Ensuring Privacy Protection in Location-based Services through Integration of Cache and Dummies

Sara Alaradi[1], Nisreen Innab[2]
Department of Information Security
Naif Arab University for Security Sciences
Riyadh, Kingdom of Saudi Arabia

***Abstract*—Location-Based Services (LBS) have recently gained much attention from the research community due to the openness of wireless networks and the daily development of mobile devices. However, using LBS is not risk free. Location privacy protection is a major issue that concerns users. Since users utilize their real location to get the benefits of the LBS, this gives an attacker the chance to track their real location and collect sensitive and personal information about the user. If the attacker is the LBS server itself, privacy issues may reach dangerous levels because all information related to the user's activities are stored and accessible on the LBS server. In this paper, we propose a novel location privacy protection method called the Safe Cycle-Based Approach (SCBA). Specifically, the SCBA ensures location privacy by generating strong dummy locations that are far away from each other and belong to different sub-areas at the same time. This ensures robustness against advanced inference attacks such as location homogeneity attacks and semantic location attacks. To achieve location privacy protection, as well as high performance, we integrate the SCBA approach with a cache. The key performance enhancement is storing the responses of historical queries to answer future ones using a bloom filter-based search technique. Compared to well-known approaches, namely the ReDS, RaDS, and HMC approaches, experimental results showed that the proposed SCBA approach produces better outputs in terms of privacy protection level, robustness against inference attacks, communication cost, cache hit ratio, and response time.**

Keywords*—*Privacy protection; dummy; cache; safe cycle; location homogeneity attack; semantic location attack

I. Introduction

Recently, the world has witnessed the birth of what is called the Internet of Things (IoT) [1, 2, 3], in which scientists have moved towards smart cities and smart systems that are supported by smart Location-Based Services (LBS) [4, 5]. Smart LBS are considered one of the most important backbones of the IoT. However, similar to other research fields, the IoT research field has issues and challenges that should be answered. Privacy protection in smart LBS is one of the most important issues and challenges [6, 7].

To identify the problem, the following figure illustrates the general (or classical) scenario of smart LBS usage.

As shown in Fig. 1, the LBS user constructs a query based on his or her real location, and the query is processed at the LBS server site. The result will then be sent back to the LBS user.

Since the LBS server can store information related to the user's activities, it is easy to track the user's real location and extract personal and sensitive information about the user (such as interests, customs, health, religious and political relationships). This, in turn, means that the LBS server can act as a hacker (i.e., malicious party) to attack the privacy of the user.

In this research, we address the privacy protection of the LBS user by protecting the real location against the LBS server. The research questions are:

- How to ensure the privacy protection of the LBS user by protecting the real location [7, 8, 9]?
- Since the LBS server can apply inference attacks such as semantic location attacks [10,11,12] and homogeneity location attacks [13], how to ensure the robustness against these kinds of inference attacks?
- How to ensure the performance of the system by enhancing the response time of the query?

To guarantee the location privacy of the LBS users, we can surround the real location of the LBS user by some dummy locations, so that the server cannot recognize the real location among the dummies.

In general, the contribution of this paper is as follows:

- In responding to the first research question, we propose a novel dummy-based approach to protect the location privacy of LBS users. Depending on the query probability, our proposed approach selects (or generates) dummy locations that ensure the highest privacy protection level according to an entropy privacy metric.
- In responding to the second research question, in terms of generating strong dummy locations, the proposed approach creates defenses against both the location homogeneity attack and the semantic location attack based on a safe cycle.
- In responding to the third research question, the proposed approach integrates with the cache, which is represented by an access point, to enhance the overall system performance by serving future queries.

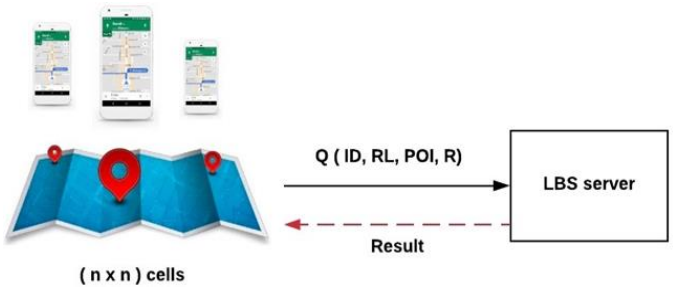


Fig. 1. Classical Scenario of Smart LBS Usage.

The rest of the paper is organized as follows: the related work is provided in Section II. In Section III, the proposed privacy protection system is presented in detail. The security analysis is discussed in Section IV. In Section V, the used metrics are defined, followed by the experimental results and evaluations in Section VI. Finally, the paper is concluded in Section VII.

II. Related Work

In general, there are two main categories of LBS privacy protection approaches: user-based approaches and server-based approaches, and each category has its own techniques, as shown in Fig. 2 [14].

A. Server-Based Approaches Category

Private Information Retrieval (PIR) protocol was proposed in [14] to retrieve POIs queried by the user. The strategy followed by the authors is that instead of determining their real position, users define an index through the provider. Depending on the processing of this index, the provider executes the PIR protocol to extract the corresponding POI with an encryption stage. Another PIR-based approach was developed in [15], in which a combination of the concept of ϵ-differential privacy and PIR is performed to ensure obtaining the same amount of information representing the query response. The key idea the authors used is to rely on the statistics of the queries to retrieve a similar heap of information for each query; thus, it could be employed to weaken the ability of the attacker who tries to obtain private information.

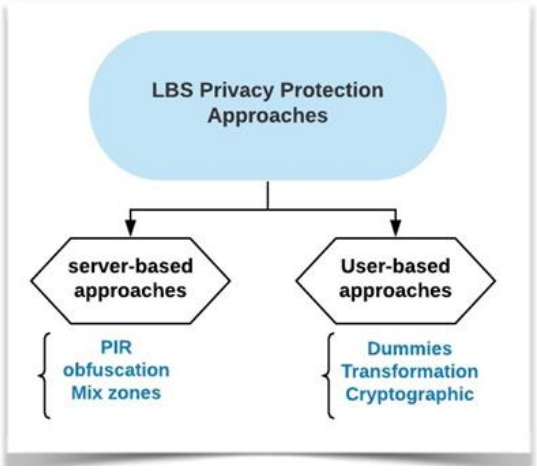


Fig. 2. Classification of LBS Privacy Protection Approaches.

Spatial obfuscation techniques protect privacy by minimizing the accuracy of the location information sent by the user to the server. A classical spatial obfuscation approach is provided in [16], in which a user sends a circular area instead of the accurate user position. Using the same idea, the work [17] presented a new approach. The difference was that instead of using geometric obfuscation shapes (i.e., circles), the authors used obfuscation graphs to apply the concept of position obfuscation to road networks. The obfuscation technique was developed in [18] to present a robustness against semantic location attacks, in which the location of the user cannot be mapped with a high probability to certain critical locations, such as a hospital. Therefore, a map-aware obfuscation approach was proposed, in which the key idea is expanding the obfuscation area adaptively in such a way that the probability of the user being in a certain semantic location is below a given threshold. The cloaking region is a protection method inspired by the obfuscation technique. The key idea is to cloak the real location of the user in spatial and temporal domains. To protect the privacy, the authors of the work [19] played on the resolution of the cloaking region through modifying the spatial-temporal dimensions, satisfying certain conditions to achieve a high k-anonymity level. Using the cloaking region method, the authors of [20] manipulated the problem of applying a constant level of privacy protection (i.e., k = constant to achieve the k-anonymity concept); however, this constant level may not be the user's preference and may not be needed. Thus, they allowed the user to express the privacy level he or she wishes so that the user can minimize the resolution of the cloaking region in the regions that the user feels relax and maximize it in other regions. A hierarchical grouping algorithm integrated with the cloaking region was proposed in [21]. To ensure the privacy protection of the users, the hierarchical grouping algorithm groups the users in different sets, and the cloaking region method is then applied to the orders of the users (i.e., their queries when asking for POIs). Finally, the hierarchical grouping algorithm collects the orders in each group, sending them together to the server. This confuses the attacker trying to determine the real locations of the users. In [22], the server acts as a location mask to camouflage the actual position of the user. The basic idea is to exploit the landmarks located in the area the user resides, hiding the real position of the user in a landmark such as a university or sports city. For the cases in which no landmark is available, the server creates an imaginative landmark based on the information stored previously about the successful tries. Similar to [22], [23] exploited the geographic context of the area where the user is located to build landmarks. The difference was that [23] dealt with moving objects, avoiding creating imaginative landmarks and considering that the motion of the objects can be exploited to find effective landmarks. In their work [24], Gedlik et al. presented a personalized K-anonymity approach, in which the server acts as an anonymizer. This approach adopts to conditions provided by the user (i.e., to protect the privacy), and a spatial-temporal mask is applied on the position of the user, providing the k-anonymity level of tolerance that the user wishes. Based on the same idea, [25] suggested personalization according to the user profile which contains the conditions of privacy protection.

One of the most popular techniques used in this group was proposed by Beresford et al. in [26] called mix zones. The users located in an area are grouped into many spatial regions. This region protects the real positions for the users by hiding them within such regions. These regions will then be mixed together, and no location updates inside a mixing zone occur during the motion of the objects. The work [27] improved the mix zones approach through the addition of a pseudonym concept. Therefore, another condition is satisfied, which is that the user must utilize another pseudonym when leaving one mix zone to another. Another development was performed on mix zones, in which the authors of [28] proposed the MobiMix approach. The essence of the development idea was to the make mix zones approach more robust against the attackers. To this end, the authors took into consideration various context information that can be exploited to derive detailed trajectories such as geometrical and temporal constraints.

B. User-based Approaches Category

In the work [29], Yanagisawa et al. 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 the dummies ensures privacy protection because the LBS server cannot recognize the real position among the dummies. Similarly, [30] used dummies to protect the privacy of the LBS users. It depends on selecting the dummy using a normalized distance to confuse the attacker and limit his/her ability to track or infer some sensitive information about the query issuer (i.e., the LBS user). Another approach using the dummies idea was presented in [31] called DUMMY-Q, but the idea is applied to the query itself rather than the location. Therefore, dummy queries of different attributes from the same location are generated to hide the real query. To make the generated dummies stronger, two aspects are taken into consideration: 1) The query context; and 2) the motion model. Hara et al. [32] developed a dummy-based approach, manipulating dummies' generation from our real life. Therefore, they considered the physical constraints of the real world. The feature that distinguishes this work was that the trajectories of the generated dummies cross the trajectories of the actual movement of the LBS user. The authors of work [37] proposed a dummy data array (DDA) algorithm for generating dummy locations to protect the location privacy of LBS users. For a given region, which is divided into a grid of cells, the key idea of the DDA algorithm is to calculate both the vertices and the edges of each cell in the grid. The DDA algorithm then randomly selects some of the cells as dummy locations. To select strong dummy locations and achieve k-anonymity, the DDA algorithm selects k cells of equal area.

Gutscher et al. proposed the idea of coordinate transformation [33], in which the users apply some geometric operations, such as shifting or rotating, over their locations before sending them to the server. To retrieve the original locations, inverse transformation functions are used. Similar to [33], the work [34] proposed a solution that allows the user to protect his/her real position using mathematical operations. These mathematical operations include enlarging the radius, shifting the center, increasing the radius, or applying double obfuscation (i.e., mixing the shifting center with any of the other operations).

Cryptographic privacy approaches utilize encryption to protect the locations of the users. Mascetti et al. [35] proposed an approach to notify users when friends (also called buddies) are within their proximity without revealing the current location of the user to the server. To achieve this, the authors assume that each user shares a secret with each of his or her buddies and use symmetric encryption techniques. Another approach was provided in [36], manipulating the problem of dealing with untrusted server. The authors based their approach on the distributed management of position information using the concept of secret sharing. The key idea of this approach is to partition the location information of the user into shares, which are then distributed among a set of untrusted servers. To recover the positions, the user needs the shares from multiple servers.

Caching-based privacy protection is considered a technique used under user-based approaches. 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 if no answer is kept by the other peers.

III. Proposed System

This section is organized so that the threat model is defined first. Next, the proposed system architecture is provided, and the details of our proposed approach are discussed. Finally, the proposed architectural details are illustrated using a sequence diagram.

A. Threat Model

Here, four main parts are defined: (1) the identity of the attacker; (2) the objective of the attacker; (3) the type of the attack; and (4) the capabilities of the attacker.

The attacker is the LBS server. Its goal is collecting sensitive or personal information about the LBS user (by detecting the real location of the LBS user). Since the attacker does not modify the collected personal information, this type of attack is passive. The attacker stacks the collected information to be converted later into actual attacks in our realistic life, such as muggings or thefts, as shown in Fig. 3.

Table 1 summarizes the capabilities of the attacker.

According to this defined threat model, the proposed system is provided as explained below.

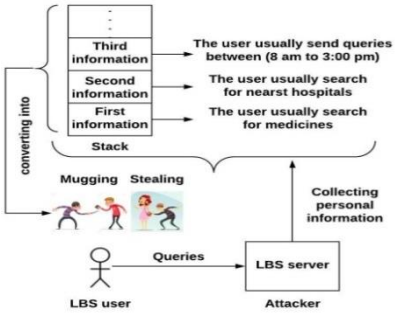


Fig. 3. Collecting and Converting Personal Information into Actual Attacks.

TABLE I. CAPABILITIES OF THE ATTACKER

Capability NO	Description
1	Tracking the real location of the LBS user.
2	Applying the location homogeneity attack.
3	Applying the semantic location attack.

B. The Proposed System Architecture

The framework of the system is composed of a number of LBS users (U_{LBS}^{number}) who are located in an area divided into $(n \times n)$ cells. The LBS users utilize LBS enabled applications, which are installed on their mobile devices. The devices of the LBS users are connected via a network. An LBS user sends a query of the following form: $Q_{(T-stamp)}^{(ID,RL,POI,R)}$, where ID is the identity of the LBS user, RL is the real location of the LBS user, POI is the point of interest, R is the range, and $T-stamp$ is the time at which the query is sent. Thus, we can say, for example, the (111-LBS) user that is located in (King Fahed Hospital) sends a query (at 9 AM), asking for (the nearest four restaurants) within a circle that has (a radius of 2 KM). After handling the sent query, the LBS server feeds back the LBS user with the corresponding response. Fig. 4 illustrates this scenario.

The proposed system is managed by three components: $Genrator_{DL}$, $Buldier_{DQ}$, and $Finder_{FQA}$. Table 2 shows the three components, their tasks, and where they are installed.

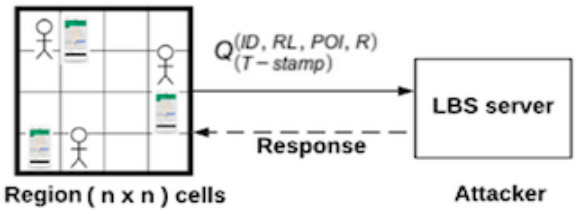


Fig. 4. Form of the Sent LBS Query.

TABLE II. COMPONENTS

Component Name	Task	Installation
$Genrator_{DL}$	Generating strong dummy locations.	Each mobile device.
$Buldier_{DQ}$	Building dummy queries.	Each mobile device.
$Finder_{FQA}$	Searching for the answer of a future query.	Access point.

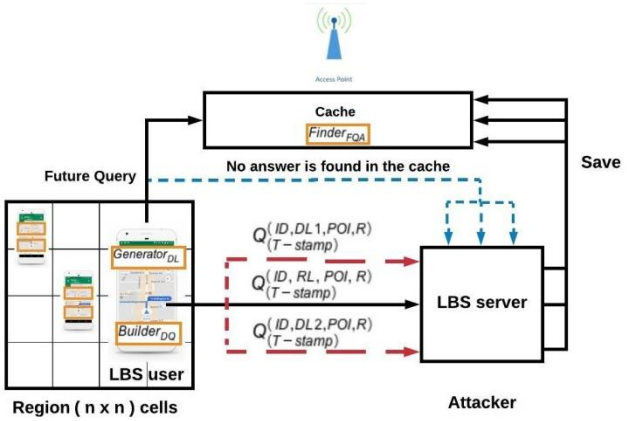


Fig. 5. The Proposed System Architecture.

Fig. 5 illustrates the proposed system architecture.

C. The Roles of Components

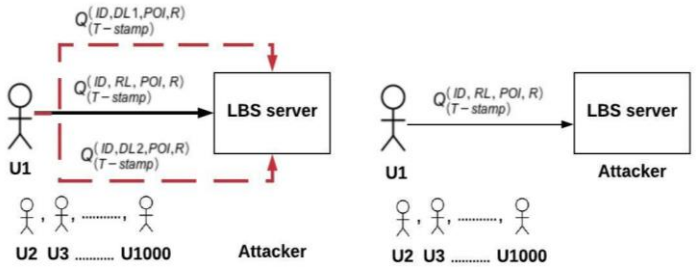
To protect the user's privacy through protecting the privacy of the location, the LBS user deliberately sends multiple queries based on dummy locations to an LBS server. Thus, the LBS server will not be able to recognize the real location of the user among the dummy locations. In this context, two important issues arise:

1) The trade-off between the performance and the achieved privacy protection level.

2) The generation of dummy locations must be robust against deductive attacks, such as homogeneity location attacks and semantic location attacks.

To explain the first issue, suppose that the number of LBS users is ($U_{LBS}^{number} = 1000$). In the case of privacy protection (Fig. 6(a)), if each user sends 3 queries to the LBS server (one of them is constructed based on the real location and the others are based on dummy locations), the total number of queries sent to the LBS server is $(1000 \times 3 = 3000)$ queries. If the privacy protection is ignored (Fig. 6(b)), each user only sends a single query (constructed based on the real location). Consequently, the total number of queries sent to the LBS server is $(1000 \times 1 = 1000)$ queries.

As a result, it is very clear that there is a trade-off between performance and privacy protection. In other words, the increased number of queries sent to the LBS server (for privacy protection purposes) will result in both low performance (i.e., long response time) and pressure on the network (i.e., network overhead).



(a) A Case of Privacy Protection. (b) Ignoring the Privacy Protection.

Fig. 6. Trade-off between Performance and Privacy Protection.

Role of the $Finder_{FQA}$ component: This component is responsible for searching for the answer of the future query, as shown in Fig. 5. To complete this task, the $Finder_{FQA}$ component uses a bloom filter technique [39]. This technique depends on a hash [k= key, V= value] that can give a direct answer about the existence or non-existence of an element within a given range. Therefore, no time is wasted in searching if the element does not exist. In this paper, the key is the Future Query (FQ), the value is the Answer of the Future Query (AFQ), and the range is represented by an array that stores the Answers of the Historical Queries (AHQ) that are answered by the LBS server. Fig. 7 illustrates the key idea of the bloom filter technique that is adopted in Fig. 5.

Depending on the bloom filter technique, the $\text{Finder}_{\text{FQA}}$ component contributes to enhance the performance because the time to answer the future query from the cache is shorter than the time to answer the future query by the LBS server. On other hand, we prevent the LBS user from dealing with the LBS server (attacker), which in turn contributes to protect his or her privacy.

Algorithm 1 shows a pseudo code for the task of the $\text{Finder}_{\text{FQA}}$ component.

Algorithm 1: Bloom-Based Search (BBS) algorithm.

Input: Future query (key).

Output: answer of future query (value).

1: **while** (cache <>∅) do
2: begin
3: val=hash(key);
4: **end while**
5: answer of future query=val;
6: return answer of future query;

Role of the $Genrator_{DL}$ component: This component is responsible for generating strong dummy locations, which is related to the second issue. To illustrate the problem of generating strong dummy locations, let the previous area be divided into $(n \times n)$ cells consisting of different sub-areas, so that each landmark is formed by combining a number of cells as shown in Fig. 8.

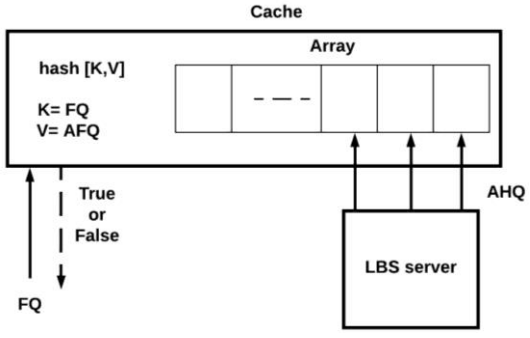


Fig. 7. Using the Bloom Filter Technique.

In Fig. 8, there are four sub-areas, which are the Restaurant Area (RA), the Medical Area (MA), the University Area (UA), and the Sport Area (SA). In addition, the LBS user is located in the MA (i.e., the real location) as are three dummy locations (D_1, D_2, D_3). The process of selecting the three dummy locations puts the privacy of the LBS user in danger of a homogeneity location attack because the attacker can infer that the LBS user suffers from a health problem (for example), since his real location and the selected dummy locations are all located in a homogeneous sub-area. Moreover, a semantic location attack can be easily successful because (for example) if the attacker noticed that all the sent queries (that are constructed based on both the real location of the user and the selected dummy locations) are issued between 8 am and 3 pm, the attacker can know the hours of the LBS user's work day. Consequently, the attacker knows the period spent by a user outside of his home, which in turn enables the attacker to rob it for example.

To solve the second issue and to ensure a high resistance against both the homogeneity location attack and the semantic location attack, we need to select (generate) strong dummy locations, as described below.

The key idea is to select dummy locations that belong to different sub-areas, as shown in Fig. 9.

To accomplish this, let each cell from the area divided into $(n \times n)$ cells be linked with a query probability $(C_{qp}^{i} | i = 1,2,\dots,n \times n)$. The C_{qp}^{i} means the probability of querying POIs from a cell in the past. Fig. 10 shows the query probabilities of all cells, and many cells may have the same query probability.

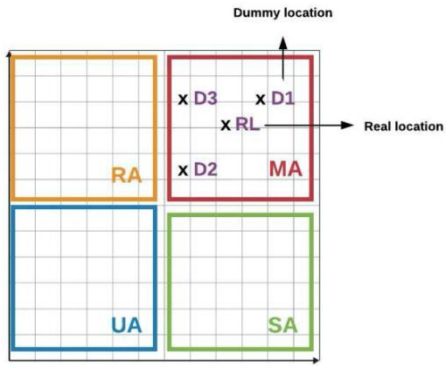


Fig. 8. Sub-Areas Formed by the Cells.

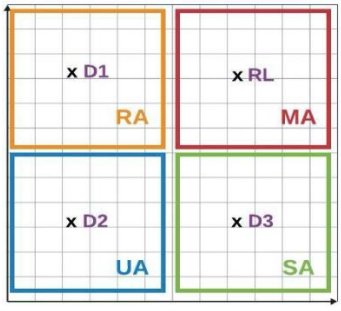


Fig. 9. The Three Dummy Locations belong to Different Sub-Areas.

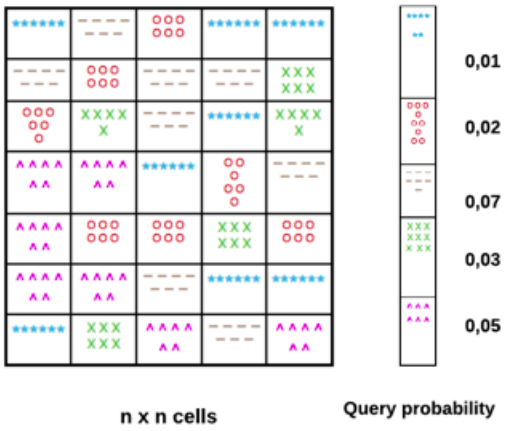


Fig. 10. Cells with Corresponding Query Probabilities.

Suppose that the LBS user is located in a specific cell, which has a specific query probability. The process of selecting the cells (as dummy locations) that have the same query probabilities as the cell where the LBS user is located ensures that the dummy locations cannot be recognized from the real location of the user. Fig. 11 illustrates this idea.

Mathematically, the key idea of preventing the attacker from recognizing the real location among the dummies is formally represented by the Entropy value. Entropy is given by:

$$\mathrm{ENT} = -\sum_{i=1}^{k} C_{qp}^{i} \times \log_2 \times C_{qp}^{i} \quad (1)$$

Compared to different query probabilities, when the query probabilities of the all k locations (real location and k-1 dummy locations) are equal, the ENT value increases. This, in turn, means a higher privacy protection level. The set of locations that meet this criterion is called the Initial Candidate Dummy Set (ICDS). Formally, ICDS is defined as:

$$ICDS = \coprod_{i=1}^{a} C_i | where\, C_{qp}^{i} = RC_{qp}\,, a < n \times n \quad (2)$$

where RC_{qp} refers to the query probability of the cell where the real location of the LBS user is located.

Problem arising from a location homogeneity attack

There is a problem related to selecting the three dummy locations in Fig. 11. This problem is highlighted when the attacker applies the location homogeneity attack, as defined in the threat model above (Table 1). Specifically, the selected three dummy locations are near enough to the real location of the LBS user and to each other's. This means that a location homogeneity attack can be easily applied at the attacker side to break the defense that is created (by the selected three dummy locations) to protect the privacy of the LBS user. In other words, the selected dummy locations in Fig. 11 are weak.

To solve this problem, we need to select the dummy locations so that they are widely spread over the $n \times n$ cells. To satisfy this condition, the dummy locations must be selected so that each dummy belongs to a different sub-area and the query probability of the LBS user's real location equals the query probabilities of the selected dummies. In this paper, a Safe Cycle-Based Approach (SCBA) is proposed to generate strong dummy locations, as shown in Fig. 12.

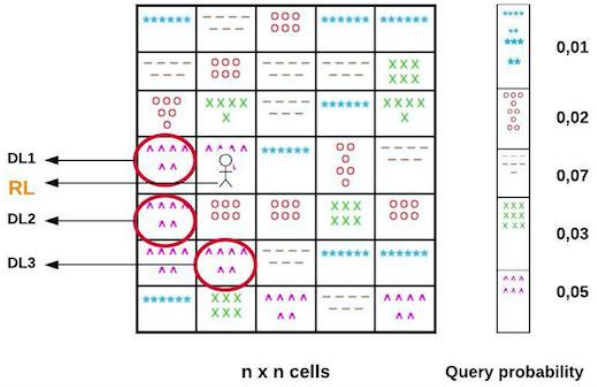


Fig. 11. Selecting Dummy Locations based on the Same Query Probability Values.

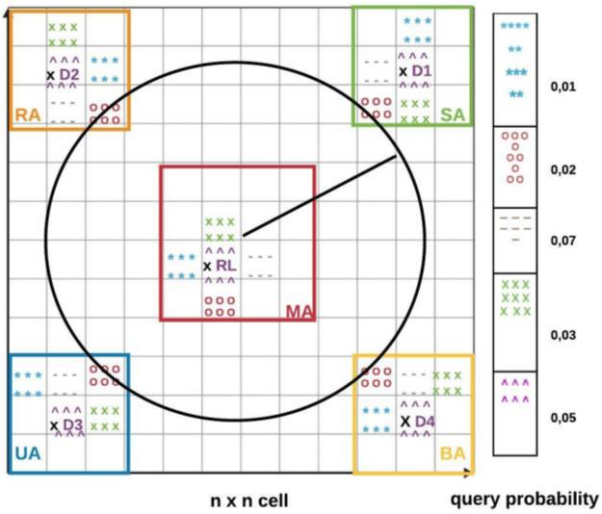


Fig. 12. Generating Strong Dummies based on the SCBA.

As shown in Fig. 12, there are five sub-areas, which are RA, SA, UA and the Business Area (BA). The real location of the LBS user is in a cell that has a query probability equal to 0.05 and belongs to the MA sub-area. The safe cycle has the following two properties: (1) its center is the real location of the LBS user; and (2) its radius is long enough that the circumference intersects with different sub areas (i.e., RA, SA, UA and BA). The four dummy locations (D_1, D_2, D_3, D_4) are selected out of the circumference of the safe cycle, belong to different sub-areas and have the same query probability as the real location.

Formally, the set of locations that meet these criteria is called the Second Candidate Dummy Set (SCDS), which form a subset of the ICDS. The SCDS is defined as:

$$SCDS = \coprod_{i=1}^{b} C_i | where\, C_{qp}^{i} = RC_{qp}\,, b < a, C_i \in SubA_i \quad (3)$$

where $SubA_i$ refers to different sub areas.

The Actual Dummy Locations Set (ADLS) is then formed by randomly selecting the (k-1) dummies from the SCDS. The ADLS is defined as:

$$ADLS = rand\,(\coprod_{i=1}^{k-1} C_i | where\, C_i \in SCDS) \quad (4)$$

Problem arising from a semantic location attack

The same problem arising from a location homogeneity attack also arises from a semantic location attack, in that the attacker exploits the time stamps attached to the sent queries to infer additional personal information about the LBS user.

The safe cycle ensures a resistance against the semantic location attack because each query that is created based on each selected dummy location has the same time stamp as the query that is created based on the real location. In other words, the all created queries are issued at the same moment. Therefore, the attacker (LBS server) will receive a package of a mixture of queries, all of them issued at the same moment. Consequently, the attacker cannot collect private information when trying to depend on temporal information.

Algorithm 2 shows a pseudo code of the task of the $Genrator_{DL}$ component.

Algorithm 2: Safe Cycle-Based Approach (SCBA) algorithm.

Input: C_{qp}^{i} query probability for each cell, RC_{qp} real location, k level of privacy protection.

Output: $ADLS$.

1: $ICDS = SCDS = ADLS = \emptyset$;
2: sort cells based on their query probabilities;
3: **for** (i=1; i < $n \times n$; i ++)
4: **if** ($C_{qp}^{i} = query\ probability\ RC_{qp}$) then
5: add C_i to $ICDS$;
6: **end if**
7: **end for**
8: create safe cycle ($RC_{qp}, radius$);
9: **while** ($ICDS$ < > $\emptyset$) do
10: **if** ($dis\ (RC_{qp}, C_i) > radius$ && ($C_i \in SubA_i$)) then
11: add C_i to $SCDS$;
12: **end if**
13: **end while**
14: **for** (i=1; i ≤ $k-1$; k ++) do
15: **while** ($SCDS$ < > $\emptyset$) do
16: randomly select C_i;
17: add C_i to $ADLS$;
18: **end while**
19: **end for**
20: return $ADLS$;

Role of the $Buldier_{DQ}$ component: This component is responsible for building queries based on the dummy locations produced by the $Genrator_{DL}$ component. These queries are called dummy queries.

D. Architecture Details

Fig. 13 shows the interaction between the $Genrator_{DL}$ and $Buldier_{DQ}$ components in the case of answering the query by the LBS server.

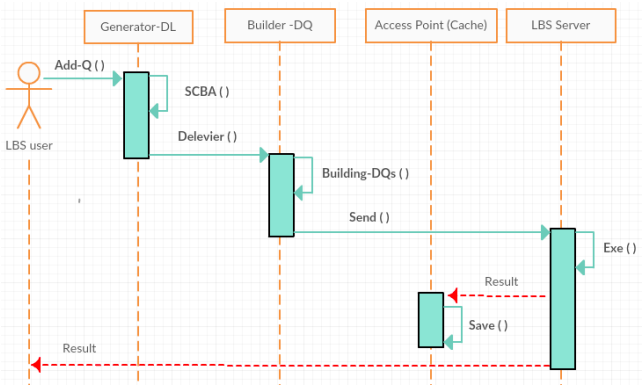


Fig. 13. Answering the Query by the LBS Server.

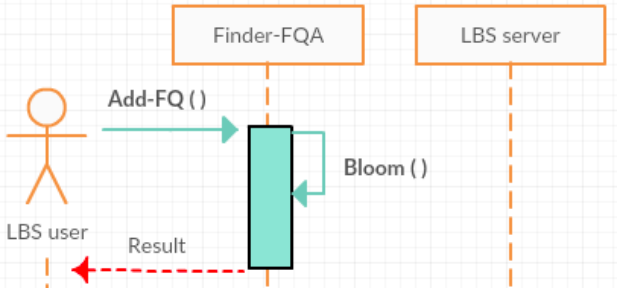


Fig. 14. Answering the Future Query by the Cache.

Fig. 14 shows the answering of the future query by the cache.

IV. Security Analysis

In this section, we discuss the resistance of the proposed SCBA algorithm against both the homogeneity location attack and the semantic location attack. In addition, we discuss the case in which the attacker tries to reverse the SCBA algorithm to break it. We define some conditions, which, when satisfied, ensure the successes of location homogeneity attacks and semantic location attacks.

Location homogeneity attack: For a given two different locations (loc_1, loc_2), this attack succeeds if the following conditions are satisfied: (1) the probabilities of the two locations (to be a real location) are different, whatever they are; and (2) the two locations belong to the same heterogeneous sub-area. When the SCBA algorithm selects k-1 dummy locations, the probability of each dummy to be the real location is $\frac{1}{k}$. In addition, the probability of locating each dummy in a cell (that may contain the real location) is C_{qp}^{i}. Consequently, the first condition is not satisfied. For the second condition, the SCBA algorithm ensures that the two locations are outside the safe cycle and belong to different heterogeneous sub-areas. Consequently, the second condition is not satisfied. Since the previous two conditions are not satisfied, the location homogeneity attack fails.

Semantic location attack: For two different given locations (loc_1, loc_2), this attack succeeds if the two conditions that adjust the success of a location homogeneity attack are satisfied as well as the following third condition: the moments at which the two queries (that are built based on the

$loc_1 and\ loc_2$ and issued) are different. Since the semantic location attack includes the location homogeneity attack, the first two conditions of a semantic location attack success are not satisfied. For the third condition, the queries that are built based on the k-1 dummy locations (that are selected by the SCBA algorithm) are issued at the same moment to be packaged and sent together to the LBS server. Consequently, the third condition is not satisfied. As a result, the semantic location attack fails.

Reversing the SCBA algorithm: When attackers try to break the SCBA algorithm by reversing it, they fail because the final (k-1) selected dummy locations are obtained in a random way from the actual set of the dummy locations (ADLS). This randomization ensures the uncertainty in the process of selecting the final dummy locations, which in turn enforces the random guessing of the real location at the attacker side.

V. Used Metrics

We use two kinds of metrics, which are privacy metrics and performance metrics, as explained below.

A. Privacy Metrics

We use two privacy metrics. The first one is the Entropy (ENT), which was previously defined in Section 1. A higher ENT value reflects a higher privacy protection level. A lower ENT value reflects a lower privacy protection level. In addition, the ENT value increases as the k value increases, where k refers to the k-anonymity level (or number of selected dummy locations including the real location).

The second privacy metric is inspired by the Entropy metric and is called Safe Side (SS). For a given LBS user, if the ENT value is equal to or higher than a predefined threshold, then the LBS user is considered in a safe side in regards to the privacy protection level. Otherwise, the LBS user is considered in a dangerous side. Formally, the SS privacy metric is defined by:

$$LBS\ User^{SS}_{the=const} = \begin{cases} safe\ side, if\ value\ (ENT) \geq thr \\ dangerous\ side, if alue\ (ENT) < thr \end{cases} \quad (5)$$

Depending on the SS privacy metric, we can evaluate two privacy protection approaches based on the number of LBS users that are in the safe side. Therefore, if the SS value is high, this means that the privacy protection approach is better for the LBS user, and vice versa.

B. Performance Metrics

We use two performance metrics, which are response time ($T_{response}$) and Cache Hit Ratio (CHR).

The response time performance metric is defined as:

$$T_{response} = T_{Create\_DQ} + T_{sent\_Q} + T_{process\_Q} + T_{recieve\_QA} \quad (6)$$

where T_{Create_DQ} refers to the time of creation of the dummy query; T_{sent_Q} refers to the time the query is sent; $T_{process\_Q}$ refers to the processing time of the query; and $T_{recieve\_QA}$ refers to the time the answer of the query is received. It is worth mentioning that a low value of the $T_{response}$ means a high performance and vice versa.

The CHR performance metric is defined as:

$$CHR = \frac{NoQAbC}{NoQAbC + NoQAbS} \quad (7)$$

where $NoQAbC$ refers to the number of queries answered by the cache, and $NoQAbS$ refers to the number of queries answered by the server. The sum of $NoQAbC$ and $NoQAbS$ refers to the total number of queries involved in the system. It is worth mentioning that a high value of CHR means that most of queries are answered by the cache, which in turn leads to a high performance.

VI. Experimental Results and Evaluations

In this section, we present a brief description of the simulation setup, and then we provide the results depending on the metrics that are defined above. The results are presented and discussed in comparison with similar approaches.

A. Simulation Setup and Configuration

We use the R programming language to implement the proposed privacy protection system. The performance evaluation is simulated on a machine with properties as summarized in Table 3.

We used the Brightkite dataset [40]. The original dataset consists of 7.3 million rows and five columns: user ID, chkintime, latitude, longitude, and locid. We downloaded 10000 user instances. The query probability is generated randomly. Furthermore, Table 4 shows the parameter values.

TABLE III. Properties

Component Name	Description
Processor	Intel.
Number of cores	I5.
Speed	1.4 GHz.
Ram	4 GB 1600 MHz DDR3.
Operating system	OS X Yosemite.

TABLE IV. Configuration

Component Name	Description
Number of cells (n × n)	150 × 150.
Number of users	10,000.
Threshold of SS	4.

B. Evaluations and Discussion

To adjust the evaluations, we select three approaches presented in the related work section for comparison purposes. Table 5 summarizes the selected approaches.

TABLE V. Selected Approaches Description

Approach Name / Ref	Publishing Year	Used Technique
Realistic Dummy Selection (ReDS) [32].	2016	Dummies
Random Dummy Selection (RaDS) [37].	2017	Dummies
Hiding in Mobile Crowd (HMC) [38].	2014	Caching

1) Privacy metrics-based evaluations: Based on the ENT privacy metric, we evaluate the SCBA, ReDS, and RaDS approaches. In addition, we evaluate the three previous approaches under the SS privacy metric.

Fig. 15 shows the corresponding entropy values under increasing K values with a step equal to 3.

ENT-based discussion: The relationship between ENT and K is that the value of ENT increases when the K value increases. Fig. 15 reflects this fact in the all approaches involved in the comparison. However, the proposed SCBA performs the best because of the constraints that are applied on the selected dummy locations, in which the selected dummies are restricted to be out of the circumference of the safe cycle. Specifically, the constraint that enforces the dummies to have the same query probabilities as the real location, in the process of selection, is the major reason behind the higher values of entropy. Since this condition is not considered in either the ReDS or RaDS approach, the corresponding entropy values are less than those in the SCBA. The RaDS approach performs the worst among the approaches because it selects the dummy locations in a random way. Since the process of dummies' selection is not adjusted by any constraint, the ENT values depend on the current query probability. This in turn leads to the lowest ENT values. Sometimes it happens that the selected dummy locations have the same query probability (or close to each other's). This can occur only by chance, which explains why some of the ENT values are higher than those generated in the ReDS approach. The ReDS approach outperforms the RaDS approach because the generated dummies rely on the actual trajectory of the LBS user's motion. Since the trajectory covers wide area, it passes through many cells that have the same query probabilities. This results in higher ENT values.

Safe Side (SS)-based discussion: Under the threat of a location homogeneity attack, the increased number of LBS users in a step equals 50 and fixing the threshold of ENT to be 4 with k=6, we evaluate the resistance of the three approaches.

Fig. 16 shows that the proposed SCBA has the highest number of LBS users that are at the safe side. To make this clearer, we arrange and calculate the percentages of safety for each approach. Table 6 summarizes the obtained results.

From Table 6, we can infer that the safety percentage of the SCBA approach varies in the range of [84 % - 98 %], and in ranges of [40 % - 73 %] and [15 % - 42 %] for the ReDS and RaDS approaches, respectively. From these safety percentages, it is obvious that the SCBA has the highest resistance against location homogeneity attacks because of the good design of the SCBA approach according to the factors that the attacker may exploit to infer personal information. In other words, the selected dummy locations weaken the ability of the attacker to collect personal information about the LBS user since each dummy belongs to a different sub-area and is outside the circumference of the safe cycle. Compared to the RaDS approach, the ReDS approach has a higher resistance against location homogeneity attacks. This can be justified by the nature of the generation of the dummy locations, in that they are generated along with the trajectory motion of the LBS user. During the motion, different sub-areas are passed by the trajectory. This leads to a higher resistance against the location homogeneity attack. Meanwhile, in the RaDS approach, the dummies are selected statically without any consideration of the sub-areas where they are located. As a result, we can rank the previous approaches according to their resistance against the location homogeneity attack as follows: the SCBA approach comes in on top, followed by the ReDS approach, and the RaDS approach comes at the end.

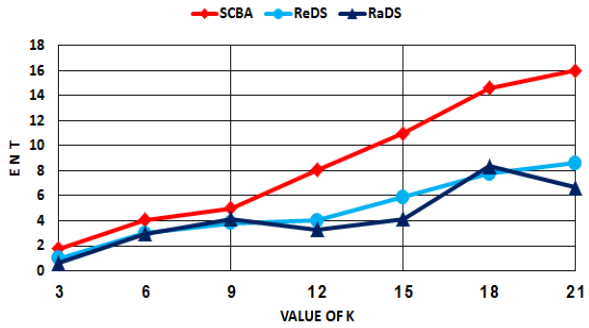


Fig. 15. ENT Value vs. K Value.

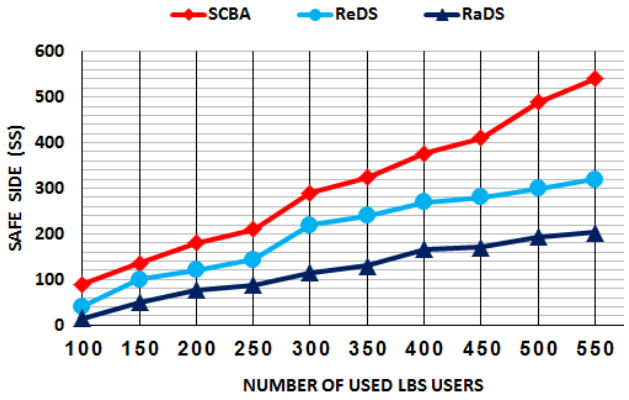


Fig. 16. Resistance Against a Location Homogeneity Attack, K=6,Thr=4, Step of Increasing U_{LBS}^{Number}=50.

TABLE VI. SAFETY PERCENTAGES UNDER A LOCATION HOMOGENEITY ATTACK THREAT

Approach / NO. of Users	SCBA		ReDS		RaDS	
	SS value	*SS%*	*SS value*	*SS%*	*SS value*	*SS%*
100	90	90 %	40	40 %	15	15 %
150	137	91 %	100	67 %	50	33 %
200	180	90 %	120	60 %	77	39 %
250	210	84 %	143	57 %	88	35 %
300	290	97 %	220	73 %	115	38 %
350	324	93 %	240	69 %	130	37 %
400	377	94 %	270	66 %	166	42 %
450	411	91 %	280	62 %	170	38 %
500	489	98 %	300	60 %	194	39 %
550	540	98 %	320	58 %	203	37 %

Using the same parameters used to test the resistance of the three approaches against the location homogeneity attack, we test them under the threat of a semantic location attack. Fig. 17 illustrates the evaluations.

Again, Fig. 17 shows that the all three approaches are negatively affected by the semantic location attack. However, the proposed SCBA approach has the highest number of LBS users that are at the safe side. Specifically, the SCBA approach is negatively affected a small amount. In contrast, the ReDS and RaDS approaches are highly and negatively affected by the semantic location attack. Following the same strategy, we arrange and calculate the percentages of safety for each approach. Table 7 summarizes the obtained results.

From Table 7, we can infer that the safety percentage of the SCBA approach varies in the range of [80 % - 96 %] and in ranges of [20 % - 50 %] and [9 % - 32 %] for the ReDS and RaDS approaches, respectively. Again, it is obvious that the SCBA has the highest resistance against semantic location attacks (or the lowest decrease in the safety percentage). The reason behind the small decrease in the safety percentage of the SCBA approach is that it was originally designed to be robust against semantic location attacks. Accordingly, the time stamps attached to both the real query (constructed based on the real location) and the dummy queries (constructed based on the dummy locations by the builder component) are the same (i.e., all of them are issued at the same moment). Regarding the large decrease in the safety percentage for both the ReDS and RaDS approaches, it is justified by the poor design against semantic location attacks. However, the ReDS approach has a higher resistance against semantic location attacks when compared to the RaDS approach because the ability of the attacker to employ temporal information (under tracking moving objects terms) to collect personal information is weak. Meanwhile, the ability to track stationary objects (in the RaDS approach) is strong, which is the reason for the lowest resistance against semantic location attacks with a maximum decrease in the safety percentage. The rankings of the three approaches according to the resistance against semantic location attacks are the same as that inferred for location homogeneity attacks.

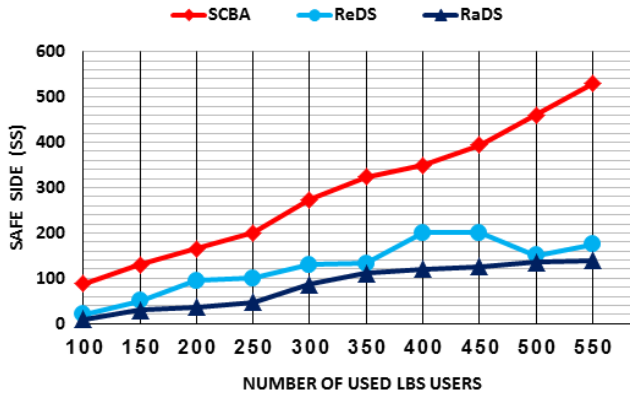


Fig. 17. Resistance against a Semantic Location Attack, K=6, Thr=4, Step of Increasing U_{LBS}^{Number}=50.

TABLE VII. SAFETY PERCENTAGES UNDER A SEMANTIC LOCATION ATTACK THREAT

Approach / NO. of Users	SCBA		ReDS		RaDS	
	SS value	*SS%*	*SS value*	*SS%*	*SS value*	*SS%*
100	88	88 %	20	20 %	9	9 %
150	130	87 %	50	33 %	30	20 %
200	165	83 %	95	48 %	36	18 %
250	200	80 %	100	40 %	47	24 %
300	273	91 %	130	43 %	86	29 %
350	320	93 %	133	38 %	111	32 %
400	350	88 %	201	50 %	120	30 %
450	394	88 %	200	44 %	126	28 %
500	460	92 %	150	30 %	136	27 %
550	530	96 %	175	32 %	140	25 %

2) Performance metrics-based evaluations: We evaluate the SCBA, ReDS, and HMC approaches using the performance metrics mentioned in the previous section.

Fig. 18 shows the communication cost under an increasing number of sent queries with a step equal to 10, in which the queries are randomly selected and sent to the LBS server for manipulation.

Cache Hit Ratio (CHR)-based discussion: In this paper, communication cost is a term that refers to the number of queries that are sent to the LBS server. Fig. 18 shows that all of the first ten queries (at the horizontal axes) are sent to the LBS server in the three approaches because at the beginning, the caches of both the SCBA and HMC approaches are empty. The ReDS approach performs the worst compared to the others because it does not use response caching at all. Therefore, all the queries are sent to the LBS server. Consequently, its corresponding curve increases in a linear manner. Compared to the ReDS approach, the HMC approach performs better because some of the sent queries find their answers in the caches of the mobile devices of LBS users. Therefore, the number of sent queries to the LBS server decreases as time progresses. The middle part of the curve related to the HMC approach reflects an increased number of queries that are sent to the LBS server. This can be justified by (1) the limitation of the caches of mobile devices, in which their size is less than the size access point; (2) sometimes LBS users delete the responses of the historical queries for mobile device performance purposes, and (3) LBS users may leave the collaboration session established with other peers. The SCBA provides the best performance. This is because of the uniform space of search for the answers of future queries, which is represented by the access point (i.e., the cache).

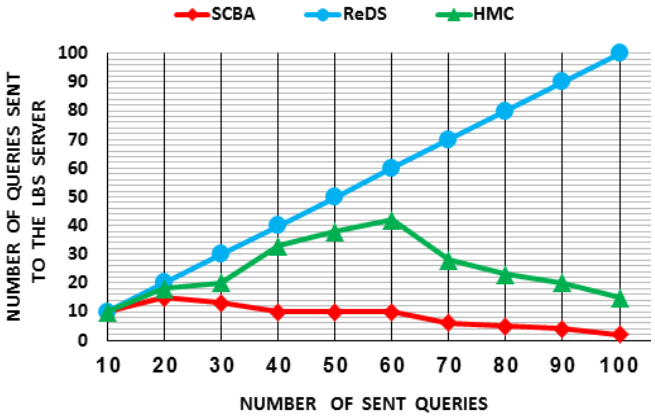


Fig. 18. Communication Cost vs. Number of Sent Queries.

Fig. 19 shows the cache hit ratio under an increasing period of running the simulation with a step equal to 5 (i.e., at different snapshots in an increased manner).

In general, the relationship between the CHR and the time progress is that the CHR increases as the time progress increases because as time progresses, the cache is filled by the answers of the historical queries, and many future queries can find their answers in the cache. Actually, Fig. 19 supports Fig. 18. The ReDS approach provided zero CHR values since no responses are cached to serve future queries. Compared to the SCBA approach, the HMC approach performs less well due to the higher number of queries that are sent, and consequently, answered by the LBS server. The SCBA approach provides the best CHR values in time progress because it has the maximum number of queries sent and, consequently, answered by the cache.

Time response-based discussion: Fig. 20 shows the time response values under an increasing number of sent queries with a step equal to 5, in which the queries are randomly selected and sent to the LBS server for manipulation and protected by a (k=3) privacy level. The first five queries are selected at the beginning of the simulation run, and the rest are selected at different snapshots.

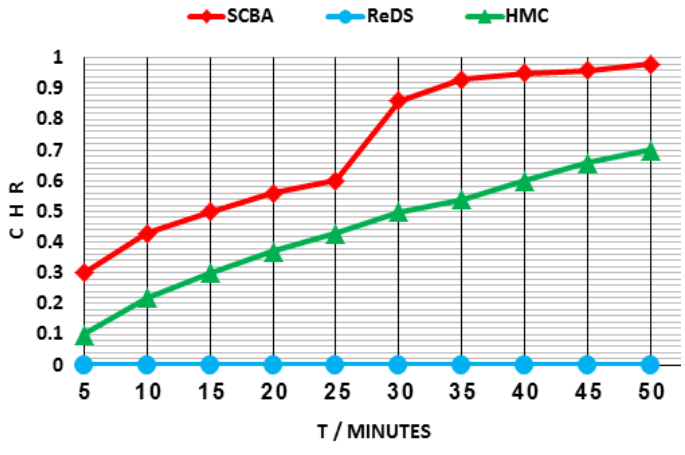


Fig. 19. CHR vs. Time Progress.

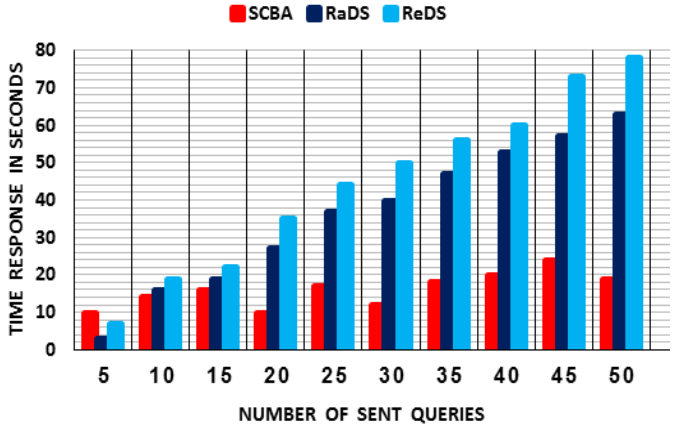


Fig. 20. Number of Sent Queries vs. Time Response, K=3.

For the first five queries, the RaDS approach performs the best. This due to two major reasons: (1) the process of selecting dummy locations in a random way takes a short time compared to the time of generating dummy locations based on the motion of the LBS user (in the ReDS approach), or compared to selecting dummy locations based on two factors (in the SCBA approach); and (2) due to the first reason, the process of creating the dummy queries (T_{Create_DQ}) requires less time when compared to the ReDS and SCBA approaches. The ReDS overcomes the proposed SCBA approach due to the sum of the four times: (1) the time of generating the query probabilities; (2) the time of forming the initial candidate set of dummy locations (i.e., satisfying the first condition of the dummies' selection); (3) the time of forming the second candidate set of dummy locations (i.e., creating the safe cycle); and (4) the time of forming the actual set of dummy locations is longer than the time generating dummies in the ReDS approach.

For the rest of the queries involved in Fig. 20, the proposed SCBA approach performs the best. The reasons are: (1) as time progresses, many future queries find their answers in the cache; (2) due to the previous reason, the time of processing the query ($T_{process\_Q}$) (i.e., the time of searching for the answer of the query, which is promoted by a bloom filter technique) is less when compared to the processing times in the ReDS or RaDS approaches; and (3) the time of creating dummy queries (T_{Create_DQ}) is zero, since there is no need to generate dummies due to the answering of the future queries by the cache (i.e., no need to protect the privacy of the LBS user against the LBS server, which is the attacker). The RaDS approach performs better than the ReDS approach for the same reasons, to justify the results of the first five queries.

VII. Conclusion

On one hand, location-based services have been paid much attention by users due to their valuable benefits in our realistic life. On other hand, researchers caution about a privacy protection concern related to the usage of location-based services. In regard to location privacy protection, we propose a new location privacy protection system. The proposed system

is managed by three main components: the $Genrator_{DL}$,. $Buldier_{DQ}$, and $Finder_{FQA}$. The $Genrator_{DL}$ component executes a novel location privacy protection method called the Safe Cycle-Based Approach (SCBA). The SCBA generates strong dummy locations based on two factors: (1) selecting dummy locations that have the same query probabilities as the real location; and (2) the selected dummy locations are outside the circumference of the safe cycle to ensure robustness against inference attacks that may be applied by the LBS server (a malicious party). To prevent dealing with the LBS server, the SCBA integrates with the cache represented by an access point. In the access point, the responses of the historical queries are stored. The cached responses are used to answer future queries. The $Finder_{FQA}$ component searches the answers of the future queries based on a bloom filter technique to enhance the response time of the privacy protection system. Based on two privacy metrics, which are entropy and safe side, the SCBA outperforms similar dummy-based approaches in terms of privacy protection level and resistance against both the location homogeneity attack and the semantic location attack. Based on two performance metrics, the cache hit ratio and time response, the SCBA approach, supported by integration with the cache, outperforms similar approaches in terms of communication cost, cache hit ratio, and response time to the sent queries.

In future work, we intend to cover a wider spectrum of attacks, such as the query sampling attack, which targets the query privacy by analyzing the sent queries, and the denial of service attack, which targets the availability of the system. In addition, we intend to deal with the man in the middle attack, in which any LBS user may acts as an attacker rather than the LBS server.

REFERENCES

[1] Wortmann, Felix, and Kristina Flüchter. "Internet of things." Business & Information Systems Engineering 57.3 (2015): 221-224.

[2] Osseiran, Afif, et al. "Internet of Things." IEEE Communications Standards Magazine 1.2 (2017): 84-84.

[3] Cui, Xiaoyi. "The internet of things." Ethical Ripples of Creativity and Innovation. Palgrave Macmillan, London, 2016. 61-68.

[4] Aly, Heba, Moustafa Youssef, and Ashok Agrawala. "Towards Ubiquitous Accessibility Digital Maps for Smart Cities." Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2017.

[5] Buhalis, Dimitrios, and Aditya Amaranggana. "Smart tourism destinations enhancing tourism experience through personalisation of services." Information and communication technologies in tourism 2015. Springer, Cham, 2015. 377-389.

[6] Shin, Kang G., et al. "Privacy protection for users of location-based services." IEEE Wireless Communications 19.1 (2012).

[7] Wernke, Marius, et al. "A classification of location privacy attacks and approaches." Personal and ubiquitous computing 18.1 (2014): 163-175.

[8] Niu, Ben, et al. "Achieving k-anonymity in privacy-aware location-based services." INFOCOM, 2014 Proceedings IEEE. IEEE, 2014.

[9] Gao, Sheng, et al. "LTPPM: a location and trajectory privacy protection mechanism in participatory sensing." Wireless Communications and Mobile Computing 15.1 (2015): 155-169.

[10] Chen, Shu, and Hong Shen. "Semantic-Aware Dummy Selection for Location Privacy Preservation." Trustcom/BigDataSE/I SPA, 2016 IEEE. IEEE, 2016.

[11] Ağır, Berker, et al. "On the privacy implications of location semantics." Proceedings on Privacy Enhancing Technologies 2016.4 (2016): 165-183.

[12] Lee, Byoungyoung, et al. "Protecting location privacy using location semantics." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[13] Pan, Xiao, et al. "Protecting personalized privacy against sensitivity homogeneity attacks over road networks in mobile services." Frontiers of Computer Science 10.2 (2016): 370-386.

[14] Paulet, Russell, et al. "Privacy-preserving and content-protecting location based queries." IEEE Transactions on Knowledge and Data Engineering 26.5 (2014): 1200-1210.

[15] Fung, Eric, Georgios Kellaris, and Dimitris Papadias. "Combining Differential Privacy and PIR for Efficient Strong Location Privacy." International Symposium on Spatial and Temporal Databases. Springer International Publishing, 2015.

[16] Ardagna C, Cremonini M, Damiani E, De Capitani di Vimercati S, Samarati P (2007) 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, pp 47–60.

[17] Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: Proceedings of the third international conference on pervasive computing (Pervasive '05), Munich, Germany, pp 152–170.

[18] Damiani ML, Bertino E, Silvestri C (2010) The probe framework for the personalized cloaking of private locations. Trans Data Priv 3(2):123–148.

[19] Gruteser, Marco, and Dirk Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking." Proceedings of the 1st international conference on Mobile systems, applications and services. ACM, 2003.

[20] Xu, Toby, and Ying Cai. "Feeling-based location privacy protection for locationbased services." Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[21] Lin, Chi, Gouge Wu, and Chang Wu Yu. "Protecting location privacy and query privacy: a combined clustering approach." Concurrency and Computation: Practice and Experience 27.12 (2015): 3021-3043.

[22] Shao, Zhou, David Taniar, and Kiki Maulana Adhinugraha. "Range-kNN queries with privacy protection in a mobile environment." Pervasive and Mobile Computing 24 (2015): 30-49.

[23] Saravanan, Shanthi, and Balasundaram Sadhu Ramakrishnan. "Preserving privacy in the context of location based services through location hider in mobiletourism." Information Technology & Tourism 16.2 (2016): 229-248.13 | P a g e

[24] Gedik, Bugra, and Ling Liu. "Protecting location privacy with personalized kanonymity: Architecture and algorithms." IEEE Transactions on Mobile Computing 7.1 (2008): 1-18.

[25] Mokbel, Mohamed F., Chi-Yin Chow, and Walid G. Aref. "The new Casper: query processing for location services without compromising privacy."Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006.

[26] Beresford AR, Stajano F (2004) Mix zones: user privacy in location-aware services. In: Proceedings of the second IEEE annual conference on pervasive computing and communications workshops (PerCom '04 Workshops), pp 127–131.

[27] Beresford, Alastair R., and Frank Stajano. "Location privacy in pervasive computing." IEEE Pervasive computing 2.1 (2008): 46-55.

[28] Palanisamy B, Liu L (2011) Mobimix: protecting location privacy with mixzones over road networks. In: Proceedings of the 27th IEEE international conference on data engineering (ICDE '11), pp 494–505.

[29] H. Kido, Y. Yanagisawa, and T. Satoh, ―An Anonymous Communication Technique Using Dummies for Location- based Services,$ IEEE Proc. Int'l. Conf. Pervasive Services, ICPS '05, July 2005.

[30] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li,―Achieving k-anonymity in privacyaware location-based services,‖ in Proc. of IEEE INFOCOM 2014.

[31] A. Pingley et al., ―Protection of Query Privacy for Continuous Location Based Services,$ IEEE INFOCOM'11, Apr. 2011.

[32] Hara, Takahiro, et al. "Dummy-Based User Location Anonymization Under RealWorld Constraints." IEEE Access 4 (2016): 673-687.

[33] Gutscher A (2006) 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, pp 354–354.

[34] Ardagna, Claudio Agostino, et al. "Location privacy protection through obfuscation-based techniques." IFIP Annual Conference on Data and Applications Security and Privacy. Springer Berlin Heidelberg, 2007.

[35] Mascetti S, Freni D, Bettini C, Wang XS, Jajodia S (2011) Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. VLDB J 20(4):541–566.

[36] Marias G, Delakouridis C, Kazatzopoulos L, Georgiadis P (2005) Location privacy through secret sharing techniques. In: Proceedings of the 1st international IEEE WoWMoM workshop on trust, security and privacy for ubiquitous computing (WOWMOM '05), pp 614–620.

[37] Alrahhal, Mohamad Shady, et al. "AES-Route Server Model for Location based Services in Road Networks." INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 8.8 (2017): 361-368.

[38] Shokri, Reza, et al. "Hiding in the mobile crowd: Locationprivacy through collaboration." Dependable and Secure Computing, IEEE Transactions on 11.3 (2014): 266-279.

[39] Singh, Amritpal, et al. "Bloom filter based optimization scheme for massive data handling in IoT environment." Future Generation Computer Systems 82 (2018): 440-449.

[40] SNAP website, (2018), avaliable: https://snap.stanford.edu/data/loc-brightkite.html.