location entropy privacy metric is proposed (this will be discussed in the next section), but the main question is as follows: What are the criteria of the leader election?

From the statement that could be inferred from the scenario included in Figure 3, which states "the connection to the LBS server still stands in case no query answer is found in the cache", a certain number of connections to the LBS server is imagined, and that is related to each LBS user in the past.

Let S represent the set of connections between the LBS users and server previously (as shown in Figure 1).

$$S = \left\{C_{past}(U_i).C_{past}(U_{i+1}).C_{past}\left(U_g\right)\right\} \quad (1)$$

where $C_{past}(U_i)$ is the number of connections to the untrusted LBS server related to the $user_i$, $C_{past}(U_{i+1})$ is the number of connections to the untrusted LBS server related to $user_{i+1}$, and $C_{past}(U_g)$ is the number of connections to the untrusted LBS server related to the $user_g$.

The criteria of the leader election will be based on the maximum number of connections to the LBS server. Therefore, the leader will be the user that satisfies the following condition:

$$Leader\_cret = max(S) \quad (2)$$

The reason behind this is that the probability of an attack launch against him (i.e., the user who has the maximum number of connections to the LBS server) will have the highest value compared to the other cluster members; the amount of information collected about him will be the greatest on the attacker's side. In addition, selecting the user that has the minimum number of LBS connections (to be the Leader) leads to negative impact on the user that has the maximum number of LBS connections. This case will put the user that has the maximum number of LBS server connections in a dangerous situation, especially when it comes to talking about the frequent coercion of the LBS server connection (i.e., in the case where no query answer is found in the cache). That is because the number of connections to the LBS server (attacker) increases, which in turn allows the attacker to collect more sensitive data about him. In other words, electing the LBS user that has the minimum number of connections to the LBS server does not serve the members of the cluster to protect privacy against the malicious LBS server. In light of this discussion, the need of the LBS user, which has the maximum number of LBS server connections, to be the leader will exactly match the need for full prevention of an LBS server connection as it relates to the other cluster members. Motivated by this mutual benefit underlying the symbiotic relationship, a robust TTP (the leader) is proposed. It is worth mentioning that the TTP approach is an optimum answer for the question related to the assumption of existing TTPs in all previous works.

Another important question that arises is related to the trust of this elected leader. The question is as follows: "Why must we trust the leader and not trust the LBS server at the same time?" The answer to this question is provided in the next sub section.

### *3.2. Trusting in The Leader*

This sub section discusses the trust issues related to the elected Leader. It provides additional criteria that depend on the reputation of the elected Leader, taking into consideration the impact of the previous condition expressed by (2).

To make this work more distinctive with respect to previous works, an answer to the previous question above is provided to scale up the cluster members' trust level in their elected leader. In short, the problem can be described by supposing that there is a probability of converting the leader himself

to an attacker. Since there is no difference between the elected leader and the LBS server as TTP, any LBS user contained in each cluster can expect to be tracked and attacked by the leader.

To manipulate this problem, we based things on the scenario proposed in [38] since their idea could be locally applied to the cluster members. Each LBS user located within each cell can store information about some of his/her visited POIs in their mobile devices' cache and can send this information as a kind of a helping hand to the neighbors to avoid dealing with untrusted third parties. The key idea is to construct a trust level for each cluster member as follows: Each LBS user will deliberately send a query (that previously has its answer in his/her mobile device cache) called a test query to all the cluster members. Based on the received answer, the trust level related to each user, except the sender, will be decreased or increased.

In general, let $TL_i(\dddot{value})$ represent the trust level value of a $user_i$, where it is located at $user_{i+1}$. If the received answer is true, then $TL_i(\dddot{value}) = TL_i(\dddot{value} + 1)$. If the received answer is fake, then $TL_i(\dddot{value}) = TL_i(\dddot{value} - 1)$. The resultant new value is called ***local reputation*** $(user\_L\_rep)$ related to $user_i$ that $user_{i+1}$ constructed about him/her. By each cluster member applying this process on the remainders, the local reputations can be obtained. In other words, each user included in the cluster will have pairs of sets of size $(g-1)$ about local reputations related to each of the remainders, and these local reputations could be increased or decreased depending on the credibility of the test query answer. As a result, we obtain the following pairs of sets:

$$\begin{cases} user_i\_L\_rep \\ user_{i+1}\_L\_rep \\ user_g\_L\_rep \end{cases} = \begin{cases} \left[\langle L\_rep(user_{i+1})\rangle.\langle L\_rep\left(user_g\right)\rangle\right] \\ \left[\langle L\_rep(user_i)\rangle.\langle L\_rep\left(user_g\right)\rangle\right] \\ [\langle L\_rep(user_i)\rangle.\langle L\_rep(user_{i+1})\rangle] \end{cases} \tag{3}$$

where $user_i\_L\_rep$ is the local reputation constructed about $user_i$ by both $user_{i+1}$ and $user_g$, $user_{i+1}\_L\_rep$ is the local reputation constructed about $user_{i+1}$ by both $user_i$ and $user_g$, and $user_g\_L\_rep$ is the local reputation constructed about $user_g$ by both $user_i$ and $user_{i+1}$.

Relying on the sum of the local reputation values constructed by other cluster members, the ***reputation*** of $user_i$ is obtained. So, $user_i\_rep(\widetilde{value})$, $user_{i+1}\_rep(\widetilde{value})$, and $user_g\_rep(\widetilde{value})$ are calculated as it is illustrated in Figure 5 below. Note that this process will be repeated over all the clusters (or cells) involved in our system model.

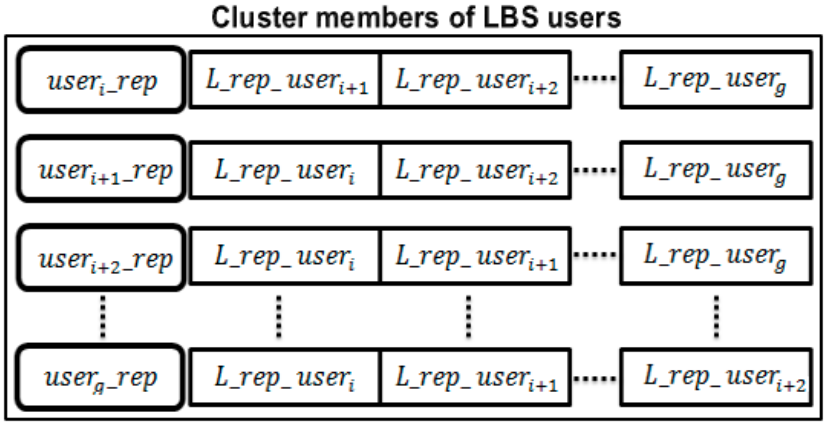


**Figure 5.** Local reputations of cluster members.

Based on the representation in formula (1), each $user\_rep(\widetilde{value})$ will match a certain number of LBS server connections in the past related to each cluster member as follows:

$$\begin{cases} [user_i] \\ [user_{i+1}] \\ \left[user_g\right] \end{cases} = \begin{cases} [\langle user_i\_rep(\widetilde{value})\rangle.\langle C_{past}(U_i)\rangle] \\ [\langle user_{i+1}\_rep(\widetilde{value})\rangle.\langle C_{past}(U_{i+1})\rangle] \\ [\langle user_g\_rep(\widetilde{value})\rangle.\langle C_{past}(U_g)\rangle] \end{cases} \tag{4}$$

By multiplying the two components related to each user in (4), the general reputation *G_rep* of each user is calculated as follows:

$$\begin{Bmatrix} G\_rep\_user_i \\ G\_rep\_user_{i+1} \\ G\_rep\_user_g \end{Bmatrix} = \begin{Bmatrix} [\langle user_i\_rep(\widetilde{value}) \times C_{past}(U_i)\rangle] \\ [\langle user_{i+1}\_rep(\widetilde{value}) \times C_{past}(U_{i+1})\rangle] \\ [\langle user_g\_rep(\widetilde{value}) \times C_{past}(U_g)\rangle] \end{Bmatrix} \quad (5)$$

As a result, the new criteria of Leader election will be:

$$leader\_rep = argmax\begin{Bmatrix} G\_rep\_user_i \\ G\_rep\_user_{i+1} \\ G\_rep\_user_g \end{Bmatrix} .0 \quad (6)$$

It is worth mentioning that a special case may occur when the maximum general reputation is the same for two users or more. In this case, the leader is elected randomly based on the same criteria. After electing a leader under the new criteria, all cluster members will be trusted for receiving the true answers to their queries sent to the LBS server by their own leader. The corresponding pseudo code for electing the leader is included in Algorithm 1.

**Algorithm 1:** Leader Election Algorithm

**Input:** $n \times n$ (number of cells or clusters), $g$ (number of LBS users in a cell or cluster), $C_{past\_u}$ (number of connections to the LBS server in the past for user $u$), $HashTable(key = user, val = G_{rep})$.
**Output:** $GLeader_C$ (general reputation of the leader in cell $c$)

```
for c = 1 to n×n do
    for u = 1 to g do
        i ← 1
        sum_rep ← 0
        while (i ≤ g) & &(u ≠ i)
            sum_rep += local_rep(u,i)
        end while
        G_rep = sum_rep × C_past_u
        HashTable[u] = G_rep
    end for
    GLeader_C = key(max(HashTable[val]))
    return GLeader_C
end for
```

Algorithm 2 shows the pseudo code for calculating the local reputation.

**Algorithm 2:** Calculating the Local Reputation ($local_{rep}$)

**Function** ***$local_{rep}$*** **(*revciever u.sender i*)**
**Input:** $TQ_S, ATQ_S$
**Output:** $local_{rep}$
1: $Answers_{reciever\_u}$ = $Test_i(TQ_S,u)$
2: $(NRA)$ = Number of Matching $(Answers_{reciever\_u}, ATQ_S)$
3: $(NWA)$ = Number $(TQ_S) - NRA$
4: new $(TL_{reciever\_u})$ = old $(TL_{reciever\_u}) + NRA - NWA$
5: $local_{rep}$ = new $(TL_{reciever\_u})$
6: **return** $local_{rep}$

where $TQ_S$ is the test queries, $ATQ_S$ is the answers to the test queries, $(NRA)$ is the number of right answers, $(NWA)$ is the number of wrong answers, and $TL$ is the trust level of the receiver.

Although electing the leader depends on his general reputation, the probability of converting into an attacker still stands. Specifically, there are no guarantees that cluster members will not be tracked or that queries analyzed and then attacked by the elected leader himself. On the one hand, this probability will be minimized since both the leader and the cluster members are considered moving objects. So, they can leave their cluster and pass on to other clusters or cells. On the other hand, if we ***reset the global reputation*** of the leader in cases where he leaves his cluster, we can ensure that he will not be a leader anymore. Figure 6 summarizes what is discussed in this paragraph.

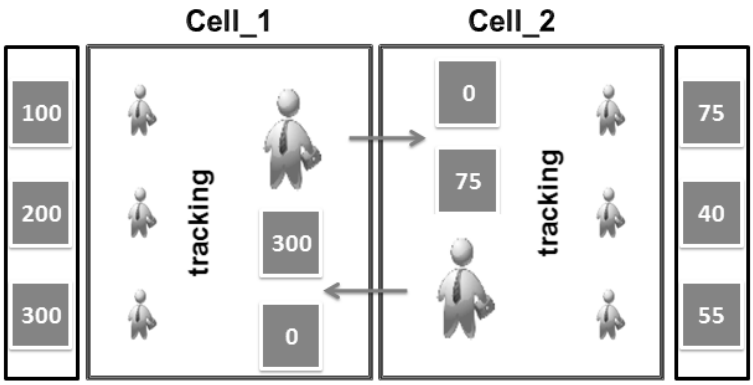


**Figure 6.** Resetting global reputation of moving Leader.

According to Figure 6, two leaders are elected in two clusters depending on the maximum general reputation among the cluster members, and this global reputation is reset for both leaders after passing their cells. Then, two new leaders will be elected based on the same criteria, and any cluster member that leaves his cluster will act under the control of the new elected leader. As a result, the cluster members' concern (about their leader acting as an attacker) is minimized. Thus, our second and final steps are achieved in the proposed model.

## 4. Used Privacy Metrics

In general, the final and actual attack (i.e., mugging, stealing, threat, or blackmail) that an attacker triggers against his victim will occur after a complete profile is obtained that is full of malicious content and holds sensitive personal information. This malicious content is gathered over time through sub inferences attacks. Any useful information that can help the attacker to determine the suitable moment of his actual attack will be added to the previous malicious content.

### *4.1. Inferences Attacks*

In inferences attacks, the attacker depends on his intuition utilization to gain personal information about his victim. Some of the most advanced inference attacks are briefly explored in this sub section.

**A Homogeneity attack** [42] means that if the users are located in a place that represents a landmark (they hide their real positions through the circumference of this landmark) such as a hospital, the attacker can infer that those users have problems related to their health without needing to accurately identify their positions, as shown in Figure 7.

**A query sampling attack** [15,43,44] is where the attacker employs the unfair location distribution of the LBS users for his own malicious purpose. This type of inference attack targets isolated users in a sparse region, as illustrated in Figure 8.

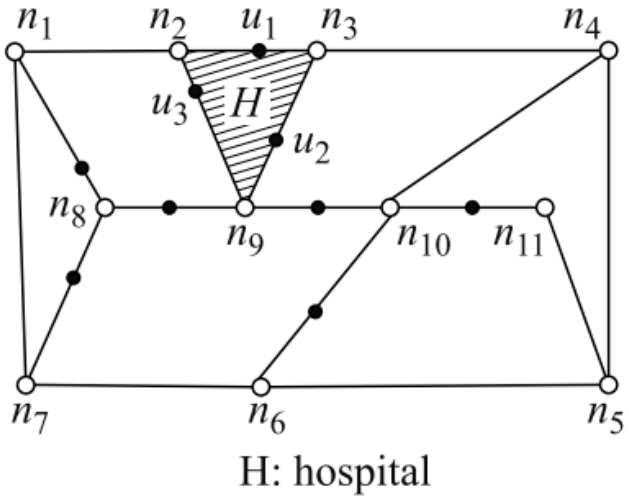


**Figure 7.** Homogeneity attack [42].

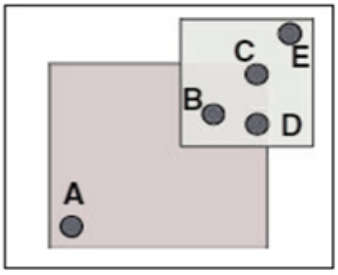


**Figure 8.** Query sampling attack [15].

**A semantic location attack** [45] is where the attacker can infer semantic meanings related to the behavior of the user by exploiting the amount of time the user stays in one place, such as a laboratory, bank, or university.

In each moment that an attacker applies one inference attack, he will have little success. These small successful attempts are related to many different sub inference attacks at various moments. Adding these small successful attempts enables the attacker to reach a suitable moment at which to launch an actual attack. Once this occurs, the goal of any privacy protection approach is to insert contradictory information into the profile, as mentioned previously regarding the state of the user who is concerned about his privacy. This confuses the attacker and obstructs his ability to determine his actual attack, as illustrated in Figure 9.

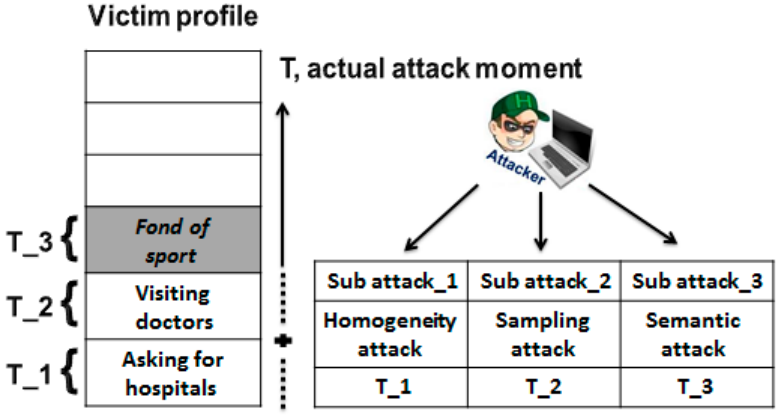


**Figure 9.** The profile of the LBS user specialized on the attacker side.

From Figure 9, it is obvious that on the attacker side there a clear conflict regarding whether this LBS user has a health problem and must participate in sports.

### 4.2. Types of Used Privacy Metrics

Many privacy metrics were examined in the survey provided in [14], where these privacy metrics are presented to assess how much the LBS user privacy has been broken for both location and query privacy. As the target is to achieve both location and query privacy, location entropy is selected since this metric could be used for both aspects. For location privacy, location entropy measures uncertainty in identifying the real position of a query issuer by quantifying the information obtained from the attacker side from location updates related to the trajectories of LBS users' motion. For query privacy, location entropy measures the unobservability when an LBS user visits a POI. This work focused on protecting the privacy of the leader (both location and query privacy) since the leader is considered the only LBS user that connects to the LBS server (a malicious party).

According to the proposed scenario in Figure 4, the queries involved in the system could be classified into two major groups. The first one includes queries that are answered by the cache, and the second one includes queries that are sent and answered by the LBS server through the leader. According to these two groups, two privacy metrics are needed in the proposed model.

#### 4.2.1. Leader Privacy Metric

Since real queries sent to the Leader that act as dummies on his side, the concept of $k-anonymity$ is automatically achieved to protect Leader privacy. Let $k$ denote to the $k-anonymity$ level (i.e., number of dummies or real queries that reach the Leader and are sent to the LBS server at $\tau$ moment). Let $p_i(i = 1, 2, \ldots, k)$ denote the probability of recognizing the $i^{th}$ location as a real location among $(k-1)$ dummies, and let $q_i$ denote the query probability of $i^{th}$ location as follows:

$$p_i = \frac{q_i}{\sum_{j=1}^{k} q_i} \tag{7}$$

Thus, location entropy at $\tau$ moment could be presented as follows:

$$E(\tau) = -\sum_{i=1}^{k} P_i \times log_2 \times P_i \tag{8}$$

When all the $k$ possible locations have the same query probability, $E(\tau)$ achieves a maximum value. In this case, the location entropy will be:

$$E(\tau)_{max} = log_2(k) \tag{9}$$

Consider that $E(\tau)_{\max} = b$. For a given $E(\tau)$ value (equal to 2, for example) where $E(\tau) < b$, this $E(\tau)$ value could be read from both viewpoints (i.e., the Leader side and attacker side). On the Leader side, he will state, for instance, "I have achieved privacy protection for my real position to be revealed by an attacker with $(2)$ value and as high as $E(\tau)$ value for high privacy protection". For the attacker side, it will be stated "I identify my victim's real position with $(b-2)$ value and as low $E(\tau)$ value as I became able to accurately identify my victim's real position, and thus I become closer to the moment of my attack launch". In general, Figure 10 illustrates our new proposed privacy metric at a certain moment of time.

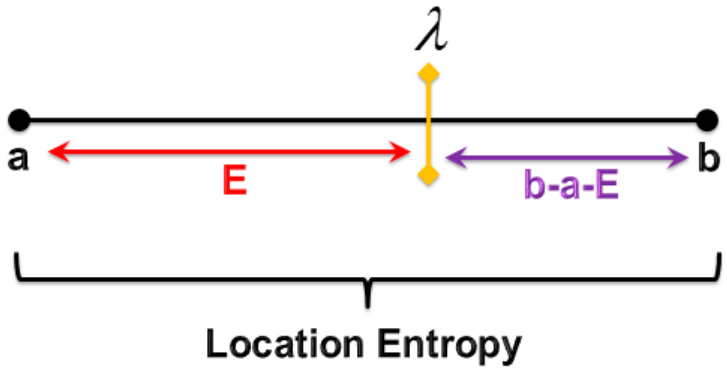


**Figure 10.** Our proposed privacy metric.

Let $\Gamma = (\tau_1, \tau_2, \tau_3, \ldots, \tau_n)$ refer to the moments at which the Leader connects to the LBS server where some of the inference attacks are applied. Thus, $\lambda$ represents the closeness of the attacker to the moment of his attack launch.

According to Figure 10, it is clear that the location entropy value varies, ranging from $a$ to $b$, $E$ represents the amount of privacy protection on the leader side, and $(b-a-E)$ represents the closeness of the attacker to his attack launch against the leader. As a result, the new privacy metric can be given as:

$$\lambda = \sum_{\tau \in (\Gamma)} (b - a - E(\tau)) \tag{10}$$

$$= \sum\nolimits_{\tau \in (\Gamma)} (log_2)k) - a - E(\tau)) \tag{11}$$

where $\tau \in \Gamma$.

Note that the proposed privacy metric, which is specialized for the Leader, could be considered a standard one; this privacy metric could be applied to any approach that belongs to any class provided in the literature review. The reason behind this is that any user included in any LBS system could be considered a Leader for himself.

#### 4.2.2. System Privacy Metric

In general, users that are finding their query answers in the cache will achieve a full privacy value under any privacy metric, as they won full prevention from dealing with untrusted LBS servers through the Leader, and no information could be inferred about both real positions and real queries. We used the privacy metric proposed in [40] called the cache hit ratio (CHR), which measures the queries answered by the cache as a proportion of the total number of queries involved in the system as follows:

$$CHR = \frac{\left|Q_{answered\_cache}\right|}{\left|Q_{answered\_server}\right| + \left|Q_{answered\_cache}\right|} \tag{12}$$

## 5. Experimental Results and Evaluation

In this paper, Matlab software is used to implement the proposed approach. The simulation inputs are assumed to be that the targeted area is divided into a $(160 \times 160)$ cell and the number of users included in the system equals (10,000). The cache is represented through a data base consisting of one table only, where the information about POIs, included in the cells, is stored through the queries that are answered by the LBS server. The information stored in the cache mainly included the type of POI and the position of the cell that is located within. A timestamp is attached for both stored information in the cache and the queries so that these timestamps will be used, through a simple comparison, to achieve the data freshness term. In addition, timestamps are also attached to the LBS users since they are considered to be moving objects. For query probability, it is generated randomly

with help provided by Google Maps API. The POIs are considered static, and we did not deal with moving queries.

Three previous approaches are selected for the comparison with the proposed approach; they include enhanced-DLS [34], hiding in a mobile crowd [38], and enhanced-CaDSA [40].

### *5.1. Communication Cost Results Evaluation*

Based on the communication costs (number of queries sent to the LBS server), the proposed approach is evaluated in two respects, which are the impact of time as it progresses and the impact of the k – anonymity value. The bloom filter is used for searching the answers of the queries in the cache since it effectively minimizes the search time. By doing so, the system response will be enhanced, and the gap will be filled since the Leader must waste some time receiving real queries from some of his cluster members to protect his privacy.

Figure 11 shows a snapshot taken at a time progress of 120 min. It can be shown that the enhanced-DLS provides the worst performance among the other approaches; it does not use query response caching. Thus, all queries related to users are sent to the LBS server. In other approaches, the number of queries sent to the LBS server is decreased since many queries find their answers in the cache. Mobile Crowd approach achieved better performance compared to enhanced-DLS, but its performance was worse than that of the enhanced-CaDSA, while enhanced-CaDSA selects dummy locations that can hit more contributions in the cache based on both normalized distance and data freshness terms, and mobile crowd took none of them into consideration. The proposed approach overtakes it with respect to the time progress term; as the proposed approach does not need to generate dummies since it exploits real ones as dummies, and enhanced-CaDSA needs to create dummies for each query forced to connect to the LBS server.

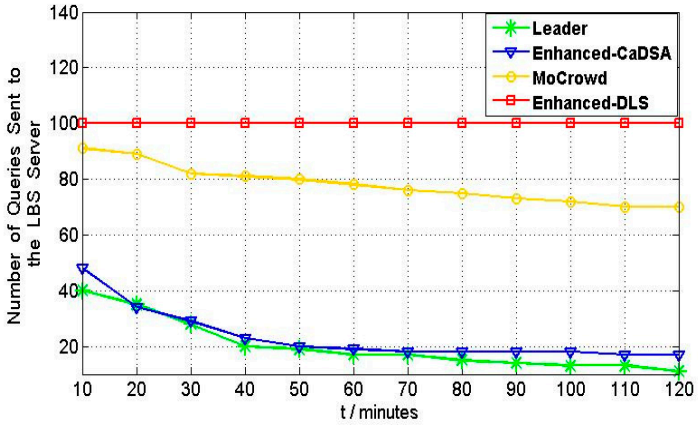


**Figure 11.** Communication cost VS. Time progress.

Figure 12 supports the results obtained in the first aspect (i.e., the time progress), where the number of queries sent to the LBS server in enhanced-DLS increases linearly as $k$ increases, and again enhanced-DLS gives the worst results among the other approaches. Enhanced-CaDSA performs better than both enhanced-DLS and Mobile Crowd due to its good cache design. Compared to enhanced-CaDSA, the proposed approach gives better results. Each real query, which acts as a "dummy" in the proposed approach, maps several generated dummies in enhanced-CaDSA. Consequently, the number of queries sent to the LBS server, to achieve the k – anonymity concept, will obviously be less.

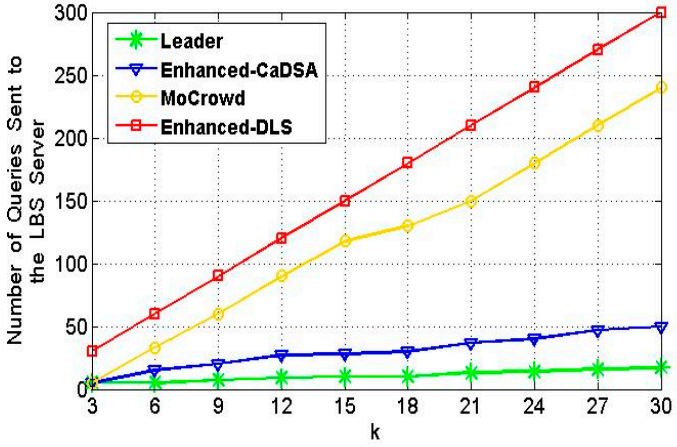


**Figure 12.** Communication cost VS. Anonymity level.

### 5.2. Resistance Against Inferences Attacks Results Evaluation

Achieving a higher $k-anonymity$ level is preferred since it represents a higher privacy protection level. However, this $k-anonymity$ level is represented by the number of generated dummies attached with the original query; this $k-anonymity$ level is tightly coupled with the quality of the generated dummies (i.e., generating strong dummies). So, even if the Leader approach achieved the minimal $k-anonymity$ level compared to the remainders, it actually achieved the best privacy protection level under the dummy generation term. To make this idea clearer, the impact of applying a mixture of inferences attacks is discussed, taking into consideration the application of the same $k-anonymity$ level at each approach included in our comparison.

Because the proposed new privacy metric $\lambda$ relates to the Leaders involved in the system, we evaluated the closeness of the attacker (LBS server or his maintainer) to the moment of his attack launch against the Leaders in the time progress. In addition, because the k-anonymity concept is achieved automatically in the proposed approach, $k$ is set to 6 for each cluster (i.e., at any moment, the Leader will receive five real queries as dummies in addition to the real query related to the Leader himself to be sent to the LBS server). Under threat of a mixture of inferences attacks (i.e., heterogeneous attack, query sampling attack, and location semantic attack), a snapshot at $(t = 120)$ is taken. Twenty leaders' situations are evaluated taking into account a threshold that equals $(thr = 0.8)$, at which the Leader is considered to be in dangerous conditions of attack by the LBS server. In order to make an identical comparison among the approaches, the same number of LBS users (i.e., 20) are randomly selected from each of the three previous approaches to be evaluated under the same threshold condition. It should be mentioned that the threat model provided in [46] is used as the basis of our approach, where every 3 min a different kind of inference attack is periodically applied. Figure 13 shows the results. It is taken into account that each LBS user involved in enhanced-DLS, Mobile Crowd, and enhanced-CaDSA is considered a leader. The comparison of the dangerous status of leaders is summarized in Table 1.

**Table 1.** Comparison of dangerous status of leaders.

| **Settings:** $t = 120$, $k = 6$, $thr = 0.8$ | | |
|---|---|---|
| **Approach** \ **Term** | **Number of Leaders Exceeded the Threshold** | **Percentage of Encroachment** |
| Leader | 2 | 0.1 |
| Enhanced-CaDSA | 12 | 0.6 |
| Mobile Crowd | 15 | 0.75 |
| Enhanced-DLS | 20 | 1 |

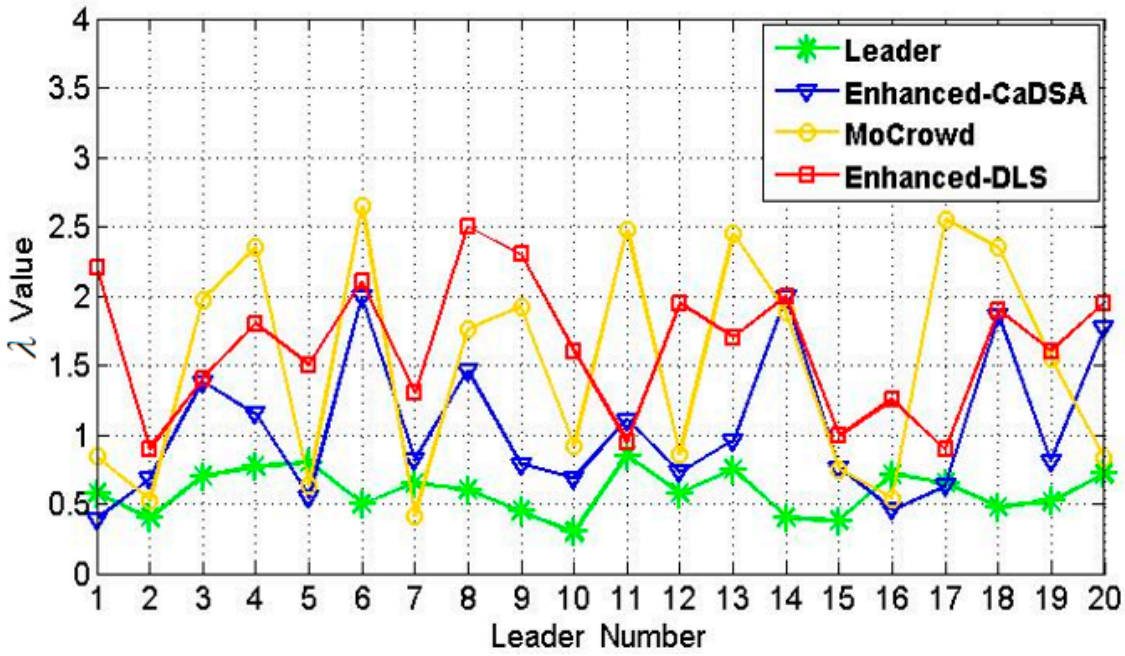


**Figure 13.** $\lambda$ values for 20 Leaders, k = 6.

Table 1 shows that the proposed approach has the minimum number of LBS users that reached a dangerous state. Three-quarters of Leaders and more than half of Leaders exceeded the threshold in Mobile Crowd and Enhanced-CaDSA, respectively. For Enhanced-DLS, all leaders exceeded the threshold since they are forced to connect to the untrusted LBS server and become vulnerable to direct threats all the time. On the one hand, because of the global reputation of the leaders is reset in the proposed approach, and given the endless continuity of this mission (i.e., the leader mission) in the other approaches, we gained the minimum number of LBS users that exceeded the defined threshold. This in turn means that the leader approach has the highest resistance against the used inferences attacks. This robustness could be justified through hiding the cluster members behind their leader. In other words, LBS users are in complete silence in the eyes of the attacker under any inference attack. Thus, the LBS user that is located in an isolated place (query sampling attack) or those that are resided in a one POI (homogeneity attack) for a long time (semantic location attack) will be in complete safety since they send their queries to a leader that is located in a different POI.

On the other hand and based on the principle that states "prevention is better than the cure," LBS users that have reached a dangerous state can be altered to give up their missions as leaders and thus keep the attacker away from the actual moment of his attack launch. Compared to Enhanced-DLS, Mobile Crowd, and Enhanced-CaDSA, this capability is not offered.

Table 2 supports the results collected in Table 1, where the threshold was redefined in different values, the simulation was re-executed at different snapshots, different leaders were randomly selected, and the percentage of the leaders that exceeded the thresholds was calculated.

**Table 2.** Percentage of encroachment of the predefined thresholds.

| Try NO | NO of Leaders | *t* | *thr* | Percentage of Encroachment | | | |
|---|---|---|---|---|---|---|---|
| | | | | Leader | Enhanced-CaDSA | Mobile Crowd | Enhanced-DLS |
| 1 | 40 | 130 | 0.75 | 0.13 | 0.53 | 0.67 | 1 |
| 2 | 60 | 140 | 0.7 | 0.15 | 0.55 | 0.83 | 1 |
| 3 | 80 | 150 | 0.65 | 0.22 | 0.42 | 0.71 | 1 |
| 4 | 100 | 160 | 0.6 | 0.17 | 0.45 | 0.59 | 1 |
| 5 | 120 | 170 | 0.55 | 0.14 | 0.4 | 0.57 | 1 |

### 5.3. Cache Hit Ratio Results Evaluation

As is shown in Figure 14 below, the enhanced-DLS achieved zero values since it does not use a cache. For the other approaches, the cache hit ratio is enhanced during the time progress because of the cache. Enhanced-CaDSA presents better cache hit ratio values compared to the Mobile Crowd. This is because the dummy locations, generated to protect the privacy of the LBS user, use normalized distance, which in turn optimizes the quality of information stored in the cache. Moreover, the data freshness term will keep the most important information that is expected to be used to answer future queries. Furthermore, the cache contained in mobile devices of users cannot be compared to the storage of access points, which represents caches in Enhanced-CaDSA. Despite the good performance of Enhanced-CaDSA, the proposed approach provides better cache hit ratio values, where the Leader approach used a hundred POIs ($p = 100$) in the system compared to one POI ($p = 1$) used in the enhanced-CaDSA. The reason behind this is that the proposed approach depends on the real or actual positions to generate dummies sent to the LBS server. This means the leader approach uses the precise locations of users and exploits their actual positions, which are likely located to search for POIs. In everyday life, compared to the selection of dummies using normalized distances, exploiting actual positions as dummies has more of an impact on the quality of information stored in the cache. The answers to the dummy queries lead to a higher probability of existing future query answers in the cache.

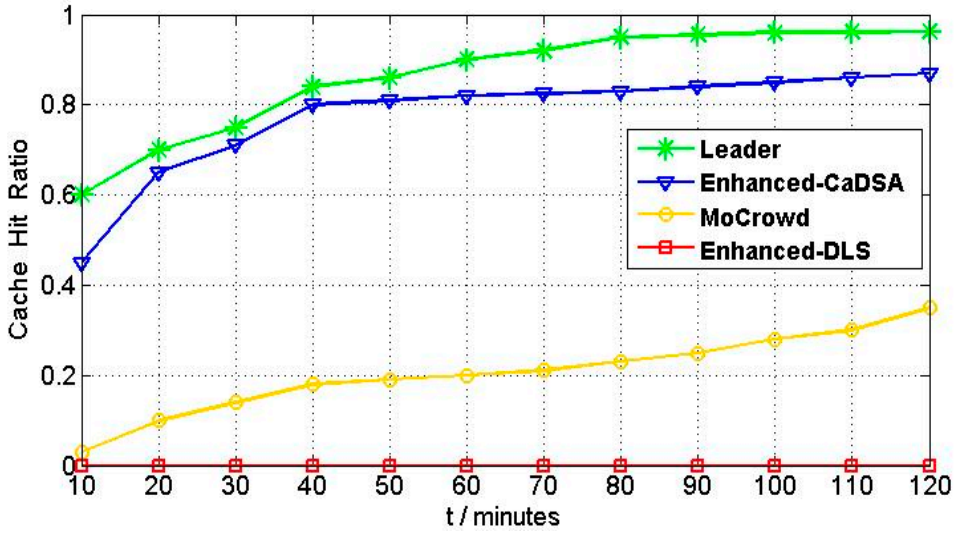


**Figure 14.** Cache hit ratio vs. time progress, p=100.

## 6. Conclusions

In this technological age, privacy is one of the major concerns of mobile device users. When it comes to achieving complete privacy protection for users of location-based services, the symbiotic relationship-based leader approach is proposed. Among the group of LBS users (a cluster), this Leader is elected based on a global reputation. This global reputation is valued through two aspects, which are (1) the number of connections (done by the LBS user in the past) with the LBS server and (2) the local reputations that the other cluster members created for the leader. Under the assumption that the leader himself acts as an attacker and to prevent this leader from being a leader again, her/his global reputation is deliberately reset when moving from one cluster to another, scaling up the cluster members' trust level in their elected leader. Compared to previous approaches, the leader approach provided better performance in terms of communication cost and cache hit ratio. Moreover, according to the new privacy metric (attacker's closeness to the moment of his/her actual attack) and under a threat mixture of advanced inferences attacks (homogeneity attack, query sampling attack, and

semantic location attack), the leader approach has the highest robustness against the previous attacks, which guarantees a high level of privacy protection.

In future work, protecting the privacy of the queries sent to the cache or those exchanged among LBS users will be taken into consideration. In addition, optimizing the availability and reliability quality attributes of the system will be manipulated by fixing the disconnecting problem that could have occurred and that is related to the leader. Moreover, using cache refreshing will be taken into account to keep only interesting responses.

**Author Contributions:** Conceptualization, Hosam Alrahhal and Mohamad Shady Alrahhal; Methodology, Hosam Alrahhal and Mohamad Shady Alrahhal; Software, Mohamad Shady Alrahhal; Razan Jamous; Formal Analysis, Kamal Jambi; Writing-Review & Editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chen, L.; Thombre, S.; Jarvinen, K.; Simona, L.E.; Alén-Savikko, A.; Leppakoski, H.; Bhuiyan, M.Z.H.; Bu-Pasha, S.; Ferrara, G.N.; Honkala, S.; et al. Robustness, Security and Privacy in Location-Based Services for Future IoT: A Survey. *IEEE Access* **2017**, *5*, 8956–8977. [CrossRef]
2. Elmisery, A.M.; Rho, S.; Botvich, D. A Fog Based Middleware for Automated Compliance With OECD Privacy Principles in Internet of Healthcare Things. *IEEE Access* **2016**, *4*, 8418–8441. [CrossRef]
3. Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A.V. Security and privacy for cloud-based IoT: Challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–33.
4. Sun, G.; Chang, V.; Ramachandran, M.; Sun, Z.; Li, G.; Yu, H.; Liao, D. Efficient location privacy algorithm for Internet of Things (IoT) services and applications. *J. Netw. Comput. Appl.* **2017**, *89*, 3–13. [CrossRef]
5. Ullah, I.; Shah, M.A. A novel model for preserving Location Privacy in Internet of Things. In Proceedings of the 2016 22nd International Conference on Automation and Computing (ICAC), Colchester, UK, 7–8 September 2016; IEEE: Piscataway, NJ, USA, 2016.
6. Abdelmoty, A.; Alrayes, F. Towards understanding location privacy awareness on geo-social networks. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 109. [CrossRef]
7. Pagallo, U.; Durante, M.; Monteleone, S. What Is New with the Internet of Things in Privacy and Data Protection? Four Legal Challenges on Sharing and Control in IoT. In *Data Protection and Privacy: (In) Visibilities and Infrastructures*; Springer International Publishing: Cham, Switzerland, 2017; pp. 59–78.
8. Hasan, A.S.M.; Qu, Q.; Li, C.; Chen, L.; Jiang, Q. An effective privacy architecture to preserve user trajectories in reward-based LBS applications. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 53. [CrossRef]
9. Alrawais, A.; Alhothaily, A.; Hu, C.; Cheng, X. Fog Computing for the Internet of Things: Security and Privacy Issues. *IEEE Internet Comput.* **2017**, *21*, 34–42. [CrossRef]
10. Ma, Y.; Wang, Y.; Yang, J.; Miao, Y.; Li, W. Big Health Application System based on Health Internet of Things and Big Data. *IEEE Access* **2017**, *5*, 7885–7897. [CrossRef]
11. Samarah, S.; Zamil, M.G.A.; AlEroud, A.F.; Rawashdeh, M.; Alhamid, M.F.; Alamri, A. An Efficient Activity Recognition Framework: Toward Privacy-Sensitive Health Data Sensing. *IEEE Access* **2017**, *5*, 3848–3859. [CrossRef]
12. Dardari, D.; Closas, P.; Djuric, P.M. Indoor tracking: Theory, methods, and technologies. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1263–1278. [CrossRef]
13. Zhang, L.; Liu, K.; Jiang, Y.; Li, X.-Y.; Liu, Y.; Yang, P.; Li, Z.; Yang, P. Montage: Combine frames with movement continuity for realtime multi-user tracking. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1019–1031. [CrossRef]
14. Shin, K.G.; Ju, X.; Chen, Z.; Hu, X. Privacy protection for users of location-based services. *IEEE Wirel. Commun.* **2012**, *19*, 30–39. [CrossRef]
15. Wernke, M.; Skvortsov, P.; Dürr, F.; Rothermel, K. A classification of location privacy attacks and approaches. *Pers. Ubiquitous Comput.* **2014**, *18*, 163–175. [CrossRef]

16. Feng, W.; Yan, Z.; Xie, H. Anonymous Authentication on Trust in Pervasive Social Networking Based on Group Signature. *IEEE Access* **2017**, *5*, 6236–6246. [CrossRef]
17. Yu, R.; Bai, Z.; Yang, L.; Wang, P.; Move, O.A.; Liu, Y. A Location Cloaking Algorithm Based on Combinatorial Optimization for Location-Based Services in 5G Networks. *IEEE Access* **2016**, *4*, 6515–6527. [CrossRef]
18. Gedik, B.; Liu, L. Protecting Location Privacy With Personalized k-Anonymity: Architecture and Algorithms. *IEEE Trans. Mob. Comput.* **2008**, *7*, 1–18. [CrossRef]
19. Gruteser, M.; Grunwald, D. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *MobiSys '03: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*; ACM: New York, NY, USA, 2003.
20. Mokbel, M.F.; Chow, C.-Y.; Aref, W.G. The New Casper: Query Processing for Location Services Without Compromising Privacy. In Proceedings of the VLDB '06, Seoul, Korea, 12–15 September 2006; ACM: New York, NY, USA, 2006; pp. 763–774.
21. Beresford, A.; Stajano, F. Location Privacy in Pervasive Computing. *IEEE Pervasive Comput.* **2003**, *2*, 46–55. [CrossRef]
22. Hoh, B.; Gruteser, M. Protecting location privacy through path confusion. In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), Athens, Greece, 5–9 September 2005; pp. 194–205.
23. Meyerowitz, J.; Roy Choudhury, R. Hiding stars with fireworks: Location privacy through camouflage. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, Beijing, China, 20–25 September 2009; pp. 345–356.
24. Xu, T.; Cai, Y. Feeling-Based Location Privacy Protection for Location-Based Services. In Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, IL, USA, 9–13 November 2019; ACM: New York, NY, USA, 2009; pp. 348–357.
25. Pingley, A.; Yu, W.; Zhang, N.; Fu, X.; Zhao, W. Cap: A context-Aware Privacy Protection System for Location-Based Services. In Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems, Montreal, QC, Canada, 22–26 June 2009; pp. 49–57.
26. Hong, S.; Liu, C.; Ren, B.; Huang, Y.; Chen, J. Personal privacy protection framework based on hidden technology for smartphones. *IEEE Access* **2017**, *5*, 6515–6526. [CrossRef]
27. Manweiler, J.; Scudellari, R.; Cox, L.P. Smile: Encounter-Based Trust for Mobile Social Services. In Proceedings of the CCS '09, Chicago, IL, USA, 9–13 November 2009; ACM: New York, NY, USA, 2009; pp. 246–255.
28. Hu, H.; Xu, J. Non-Exposure Location Anonymity. In Proceedings of the 2009 IEEE 25th International Conference on Data Engineering, Shanghai, China, 29 March–2 April 2009; pp. 1120–1131.
29. Chen, Z. Energy-Efficient Information Collection and Dissemination in Wireless Sensor Networks. Ph.D. Thesis, University of Michigan, Ann Arbor, MI, USA, 2009.
30. Ardagna, C.; Cremonini, M.; Damiani, E.; De Capitani di Vimercati, S.; Samarati, P. Location privacy protection through obfuscation-based techniques. In Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA, 8–11 July 2007; pp. 47–60.
31. Gutscher, A. Coordinate transformation—A solution for the privacy problem of location based services? In Proceedings of the 20th International Conference on Parallel and Distributed Processing (IPDPS '06), Rhodes Island, Greece, 25–29 April 2006; p. 354.
32. Kido, H.; Yanagisawa, Y.; Satoh, T. An Anonymous Communication Technique Using Dummies for Location-based Services. In Proceedings of the ICPS '05. Proceedings. International Conference on Pervasive Services 2005, Santorini, Greece, 11–14 July 2005.
33. Pingley, A.; Zhang, N.; Fu, X.; Choi, H.-A.; Subramaniam, S.; Zhao, W. Protection of Query Privacy for Continuous Location Based Services. In Proceedings of the 2011 proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011.
34. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Achieving k-anonymity in privacy-aware location-based services. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014.
35. Hara, T.; Suzuki, A.; Iwata, M.; Arase, Y.; Xie, X. Dummy-Based User Location Anonymization Under Real-World Constraints. *IEEE Access* **2016**, *4*, 673–687. [CrossRef]
36. Sun, W.; Chen, C.; Zheng, B.; Chen, C.; Liu, P. An Air Index for Spatial Query Processing in Road Networks. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 382–395. [CrossRef]

37. Dewri, R.; Thurimella, R. Exploiting service similarity for privacy in location-based search queries. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 374–383. [CrossRef]
38. Shokri, R.; Theodorakopoulos, G.; Papadimitratos, P.; Kazemi, E.; Hubaux, J.-P. Hiding in the mobile crowd: Locationprivacy through collaboration. *IEEE Trans. Dependable Secur. Comput.* **2014**, *11*, 266–279.
39. Zhu, X.; Chi, H.; Niu, B.; Zhang, W.; Li, Z.; Li, H. Mobicache: When k-anonymity meets cache. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013.
40. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H.; Ben, N. Enhancing privacy through caching in location-based services. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015.
41. Georgiadou, Y.; de By, R.A.; Ourania, K. Location Privacy in the Wake of the GDPR. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 157. [CrossRef]
42. Pan, X.; Chen, W.; Wu, L.; Piao, C.; Hu, Z. Protecting personalized privacy against sensitivity homogeneity attacks over road networks in mobile services. *Front. Comput. Sci.* **2016**, *10*, 370–386. [CrossRef]
43. Lin, C.; Wu, G.; Yu, C.W. Protecting location privacy and query privacy: A combined clustering approach. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 3021–3043. [CrossRef]
44. Saravanan, S.; Ramakrishnan, B.S. Preserving privacy in the context of location based services through location hider in mobile-tourism. *Inf. Technol. Tour.* **2016**, *16*, 229–248. [CrossRef]
45. Li, Y.; Yuan, Y.; Wang, G.; Chen, L.; Li, J. Semantic-Aware Location Privacy Preservation on Road Networks. In *International Conference on Database Systems for Advanced Applications*; Springer International Publishing: Cham, Switzerland, 2016.
46. Lee, B.; Oh, J.; Yu, H.; Kim, J. Protecting location privacy using location semantics. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2011.