Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 4607464, 11 pages
https://doi.org/10.1155/2018/4607464

Research Article

Double Cache Approach with Wireless Technology for Preserving User Privacy

Adnan A. Abi Sen,[1] **Fathy B. Eassa**,[1] **Mohammad Yamin**,[2] **and Kamal Jambi**[1]

[1]*College of Computer Science and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia*
[2]*Department of MIS, Faculty of Economics and Administration, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Correspondence should be addressed to Adnan A. Abi Sen; adnanmnm@hotmail.com

Received 23 February 2018; Revised 4 May 2018; Accepted 7 June 2018; Published 1 August 2018

Academic Editor: Milos Stojmenovic

Several methods use cache for decreasing the number of connections to protect privacy of user data and improve performance in Location Based Services (LBS). Many of these methods require users to trust other users or third parties, which could be servers. An intruder could be disguised as a user or a third party. In this article, we propose a new method, known as "Double Cache Approach", which uses a pair of caches to reduce the vulnerability of trust between users or third party and offers a vast improvement in privacy and security of user data in healthcare and other applications that use LBS. This approach divides the area into many cells and manages the cooperation among users within two caches at the access point with wireless communication. To demonstrate the superiority, we also provide simulation results of user queries, comparing the proposed method with those using only one cache. We believe that our approach would solve the trust problem optimally, achieve a comprehensive protection for users' data, and enhance the privacy and security levels.

1. Introduction

Healthcare is one of the critical domains [1] where Internet of Things (IoT) has enhanced its quality and usability [2, 3] with smart phones or wearable devices like RFID tags. Most of the healthcare applications use Location Based Services (LBS) for searching points or places of interest like hospitals, health centres, and pharmacies with the help of GPS technology [4–6]. This technology plays a significant role in medical emergencies, healthcare applications, and establishing contacts in other domains [7, 8]. However, the use of LBS services entails significant risks of breach of user's privacy and security [9, 10]. The attacker may be able to determine the location and track and build profile and pattern of the user's movements [11, 12]. The attackers can also gain user's personal and sensitive information like their whereabouts at a specific time, job, health conditions, financial and social status, and religion, political, and ethical inclinations [13–15], which could limit future use of e-healthcare systems [16–18]. A server provider can also breach the privacy, consequences of which could be quite serious [19, 20].

Many methods and techniques exist to protect privacy in application and services of IoT but none of them can ensure total protection. Before embarking on these methods, let us discuss the standard format of queries that are launched from clients to LBS server. As shown in Figure 1, there are three components in a query, namely, ID, Location, and Query-Type. Existing approaches are designed to defend privacy from an outer attacker, server provider, and all other types of attacks [21] by protecting one or more components. The methods or approaches for privacy of data in LBS, as in [22], may be grouped into eight classes, namely,

(i) Trusted Third Party (TTP) (also known as Cloaking area & K-anonymity);

(ii) Obfuscation and Land-marking;

(iii) Mix Zone, Private Information;

(iv) Retrieval (PIR) and Encryption;

(v) Dummies;

(vi) Cooperation between Peers;

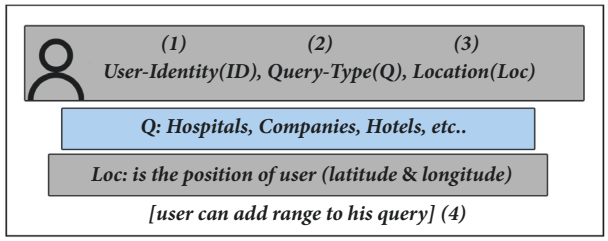

Figure 1: Query format in LBS.

(vii) Caching;

(viii) Hybrid.

Each one of them has one or more kind of limitations and consolidated open problems [21–25], which are described below.

(1) How to deal with Anonymizer or TTP without having trust?

(2) How to generate high-efficiency dummies?

(3) How to establish reliability between users and reputation of peers?

(4) Can the cache hit ratio be improved to achieve more privacy?

(5) Can the overhead computation be relaxed in the PIR approach?

In this paper we shall resolve first four problems by proposing the following:

(1) An approach, known as Double Cache, for protecting privacy in the applications of LBS from different attacks in all fields including the healthcare;

(2) A new method for generating smart dummies with high performance;

(3) A new technique for managing cache and enhancing performance;

(4) A solution for the trust issue between peers.

2. Literature Review

With the emergence of the IoT paradigm, enhancement and developments of services have taken place in several domains. In particular, healthcare has been a major beneficiary. Healthcare is now benefited by sophisticated applications and tools such as wearable sensors, which can

(i) Monitor patients and senior citizens continuously;

(ii) Help people with breathing problems to find less polluted routes;

(iii) Help in statistical or data mining processing;

(iv) Determine the likelihood of spreading a contagious viruses and diseases in specific areas

(v) Frequently update donor locations;

(vi) Help in emergencies and ambulance services;

(vii) Assist drones in transporting medical supplies to obscure locations;

(viii) Track and monitor activities of infants or children and subsequently report any associated emergencies to their parents or guardians [1, 26, 27].

Since most of these services use mobile LBS, requiring wireless connections, privacy can easily be violated by server providers through revealing the locations, movements pattern, habits, and behaviours of the people they serve [25, 28, 29].

Use of cache for preserving privacy is a relatively new approach and hence only a small number of researchers have discussed it so far. Most of classical approaches like collaboration, dummy, and K-Anonymity have used additional storage; however they have suffered from many limitations including uncertainty, overhead on user's device, disconnecting, and trust [30]. Concept of a TTP-Free class was introduced which did not need any trust with LBS server or any third party. Instead, it relied upon collaboration between users and peers to preserve privacy; however the issue of trust between peers still needed a resolution [31]. In pursuit of a resolution of trust issue, some researchers have proposed an idea of exchanging some information about previously visited area among peers to create a robust cloak area, which paved the way to consider storing some information on the client side [32]. The method in [23] used cache in TTP to store answers of some queries of users for the purpose of using them in future. It increased the privacy and performance level but resulted in decreasing the number of connections to server. This approach posed the challenges of cache management processes like data freshness, query selection, data consistency, cache hit\miss ratio, and estimation of required area to cache.

In the K-Anonymity approach, TTP would send the queries of K-Users to LBS server and store the answers for future queries [33]. This is an effective approach to protect privacy, although the cost of computation and connection turns out to be high. Use of cache was suggested to overcome this but as we know cache would require its management. To overcome the limitations of K-Anonymity, some more complex methods were suggested [34]. There are other techniques where cache is used as an aid tool in the cooperation systems to enhance the privacy and decrease the overhead on connections to LBS server [35]. In [25], mobile cache is used for saving results of the queries which can be used as answers for user queries.

Another approach known as "Mobi-Cache" is based on two ideas. The first one is an algorithm for dummies selection (DSA) in addition to an enhancing method (en-DSA) based on the Entropy of each cell (ratio of queries raised from each cell), used for optimizing the benefit of these dummies for new queries. The second idea proposed is using the cache at the access point available in each cell or specific area to store the answers of queries and dummies [36]. By having description and more details of DSA and en-DSA algorithms for careful generation, the dummies can be used or requested in the future as they achieve increased privacy and performance. The main idea is to divide the area into

many cells (equal sizes) and then calculate the Entropy metric for each one through the number of queries that are launched from it. After that, the dummies would be selected from the cells that have similar value to increase LBS server's uncertainty and use the dummies' answers in the future [37].

In the subsequent research, focus was on usage of cache, which was also divided the area into many cells, pointing each to an access point. This approach relaxed the overhead problem of dummy approach and that of the need for trusted parties [38]. In [39], Long Statistical Attack (LSA) and Regional Statistical attack (RSA) were proposed. As an enhancement of this approach, two new techniques were introduced. The first of them suggested dividing cells into four levels of privacy and changing size of each one to be bigger or smaller to achieve the required Entropy Value, and the second proposed using multinames (M-Name) for each user instead of single name (S-Name) [5]. It divided previous approaches to three classes according to the trust, namely, TTP based, Semi TTP, and Free TTP, and suggested a protocol for dealing between peers lacking trust, by hiding precise location to create a suitable cloaking area.

Collaboration between peers to obfuscate their locations into cloak area with K-users has led to more research [40], which supported continuous queries, as opposed to a snapshot, by way of using the cache. In [41], a new technique of cooperation between users by exchanging queries between themselves before sending them to a server in the Private Information Retrieval (PIR) systems was proposed to hide the identity of each one and retrieve the data from the server without revealing it. Although it enhanced the users' privacy in LBS server, it also relied upon the trust among peers in addition to overhead of using encryption.

Authors in [23] have created an integration between the cache technique and another one for exchanging queries among each pair of users to provide a solution of some problems and challenges. This includes the overhead in generating dummies and the need to trust in third party (TP) and the cache hit ratio, but at the cost of trusting another peer. Throughout this article, this approach will be referred to as P2PCache, which provided a new method to manage cache and the freshness issue of cache. A research survey of protection strategies and attack models is discussed in [42], which has outlined the following research topics in the privacy domain:

(1) Using semantics of data or locations to enhance the privacy;

(2) Preserving privacy for the location of data collection;

(3) Using an efficient indexing technique to relax the high cost;

(4) Creating a generic framework to address all privacy elements.

In view of the forging discussion, we have the following open problems in privacy domain:

(1) How to create a trusted party;

(2) How to optimize the cost of connections and overhead in some techniques;

(3) How to efficiently manage cache, freshness, and enhance the hit-ratio;

(4) How to effectively manage connections and collaboration among peers.

In this paper, as follows, we have provided solution for these problems.

3. Proposed Method

The major issues of user privacy are in the existing methods being centred around the trust and cache management.

3.1. Trust Management Issues. As mentioned earlier, ensuring trust between users and third party (server) is an open problem. The "trust" can be classified as follows:

(i) Trust with a service provider: it is regarded as the biggest threat because the service providers have all information about the users. Many techniques, as discussed before, had tried to address this issue but the problem was far from being resolved.

(ii) Trust with third party: this is known as anonymizer and used to avoid the trust with LBS server by hiding the identity of users, obfuscation their locations, generating dummy users, managing cloaking area, or helping peers to create a protected area. But this ended up shifting the problem from LBS server to another server and so did not produce an effective solution.

(iii) Trust with peers: in this case, users cooperate with each other to protect themselves. This technique as provided in [23] is superior though it still has its own issues. On one hand, the threat may still come from peers and on the other hand the management of cache is a complex procedure.

3.2. Cache Management Issues. As we discussed before, the cache was utilized to store some results of user's queries for answering future queries. In addition, it used so-called freshness technique to refresh the stored results. However, the freshness technique suffers from the following drawbacks:

(i) Sometimes the cache may not be utilized at all and so it will be cleared;

(ii) Using refresh process would affect system performance;

(iii) No distinction exists between queries that are submitted repeatedly and the ones which are submitted once only, which would adversely affect the efficiency of cache itself;

(iv) Filling cache by generated dummies would affect the cache hit-ratio adversely.

To overcome these anomalies, we propose a new technique, known as Double Cache Approach (DCA), which uses a pair of caches and replaced the freshness technique with a proposed one.

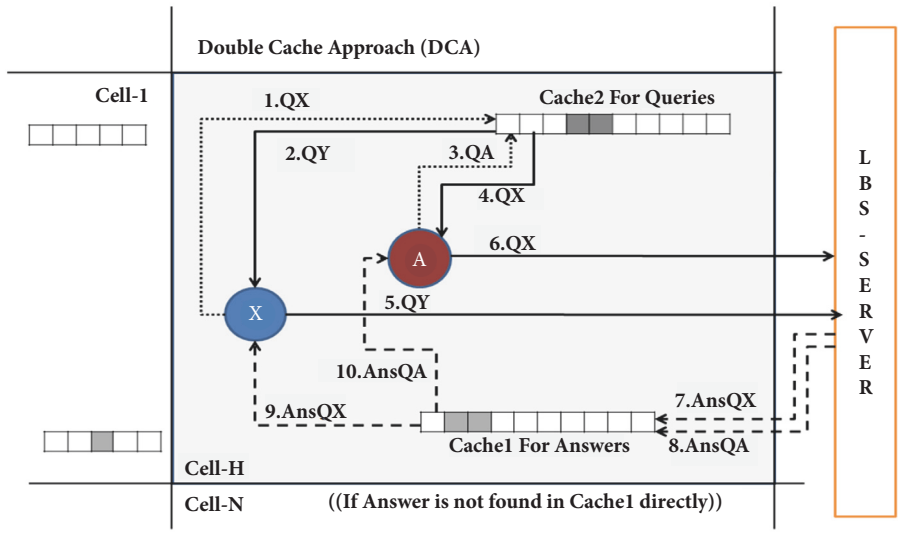

Figure 2: First scenario double cache approach.

3.3. Double Cache Approach (DCA). Initially the use of cache in privacy domain was to reduce the number of connections to LBS server, which was considered to be the biggest threat for users. In doing so, the privacy level was enhanced and so was the performance. However, this did not solve any of the open problems in the LBS system.

Before providing the details of the proposed DCA, we explain it by means of a simple example. Suppose a client or user A wants to choose a less polluted walkway without revealing their identity and the chosen walkway. For example, the query Q_A of A would be sent to another user B, who would submit it to LBS server. Consequently, server receives wrong information about B, but A would receive right answer for Q_A. To hide the information of A from B, a pair of cache would be used. This process can be repeated by A with other users. We shall call this method as the Double Cache Approach (DCA) and demonstrate the capabilities of that DCA to protect the patient location from malicious SP without effecting the quality of services. It should be noted the other known methods as those discussed in [43–46] cannot protect the privacy of A from B or SP. In reality, DCA can be seen as an enhancement of the P2PCache and removes the following anomalies of [23]:

(a) Lack of trust between peers, regarded as an open problem.

(b) Self-management for communication between peers affecting performance.

In DCA, we have a pair of caches, namely, $Cache_1$ *and* $Cache_2$, for addressing the trust issue between peers themselves and achieving new benefits. We shall use $Cache_1$ for collecting previous answers and $Cache_2$ for swapping queries between peers and managing the cooperation among them.

3.3.1. First Scenario ($Cache_2$ with First-In-First-Out). Suppose a user A wants to submit a query Q_A to LBS server. Before doing so, A should search for an answer of the query in $Cache_1$. If an answer was not found, Q_A should be swapped with Q_X, a query from some other user X in $Cache_2$, and then send Q_X to LBS. It is unlikely to find Cache2 empty but in case it was, user would submit just a dummy query to LBS server and store the query in Cache2. When an answer of Q_X was received, A would store it and look for an answer of Q_A in $Cache_1$. In the same manner X can swap Q_X with a query Q_Y of another user Y. This process is demonstrated in Figure 2.

3.3.2. Second Scenario (Cache2 with Priority). Unlike the previous case, user A can directly submit query Q_A to $Cache_2$, and wait for an answer in $Cache_1$. However, this may cause a considerable delay in getting an answer. To increase the priority of query resolution, A should submit one or (preferably) more queries of other users to LBS server. This method is shown in Figure 3. Let us calculate priority for query position in $Cache_2$. For example, let NQC be the number of current queries in the $Cache_2$, UQC the number of queries which user U has stored in $Cache_2$, and UQL number of queries which U has submitted to LBS server. So

$$Priority = \frac{\text{UQL}}{\text{UQC} + 1 + \text{NQC}} \quad (1)$$

$$Position_{in\_Cache} = \lfloor NQC - Priority * NQc \rfloor \quad (2)$$

The idea behind this second scenario is to increase the level of cooperation among peers for mutual benefit through saving of power consumption for users with limited battery/bandwidth. This situation does not arise in the first scenario, which forces users to connect LBS server. In contrast, the second scenario allows users with more resources to get more than one query from the cache and direct them to the LBS server. Users with low battery would place their queries in the cache and not connect to server.

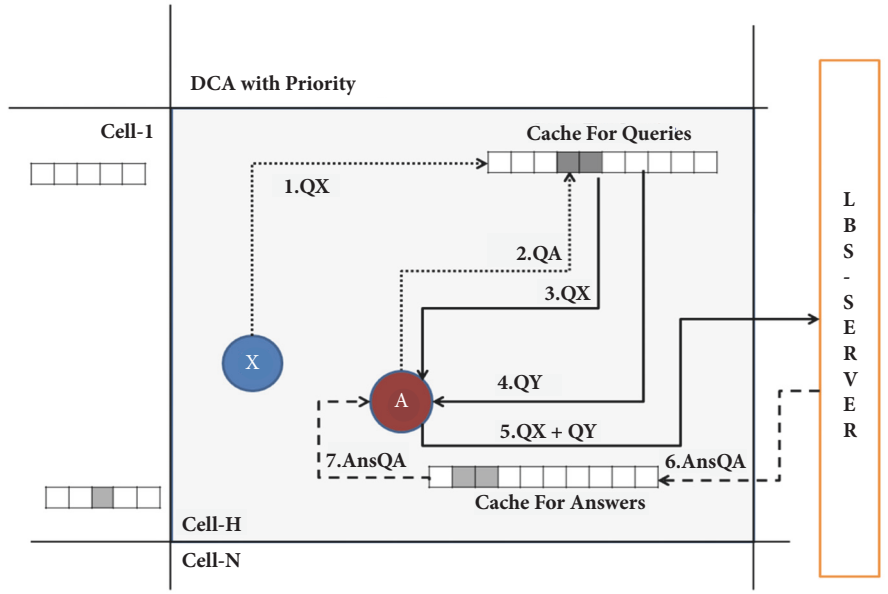

Figure 3: Second scenario double cache approach.

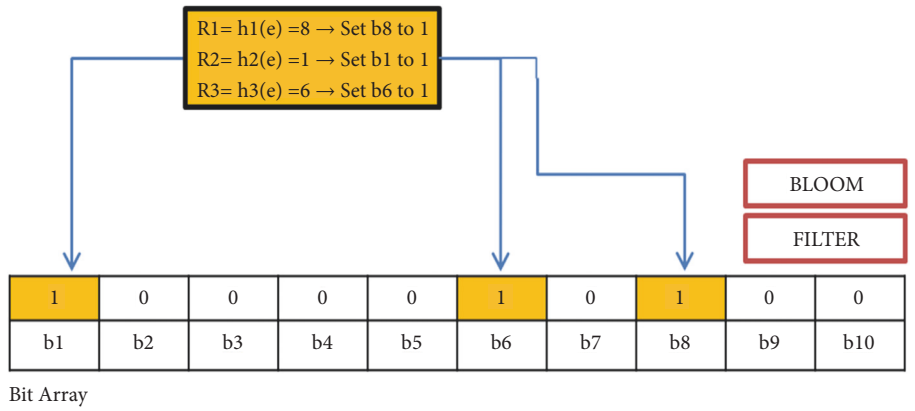

Figure 4: Hashing filter (Bloom).

According to waiting issue:

UQC = 1 often, because user usually needs answer for one query in the same time and cell.

If UQL = 1, it would revert to the first scenario.

If UQL > 1, that means the frequency of outing of queries to LBS server would outnumber the incoming quires to cache ⟶ NQC will tend to 1, eliminating need for any waiting.

If UQL=0, the query will be at the position NQC, which is the last of position in the cache. In this situation, user might wait until another user gets more than one query from cache.

Therefore, the waiting would occur only rarely, and if it did, it would be in the case when none of the other users would have taken more than one query from the cache, which is not realistic in the LBS system because all users would want to send more dummies to the LBS server. Moreover, if user was patient, they could easily take a query from the cache and send it to the LBS server to return to first scenario. Note that both scenarios are starvation-free.

3.4. Proposed Cache Management Technique. The DCA regards cache as a special queue and eradicates previous drawbacks by exploiting its size. It will hence store answers in Cache_1 and if Cache_1 was full, the oldest answer (Min ID) would be removed and the new answer would be added and Max ID would be incremented by 1. However, if answer already existed in Cache_1, then only ID of answer would be changed to MaxID+1. We have also used Bloom filter (Figure 4) (Hashing Technique) to increase the efficiency of cache and decrease the response time, especially in the miss-hit cases, where it would directly find out if the required answer for specific query was removed from cache or not determined by (O(I)), where O in [38] is the number of

Input: Query (q) of User A **Output**: Answers of q
// Search-Function
1 if (q ∈ Query in $Cache_1$) then
2 q_{ID} = (Max_{ID} in $Cache_1$) + 1
3 Return q_{ANS}
4 end
5 else
6 Insert (q in $Cache_2$)
7 Get first q~ of $Cache_2$ // Unknown User
8 Insert $q\sim_{ANS}$ to $Cache_1$ with Max_{ID}
9 Query with Min_{ID} in the $Cache_1$ will be deleted.
10 end
11 Recall Search-Function
12 End Function

Algorithm 1

elements in cache. Bloom filter directly gets an answer for O(I), if the element existed in cache or not before I search.

3.5. Algorithm of Cache Management. See Algorithm 1.

3.6. Benefits of DCA. Use of additional cache to handle query swapping among peers, with a new technique for managing caches, resolves the issue of trust between peers and enhances performance of the whole system without having to generate dummies. DCA facilitates superior collaboration among peers, unlike that in [23], where the user had to search and contact a peer directly. In particular:

(i) answers of queries would be kept in $Cache_1$ permanently without having to refresh them;

(ii) DCA would utilize the whole size of $Cache_1$;

(iii) no dummy queries would be stored in the $Cache_1$;

(iv) user would not need to trust other peers;

(v) query swapping would improve the performance and the cache hit-ratio, in addition to increasing the level of privacy by feeding LBS with misleading information about users.

3.7. Mathematical Proof for Superiority of DCA. We divide the area for N different cells and assume that q_i is the probability of the number of queries produced from the cell i, so

$$\sum qi = 1: \quad i = 1 \text{ to } N \tag{3}$$

If user sends k queries to server, one of them real and others (k-1) dummies, then

$$Pi = \frac{qi}{(\sum qi)}: \quad i = 1 \text{ to } k \tag{4}$$

where Pi denotes the probability of qi being a real query from k ones which will be (1/k) only if the whole queries have same priority.

If H is the Cache hit ratio, then

$$H = \frac{(\text{queries answered by cache})}{(\text{queries answered by cache} + \text{queries sent to server})} \tag{5}$$

The privacy level, according to Entropy metrics (E), lies in [0, 1], where E = 0 means that a server knows the user's query, and E = 1 means it does not. Then

$$E = \sum_{i=1}^{k} (pi.\log_2 (pi)) \tag{6}$$

Earlier methods have tried to increase E [38] and added k-1 dummies to the real query before sending it to server to make E larger. In case of the method of [23], E would attain maximum value of 1 for server, because query would not be sent by the user. However, the E value peers will be zero because when swapping user sends his real query to other peer so Pi = 1 thereby would be zero, which is a drawback in [23]. DCA has solved this issue by avoiding the direct dealing between peers and so E = 1, both for server and peers. This would achieve more privacy in addition to better performance. Note that the privacy and performance can be increased in two ways, namely, if H or E becomes larger.

4. Threats Models and Security/Privacy Analysis of DCA

Service providers, who can access, manipulate, and reveal user data from LBS, can be classified as active attackers, whereas the other users who only eavesdrop, may be known as passive attackers. Here we discuss the effectiveness of DCA in dealing with these kinds of attackers in the following scenarios:

(i) Semantic context: attacker has personal information like age, work history, etc. of the user. In DCA a user is not required to send personal information and exchange their query with random and this kind of threat is eliminated.

(ii) Trajectory infers: tracing user. DCA solves this problem as the user would be in a different location, which the attacker cannot trace.

(iii) Historical (temporal) attack: it happens when attacker accesses and analyses a lot of user queries. In DCA, this would not happen because a user would always submit other user's query to LBS.

(iv) Inversion attack: this happens when the attacker knows user's algorithm or technique of protection. In DCA, again this would not happen because, even the attacker knows the used technique, they would not know the real query of the user.

Other types of attacks relate to skills and knowledge of attackers about different kinds of information of cells, as follows:

(i) Diversity level: kinds of people of interest (POIs).

(ii) Closeness level (uniformity): cells have been adjacent to each other.

(iii) Congestion: number of users are close to be the same in each cell.

(iv) Location homogeneity: all POIs in the cell have same location (healthy buildings).

(v) Knowledge about the map: an attacker may be skilled to determine the type of area or crop some parts of it to make the area smaller to detect more accurate information about user's location.

In DCA, the possibilities of these can be diminished by having unequal sizes of cells. In addition, we propose that the users should change their alias when they enter a new cell.

4.1. Matrices. Here we discuss and compare privacy and performance metrics of other approaches [5, 35, 42, 47] with DCA.

4.1.1. Privacy Metrics. Many of these metrics are used to measure the system efficiency and compare between approaches. Here we discuss only the critical ones.

K-Anonymity. It refers to number of dummies which are sent to LBS server with real queries. If the number of dummies, k, equals to nine, the user would send ten queries to server (the user should keep the value of k to the minimum). We know that

$$\text{K-Anonymity} = \frac{1}{(1 + k)} \tag{7}$$

Entropy (E). It is the most important measure which refers to privacy level and quantifies the anonymity and the amount of data that LBS server has from each user. The user should aim to have it close to 1. As in (6), it is defined as

$$E = \sum pi * \log_2 (pi) \tag{8}$$

Ubiquity (U). It refers to user's existence at each point in the cell to deny attackers to detect identity. It is easily achieved in DCA because of query swapping mechanism. Actually, U is used to measure the movements of users and probability of their existent in a specific location in a cell and is defined as

$$U = 2^E \tag{9}$$

Uncertainty/Estimate Error (EE). This metric is related to the server to measure the amount of error in trying to determine the position of users. In DCA, EE will be maximum because server provider would not have any clue of the user location (only false information). It is defined as

$$EE = (E)\ 100\% \tag{10}$$

The Entropy of Privacy is related to the amount of right information in the LBS server about user A. Therefore, if the amount of right information in LBS server equals 0, that means the Entropy will be maximum (E = 1) and it leads to maximum uncertainty (EE = 100%).

4.1.2. Performance Metrics. The performance is dependent on factors like number of dummies, algorithm of generating dummies or obfuscation, encryption, cache management, number of queries sent to LBS server, and cost. In DCA, cache concept, smart dummies, and cooperation are used, to measure the performance as follows.

Cache Hit-Ratio (CHR). We use cache hit ratio to measure the percentage of the queries answered by the cache, which depends on the number of connections to LBS server and is defined as

$$\text{CHR} = \frac{(\text{number of queries answered by C1})}{(\text{total number of queries})} \tag{11}$$

Response Time (RT). It is related to the requested time of all operations in the system. It is defined as

$$RT = 2 * ST + PT \tag{12}$$

where ST is the time of sending and returning a query and PT is the processing time.

5. Simulation and Results

This section contains simulation of implementation of main features of DCA that draws a comparison with P2PCache [23] and other methods, namely, Enhanced-CaDSA, Enhanced-DLS, and MoCrowd [36, 38]. For this, we have used MATLAB 2015. We propose to divide the area into 100×100 cells with 10,000 virtual users. Out of the par of caches, first ($cache_1$) was dedicated for saving the answers of queries with POI included in cells with wireless connections and the second ($cache_2$) to store queries and manage the swapping among peers and by avoiding direct cooperation.

As mentioned earlier, the format of any query in LBS applications is {⟨Latitude, Longitude⟩, POI/TYPE, USER-ID} where POI represent the type of user's query. Enhanced approach changed the location part to cell-number and added time-stamp instead of USER-ID for cache freshness

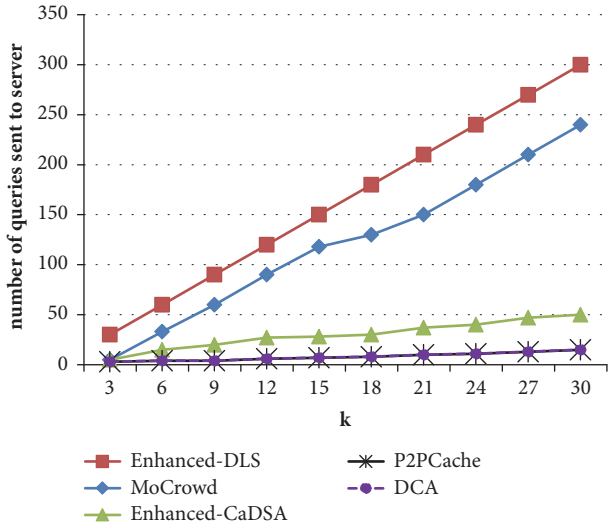

Figure 5: Cost communication with LBS server.

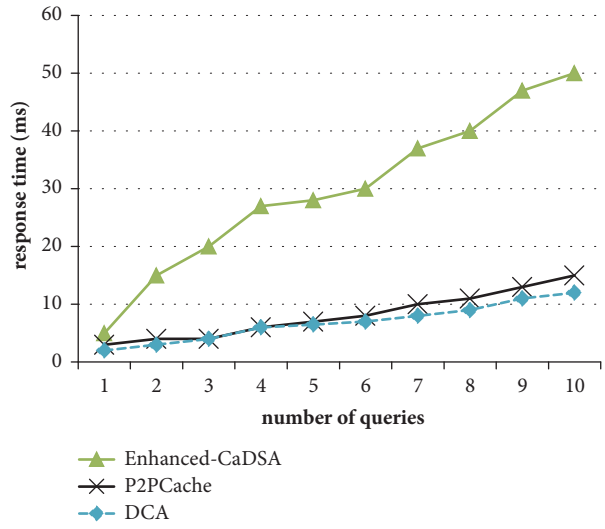

Figure 6: Response time.

operation. Therefore, each query needed less than 1 KB of the cache and after 4 hours the storage cost was 53.7 KB [38]. Consequently, the total size of cache in each cell was 100 KB, which was sufficient. The count of POIs from Google-API for New-York city was around 250,000, requiring 250 MB storage. Finally, either 3G or 4G Wi-Fi connection in the smart cities environment or in specific areas and infrastructure would be adequate.

In DCA, same query format was used as in P2PCache but removed the part of the USER-ID and Time-Stamp {⟨Latitude, Longitude⟩, POI} to make it cost less. However, we added another cache ($cache_2$) to facilitate query swapping.

We now provide examples to prove our claim of supremacy of DCA over earlier available methods. Note that the connection between peers would rely on the mobile app and Wi-Fi, despite having $Cache_2$.

5.1. Cost Communication. It is a measure of the number of connections in LBS server and amount of data on the link, which in our approach, as shown in Figure 5, has better performance than any other methods described in [38]. Notice that DCA and P2PCache have used the minimum and same number of queries sent to LBS server; however the difference lies in the management of cooperation between peers.

5.2. Response Time. DCA overcame the anomalies of P2PCache [23] and Enhanced-CaDSA discussed in [38], and, with the help of Bloom filter, saved time in case of miss-hit in the $cache_2$. Furthermore, the swapping technique did not need searching for or direct dealing with peers, resulting in improvement of performance and enhancement of privacy, as can be seen in Figure 6.

5.3. Cache Hit Ratio. This metric can be improved by storing the queries in $Cache_2$, which may later be requested by other users. As shown in Figure 7, DCA and P2PCache achieved

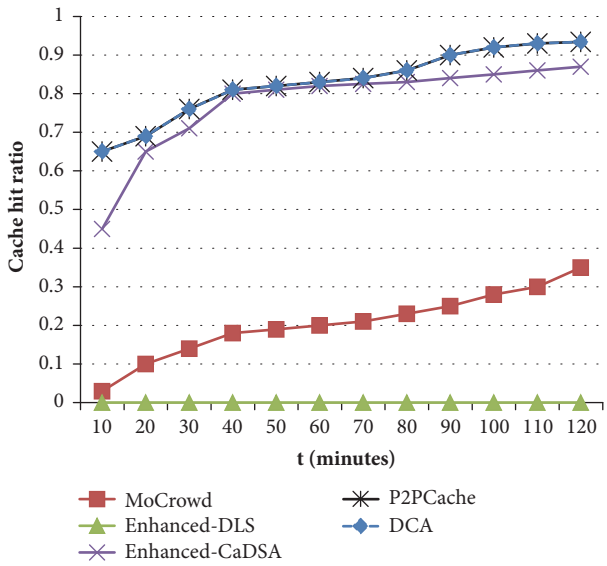

Figure 7: Cache hit-ratio.

the best result because we submitted real queries in $Cache_2$ instead of dummies and better management of cache as explained earlier.

5.4. Privacy Metric (Entropy). Other methods, as in [38], have used dummies (k = 10, 20 or 30) along with the real query to send to LBS to increase the privacy and to maximize this metric. However, it would reduce performance and cache hit-ratio. In DCA, user would just send only one query and the same would be done by the other user, resulting in enhancing the performance and cache hit ration, in addition to maximizing Entropy Value to 1 as shown in Figure 8.

5.5. Amount of Data Collected by the LBS Server. Like Entropy, it relates to the amount of information that the server can obtain for each user. When k increases, the ratio

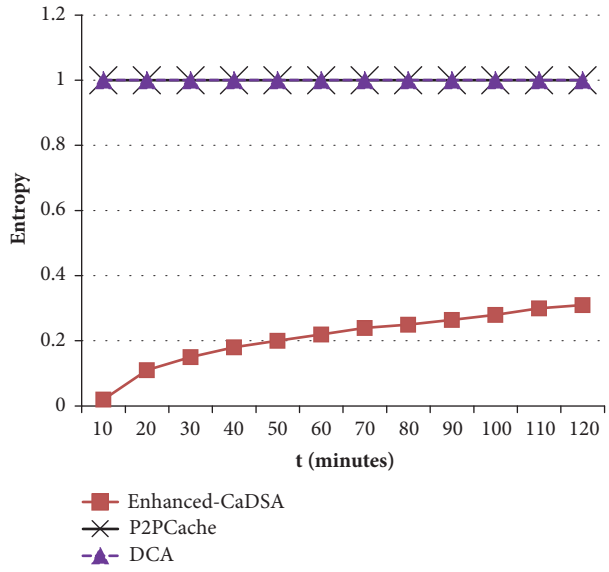

Figure 8: Entropy metrics.

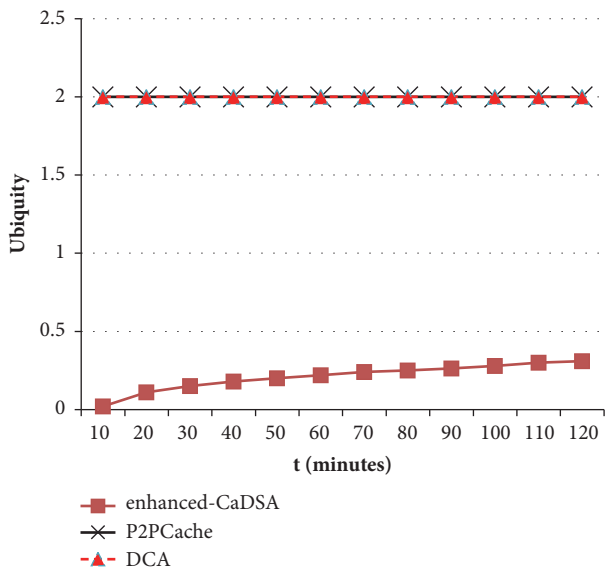

Figure 9: Ubiquity metrics.

would decrease because the server cannot distinguish the actual query from dummies. Thus, DCA would perform better than other approaches because the number of queries is only one, information obtained by server is zero, and so is the ratio.

$$\text{Information Ratio in LBS Server} = \left(\frac{\text{Number of Actual Queries}}{1} + K\right) \quad (13)$$

5.6. Amount of Data Collected by the LBS Server. It refers to degree of spread of a user in a cell and so if it increases, so does privacy because LBS server would not be able to determine the real location of user. Ubiquity (Figure 9) in

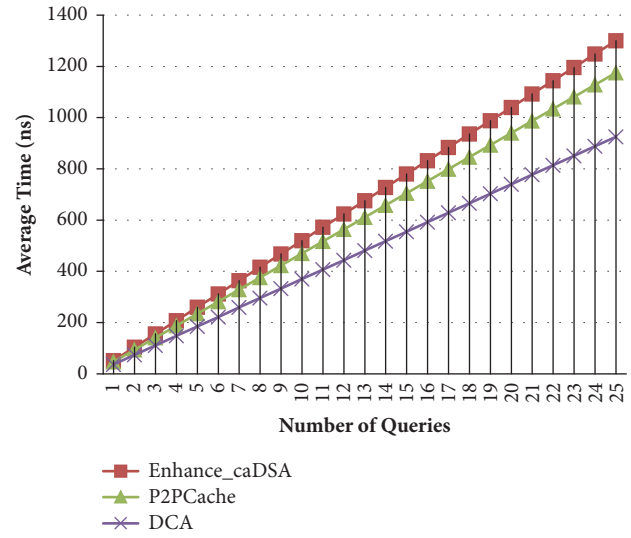

Figure 10: Search performance of cache.

DCA is the best as compared with other methods because it has best Entropy and the swapping technique in $Cache_2$, which amounts to better spread of data in the whole cell.

5.7. Performance of Searching in Cache. As we discussed before, the Enhanced-CaDSA approach [38] uses freshness-time method to manage the content of cache and refresh the time-stamp of the remaining duration of each query in the cache. In the case of DCA, we achieve better results by updating the position of higher order queries and removing the least requested ones in $Cache_1$. In the simulation shown in Figure 10, we have used the hit-ratio, Tc-hit, the response time in the hit case, TC-mis, the response time in the missed case, Tl, the response time from LBS server, Ct, and the response time of collaboration among peers. Using Bloom filter, we get TC-mis ≈ 0, and, for $Cache_2$, we get Ct ≈ 0. Also Avg_enhance = Tc-hit $*$ H + (1-H) (Tl+Tc-miss), Avg_Previous = Tc-hit $*$ H + (1-H)(Tl) + Ct, and Avg_DoubleC = Tc-hit $*$ H + (1-H)(Tl)

5.8. Comparing between Cache Management Methods. In earlier methods of freshness [38], for each stored query time, there are N write operations on cache, where N is the count of current stored queries in cache. When N is changed, so is the ratio of management time. In the hit case of cache, there is just one read operation, and in the miss case there are N read operations. In the miss case of DCA, there is just one read operation, and in the hit case there is one write and one read operation. There is no dealing with freshness here; instead, after each hit or miss, there is one write operation to change the order of selected query. From Figure 11, we can note that the time of management will be static in DCA, whereas it would increase adversely with increase in the time period of frequency freshness.

5.9. Limitations. For performance, trust, and privacy, DCA approach would be better. However, there are some drawbacks:

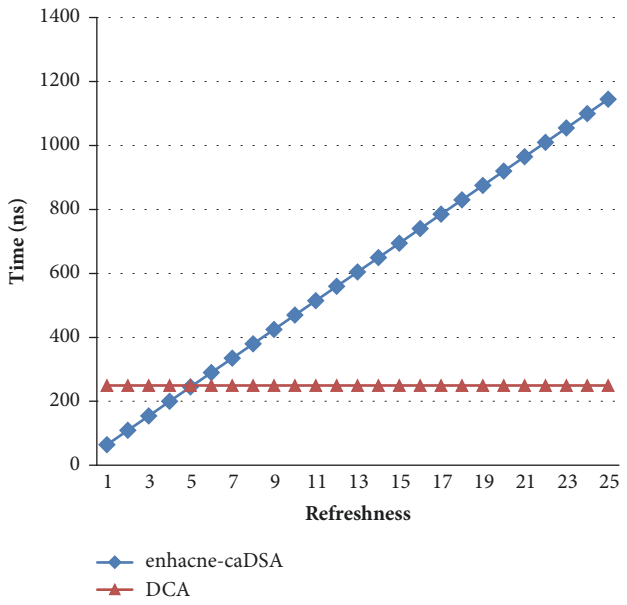

Figure 11: Cache management.

(a) Cost of distribution of the caches by dividing at the access points cells in the physical environment would be higher. Therefore, we have proposed that this issue should be dealt with smart city, which the governments or significantly large organisations can afford by investing the resources required.

(b) Cache needs to be protected from eavesdropper or hacker.

(c) DCA would work well with LBS but there are other privacy issues in healthcare domain which we shall address in future.

6. Conclusions

In this paper, we proposed DCA as a novel approach for preserving privacy and security in LBS of IoT applications in general and the medical field in particular. It is the first technique which uses the idea of two caches, addresses the trust among peers, and manages the relations between them in wireless environment. Simulation results have shown overcoming superiority of this approach over previously known ones dealing with the privacy, cost of communication, and performance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H. Chen, A. Nicogossian, S. Olsson et al., "Electronic health," *International journal of telemedicine and applications*, vol. 2009, Article ID 308710, 2 pages, 2009.

[2] C. Stingl and D. Slamanig, "Health records and the cloud computing paradigm from a privacy perspective," *Journal of Healthcare Engineering*, vol. 2, no. 4, pp. 487–508, 2011.

[3] Q. Huang, L. Wang, and Y. Yang, "Secure and Privacy-Preserving Data Sharing and Collaboration in Mobile Healthcare Social Networks of Smart Cities," *Security and Communication Networks*, vol. 2017, Article ID 6426495, 12 pages, 2017.

[4] W. Sun, C. Chen, B. Zheng, C. Chen, and P. Liu, "An air index for spatial query processing in road networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 382–395, 2015.

[5] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[6] D. Gavalas, P. Nicopolitidis, A. Kameas et al., "Smart Cities: Recent Trends, Methodologies, and Applications," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 7090963, 2017.

[7] S. Guan, Y. Zhang, and Y. Ji, "Privacy-preserving health data collection for preschool children," *Computational and mathematical methods in medicine*, vol. 2013, Article ID 501607, 5 pages, 2013.

[8] S. I. Konomi and C. S. Nam, "Supporting collaborative privacy-observant information sharing using RFID-tagged objects," *Advances in Human-Computer Interaction*, vol. 2009, 5 pages, 2009.

[9] R. S. Zuberi and S. N. Ahmad, "Secure mix-zones for privacy protection of road network location based services users," *Journal of Computer Networks and Communications*, vol. 2016, Article ID 3821593, 8 pages, 2016.

[10] J. Wang and Z. Wang, "A survey on personal data cloud," *The Scientific World Journal*, vol. 2014, Article ID 969150, 13 pages, 2014.

[11] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, Article ID 7576307, 12 pages, 2017.

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

[13] M. S. Alrahal, M. U. Ashraf, and A. Abesen, "AES-Route Server Model for Location based Services in," in *Proceedings of the AES-Route Server Model for Location based Services in Road Networks. International Journal Of Advanced Computer Science And Applications*, vol. 8, pp. 361–368, 2017.

[14] A. Ukil, S. Bandyopadhyay, and A. Pal, "IoT-privacy: To be private or not to be private," in *Proceedings of the In Communications Workshops (INFOCOM WKSHPS, '14)*, pp. 123-124, 2014.

[15] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[16] A. B. Labrique, G. D. Kirk, R. P. Westergaard, and M. W. Merritt, "Ethical issues in mHealth research involving persons living with HIV/AIDS and substance abuse," *AIDS research and treatment*, vol. 2013, Article ID 189645, 6 pages, 2013.

[17] A. Fragopoulos, J. Gialelis, and D. Serpanos, "Security framework for pervasive healthcare architectures utilizing," *International journal of telemedicine and applications*, vol. 2009, Article ID 461560, 9 pages, 2009.

[18] J. M. Shin, "Secure Remote Health Monitoring with Unreliable Mobile Devices," *Journal of Biomedicine and Biotechnology*, vol. 2012, Article ID 546021, 5 pages, 2012.

[19] J. Bou Abdo, T. Bourgeau, J. Demerjian, and H. Chaouchi, "Extended privacy in crowdsourced location-based services using mobile cloud computing," *Mobile Information Systems*, vol. 2016, Article ID 7867206, 13 pages, 2016.

[20] J. Zhong, W. Wu, C. Cao, and W. Feng, "A Variable Weight Privacy-Preserving Algorithm for the Mobile Crowd Sensing Network," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 3053202, 7 pages, 2017.

[21] B. Kang, J. Wang, and D. Shao, "Attack on Privacy-Preserving Public Auditing Schemes for Cloud Storage," *Problems in Engineering*, Article ID 8062182, 6 pages, 2017.

[22] A. A. A. Sen, F. A. Eassa, K. Jambi, and M. Yamin, "Preserving privacy in internet of things: a survey," *International Journal of Information Technology*, pp. 1–12, 2018.

[23] M. Yamin and A. A. A. Sen, "Improving Privacy and Security of User Data in Location Based Services," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 9, no. 1, pp. 19–42, 2018.

[24] O. Vernesan, P. Friess, G. Woysch et al., *Europes IoT strategic research agenda, 2012., The Internet of Things*, UK, Halifax, 2012.

[25] A. AbiSen, F. Albouraey, and K. A. Jambi, "A Survey of FOG Computing: Properties, Roles and Challenges," *Computing for Sustainable Global Development (BVICAM)*.

[26] I. Y. Jung, G. J. Jang, and S. J. Kang, "Secure eHealth-care service on self-organizing software platform," *Mathematical Problems in Engineering*, vol. 2014, Article ID 350876, 9 pages, 2014.

[27] Z. Lin, X. Xiao, Y. Sun, Y. Zhang, and Y. Ma, "A Privacy-Preserving Intelligent Medical Diagnosis System Based on Oblivious Keyword Search," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8632183, 7 pages, 2017.

[28] L. Sun, M. Yamin, C. Mushi, K. Liu, M. Alsaigh, and F. Chen, "Information analytics for healthcare service discovery," *Journal of healthcare engineering*, vol. 5, no. 4, pp. 457–478, 2014.

[29] A. A. Sen, F. Eassa, K. Jambi et al., "Preserving privacy in internet of things: a survey," *International Journal of Information*, vol. 10, no. 2, pp. 189–200, 2018.

[30] M. Mokbel and C. Chow, "Challenges in Preserving Location Privacy in Peer-to-Peer Environments," in *Proceedings of the Seventh International Conference on Web-Age Information Management Workshops*, pp. 1-1, Hong Kong, China, June 2006.

[31] A. Solanas, J. Domingo-Ferrer, and A. Martínez-Ballesté, "Location privacy in location-based services: Beyond TTP-based schemes," in *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, PiLBA 2008*, pp. 12–23, Spain, October 2008.

[32] D. Miorandi, S. Sicari, F. de Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[33] Y. Chen, J. Bao, W. Ku, and J. Huang, "Cache Management Techniques for Privacy Preserving Location-based Services," in *Proceedings of the Ninth International Conference on Mobile Data Management Workshops, MDMW*, pp. 88–96, Beijing, China, April 2008.

[34] J. Cacheclock Meyerowitz and R. Roy Choudhury, "Hiding stars with fireworks: location privacy through camouflage," in *Proceedings of the In Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 345–356, 2009.

[35] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *Wireless Communications*, vol. 19, no. 1, 2012.

[36] X. Zhu, H. Chi, B. Niu, W. Zhang, Z. Li, and H. Li, "Mobicache, When k-anonymity meets cache," in *Proceedings of the Global Communications Conference (GLOBECOM, '13)*, pp. 820–825, 2013.

[37] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the Proceedings IEEE*, pp. 754–762, 2014.

[38] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proceedings of the IEEE Conference Computer Communications (INFOCOM '15)*, pp. 1017–1025, 2015.

[39] Y. Sun, M. Chen, L. Hu, Y. Qian, and M. M. Hassan, "ASA, Against statistical attacks for privacy-aware users in Location Based," *Service. Generation Computer Systems*, vol. 70, pp. 48–58, 2017.

[40] C. Ma, L. Zhang, S. Yang, and X. Zheng, "Hiding Yourself Behind Collaborative Users When Using Continuous Location-Based Services. of Circuits," *Systems and Computers*, vol. 26, no. 07, Article ID 1750119, p. 10, 2017.

[41] J. Domingo-Ferrer, M. Bras-Amorós, Q. Wu, and J. Manjón, "User-private information retrieval based on a peer-to-peer community," *Knowledge Engineering*, vol. 68, no. 11, pp. 1237–1252, 2009.

[42] P. Jagwani and S. Kaushik, "Privacy in Location Based Services: Protection Strategies, Attack Models and Open Challenges," in *Proceedings of the International Conference on Information Science and Applications*, pp. 12–21, Singapore, 2017.

[43] Y. L. Chen, B. C. Cheng, H. L. Chen et al., "A privacy-preserved analytical method for eHealth database with minimized information loss," *BioMed Research International*, Article ID 521267, 9 pages, 2012.

[44] S. S. Kim, Y. H. Lee, J. M. Kim, D. S. Seo, G. H. Kim, and Y. S. Shin, "Privacy protection for personal health device communication and healthcare building applications," *Journal of Applied Mathematics*, vol. 2014, Article ID 462453, 5 pages, 2014.

[45] P. Zhang, M. Durresi, and A. Durresi, "Enhanced Internet Mobility and Privacy Using Public Cloud," *Mobile Information Systems*, vol. 2017, Article ID 4725858, 11 pages, 2017.

[46] J. Qu, G. Zhang, and Z. Fang, "Prophet: A Context-Aware Location Privacy-Preserving Scheme in Location Sharing Service," *Discrete Dynamics in Nature and Society*, vol. 2017, 7 pages, 2017.

[47] M. S. Alrahhal, M. Khemakhem, and K. Jambi, "A survey on privacy of location-based services: classification, inference attacks, and challenges," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 24, 2017.