Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SPATIAL CLOAKING FOR LOCATION PRIVACY PROTECTION OF SMART HEALTH CARE SYSTEMS IN FOG COMPUTING

MAHMOUD ASASSFEH [1], WESAM ALMOBAIDEEN[1,2], NADIM OBEID[3]

[1] Department of Computer Science-King Abdullah II School for information technology, University of Jordan, Amman-Jordan.
[2] Rochester Institute of Technology in Dubai, Dubai, UAE.
[2]Department of Computer Information Systems-King Abdullah II School for information technology, University of Jordan, Amman-Jordan.
Email: habashneh_m@yahoo.com, Wesam.almobaideen@gmail.com, nadim@ju.edu.jo

ABSTRACT

Geographic information like location data is essential for a smart health care system. Patient information data is periodically sent to hospitals or medical centers to improve healthcare services presented to patients. The location data with a timestamp can disclose the user's private data like user financial activity, medical status, lifestyle, and places frequently visited by the user. Privacy protection approaches include spatial cloaking that is used to conceal the location of the user, into a cloaking area that satisfies the user privacy requirement when using the location to get healthcare services, or when using location-based services (LBS) to get any other services. Spatial cloaking is used in many location privacy solutions, however, most of them have some disadvantages that are related to communication and computation costs. In this paper an effective spatial cloaking algorithm to preserve location privacy (LOCACY) is presented. A secure version of the A* heuristic search algorithm (SecA*) has been developed to perform two essential functions: the first is to better support the proposed spatial cloaking algorithm, and the second is to enable a mobile patient to avoid infected areas while traveling between various locations. The proposed spatial cloaking algorithm outperforms rival algorithms such as bottom-up, top-down, and Aman algorithms in terms of communication and computation costs and achieves average enhancement of 56% better than the recently proposed Aman algorithm. Evaluating the secure A* algorithm shows that it provides a safe path and improves the provision of privacy.

Keywords: *Smart Healthcare Systems, Fog Computing, Location Privacy, Spatial Cloaking.*

1. INTRODUCTION

With the ongoing advancement of location tracking such as GPS, more and more applications in wireless networks have taken advantage of location information of wireless users and devices in their design and development. However, revealing location information will raise security and privacy concerns [1, 2, 3].

There is a need to send location data to hospital for the following reasons:

- The time stamp along together with the patient's location will help with the detection and treatment of viruses that might have infected the patient when traveling through infected areas.
- Identifying a patient's location in the event of an emergency will help healthcare professionals and hospitals get an ambulance to the patient's location immediately.
- Sending a person's location, while moving to the hospital helps in getting a warning throughout a mobile device whether that person comes into proximity to either of these contaminated areas.

Nowadays, it is possible to mount a GPS sensor on a patient due to the production of compact and low cost (GPS) systems [2]. The patient's location information can be submitted to the hospitals regularly, which may allow an intruder to collect a large volume of possibly confidential information by analyzing the location data [4, 5]. Knowing physical locations, such as health facilities, can reveal private person's health issues. Similarly, frequent visits to such types of locations may be attributed to one's lifestyles. By learning a patient's location data for a duration of time, the relevant data may be accessed [6]. As a result, protecting a patient's location privacy is critical since an intruder can deduce some of the user's habits and interests by tracking the locations accessed by that user.

Another need to transmit location data arises, when services are requested from location-based services (LBS), which is a type of service that provides information to a mobile user according to the location of that user and allows access to application servers such as transportation, healthcare, and entertainment servers [7,8, 9].

Because LBS servers are not trusted, submitting precise location data runs the risk of compromising user location privacy [10, 11]. Moreover, an adversary may deduce private mobile user data based on location information queries [11, 12].

Multiple approaches have been suggested in order to provide location privacy and prevent intruders from exposing the location of a mobile user while communicating with hospitals or LSB. Examples of such approaches are spatial cloaking, location transformation, dummies and cryptography [13].

One of the influential techniques of implementing spatial cloaking is K-anonymity which is used to blur the client location among K number of other users' locations [16]. K-anonymity requires two parameters, K and A-min. Where K indicates the number of users located in the cloaked area, and A-min indicates the minimum scale of the cloaked area comprising K users [12, 14]. The cloaked area is formed either on client side or on trusted third-party [17]. One specific challenge of this method is that it needs an adequate number of users within the same geographical location, or a path from the source location to the destination, to maintain anonymity while the user is moving [18, 19].

The geospatial information has rich data about temporal, spatial distributions as well as mobile users' distributions inside a certain region, which are given by density servers stationed in the cloud. This conventional configuration increases the latency when requesting information from density servers [20]. Since we need this information to support healthcare systems, a little delay can cost a patient's life, therefore, to enhance services and applications, the density servers should be setup in fog layer [21]. which is a layer in between a traditional gateway and a remote cloud server. Fog layer helps save network bandwidth, increment throughput and decreases latency near the edge of the geo-spatial users [22].

Requesting and responding to how many users in the same geographical location from density servers is done through communication round, between the mobile device, and the density server, so the less the communication rounds the better the efficiency of the solution. Computation cost, which is the time from the query issuance till receiving the results in the user's device and computing of spatial area, is also depending on the communication round, since the algorithm is executed in each round, these challenges should be tackled to enhance the performance of any suggested solution [10, 23].

In this paper we present an effective spatial cloaking algorithm that minimizes the computation cost to calculate the cloaked area and reduces the communication rounds between a mobile device and a fog density server. It also minimizes latency since it uses fog computing instead of cloud-computing. We compared the proposed algorithm performance with some rival algorithms such as bottom-up, top-down and Aman algorithms [15, 9, 10] in terms of communication and computation costs. Moreover, we have developed a secure version of the A* heuristic search algorithm to provide two main functionalities, the first is to better support the proposed spatial cloaking algorithm, and the second is to allow a mobile patient to avoid infected areas while moving between places.

From above discussion our research contributions are as follows:

- Propose a new algorithm to compute spatial cloaking area that named "LOCACY". This algorithm should minimize the computation and communication cost, which makes it suitable to be executed in mobile devices to provide location privacy.
- Design a new architecture for spatial cloaking that uses fog system computing instead of cloud computing to decrease the communication latency.
- Propose SecA* heuristic search algorithm that provides spatial clocking through LOCACY by choosing a safe path for users.

The rest of the paper is organized as follows. Section 2 highlights related work. The description of the developed spatial cloaking algorithm is given in Section 3. Section 4 discusses the performance evaluation of the presented algorithm in comparison with other related rival algorithms. In Section 5 we present a heuristic search algorithm used to select a safe path for mobile patients. Section 6 concludes this paper.

2. RELATED WORK

To preserve location privacy, there have been many techniques introduced in the literature to protect people's location privacy in different application scenarios [24]. Most of the location privacy protection techniques fall into two categories:

anonymity-based method and obfuscation-based method [24].

An anonymity-based method is also named as spatial cloaking technology, among which k-anonymity method is the most well-known. By employing the quad tree data structure. The k-anonymity method can guarantee that a cloaking area of one user contains at least $k-1$ other users. Thus, k users in the same area are indistinguishable from each other. The probability of separating or recognizing each individual is reduced to 1/k in this manner [12, 32]. Most of the approaches related to spatial cloaking need, of course, the intervention of a trusted third party that acts as an anonymity server, which is the weakness part of these approaches because it is not trusted.

On the other hand, the obfuscation-based method protects location privacy by producing a fake user location or by separating locations from identities, Spatial obfuscation approaches preserve privacy by minimizing the accuracy of location data transmitted from the user to the LBS, and this can be achieved at the user's site without the involvement of a trusted third party, which is a significant benefit of this class of approaches over the spatial cloaking that require trusted third party to function as an anonymity server [25].

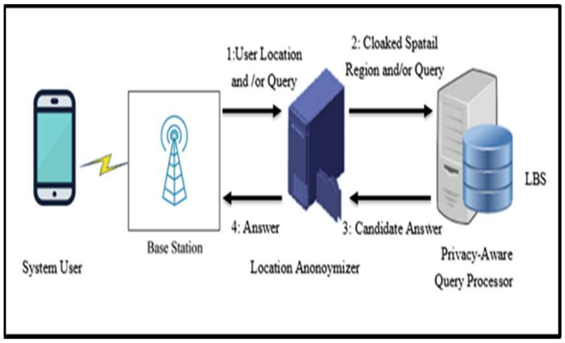


Fig.1 The Architecture Of Aman System

However, obfuscation-based method is not very robust because it is subject to triangulation attacks. A user sending two consecutive signals from different zones would reveal that she/he is close to the border between them, and three consecutive signals from various zones would disclose her/his location quite accurately [25].

In this section we highlight some of prominent researches that address location privacy.

In paper [10], the authors propose in device k-anonymity cloaking technique called Aman, the architecture of the system composed of the cloud server, LBS server, and the user, as shown in Figure 1.

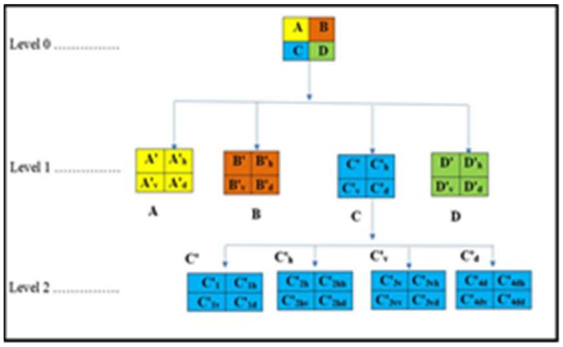


Fig. 2 The Quad Tree Data Structure

They proposed Aman algorithm that searches, and looks for a cloaked area in the structure, and composition of quad-tree indexing very effectively. The quest begins at an intermediate approximate level in quad-tree, which is close to the target cloaked area. Figure2. Describe the data structure of the quad tree.

In this approach, the user sends a request to the density cloud server to access the density data (user distribution) of the calculated level. As soon as the data is received, the cloaked area is computed in the device by the user, and then send the service request containing the cloaked area to LBS. LBS server executes the request and forwards the answer to the user, who filters the response to get the result [10].

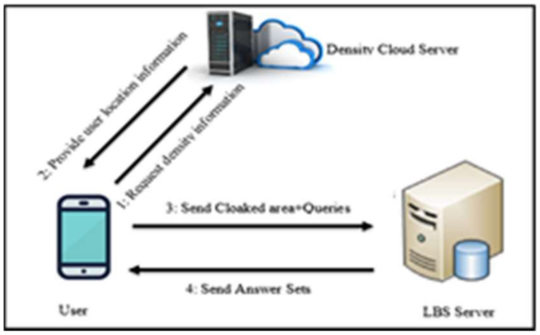

Density Cloud Server
2: Provide user location information
1: Request density information
3: Send Cloaked area+Queries
4: Send Answer Sets
User
LBS Server

Fig. 3 Casper Architecture

In paper [15] the authors propose an approach that consists of mobile user, location anonymizer, privacy-aware query processor as shown in Figure 3. Location anonymizer collects the mobile users' location and updates continuously [26]. Afterwards it masks the users' location in the cloaked area containing (Amin, K), and sends it to the LBS. The privacy aware query processor is integrated with LBS to deal with

cloaked area, rather than specific locations, and it returns a set of answers to the user through location anonymizer, Figure 3 depicts the scheme architecture which is called Casper.

The cloaking algorithm uses bottom-up search, it begins looking from the leaf of quad tree, afterward, going up the pyramid till the user's cell is fulfilled

(K, A-min), which is considered as a spatial cloaked area of the client [27].
In [12] the author proposed Peer to Peer (P2P) spatial cloaking algorithm, which consist of two components: mobile users, and location-based server, each user has a privacy profile that consist of two parameters K and A-min Figure 4 depict peer to peer architecture.

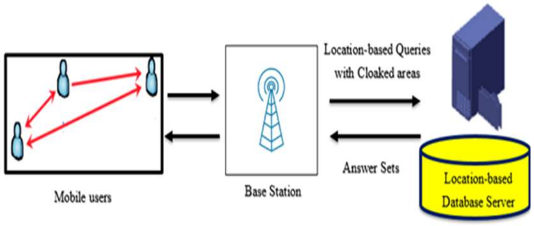


Fig. 4 Peer To Peer Architecture

In this architecture, the mobile device has two wireless networks one specified to contact location-based database servers (LBS), and the other one is to contact other users. The idea of their P2P spatial cloaking is that mobile users contact other users to find at least K-1 peers, the user calculates the cloaked area that includes the mobile user and K-1 closest users. The user expands the cloaked area to be at least A-min, then the user sends a request with the cloaked area as his blurred location to LBS, which receives the request and processes it using a privacy-aware query processor that returns a set of results to the mobile user.
In [6] the authors presented a location privacy protection technique in which location privacy is preserved, while keeping the use of the location data. In the proposed technique, the main processing unit (MPU) connected to a patient’s body produces the perturbed location, by considering the distance between the patient’s location and the pre-defined patient’s sensitive locations, in this technique no need to trust other parties while preserving the privacy. Figure 5 illustrates this mechanism.

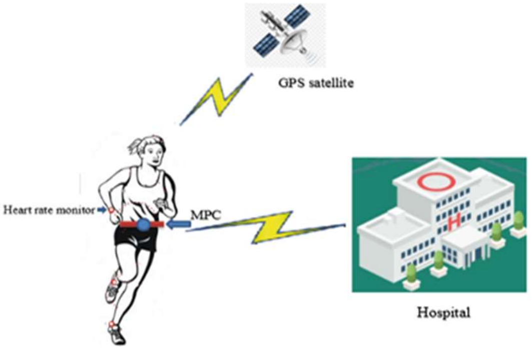


Fig. 5 The Smart Health System.

From previous related work the approach presented in [10]. Allows several changes in the creation of the cloaked area, by computing cloaked area in user device after collecting information from the density cloud server, beside that it eliminate the usage of central trusted anonymizer, which is
expensive and complex, on the other hand this scheme has high communication and computing costs relative to centralized approaches that use location anonymizer, in addition to the latency of this method due to cloud server reliance. Moreover, the authors use equation Li= log4 U/K to start looking at the intermediate level in the quad tree to minimize the communication rounds, where U is the Complete number of users, and K is the number of users Stationed in the cloaked area, this equation is generally considered correct in case of the uniform distribution of online users in the space, however, this assumption is not accurate in real life. Since online users are randomly distributed and altered all the time, this equation's output is not reliable and not consistent. In [15] the proposed scheme (Casper) has the following advantages:

- Efficient in term of request processing time and computation of cloaked area time
- Scalable in terms of supporting a lot of users.

The drawbacks of this scheme appeared in location anonymizer, which is costly and sophisticated and not all the time trusted, besides that it knows a lot about the users.
In [12] the proposed scheme minimizes the communication overhead, and decreases the effect of a network partition, besides that the system addresses the center of cloaked area privacy attack, by using cloaked area adjustment scheme.
The limitations of this scheme are:

- Limitation in transmission range and constraint on communication resources.
- This technique assumes all users or (peers) are trusted, this claim might be wrong.
- There is a compromise between privacy and accuracy.

In [6] the proposed scheme didn’t use trusted third parties, and they reduced the communication overhead, but the computation overhead in the main processing unit (MPU) is high compared to other mechanisms. Table 1. Show the comparison between these schemes.

Table 1 The Comparison Between SCHEMES.

Paper	Architecture	Major strength	Major weakness
Hiba Jadallah et al [10]	Decentralized	1.Stronger privacy support. 2. did not use location anonymizer which is costly.	1. The communication and computation cost is still high after optimization. 2. high latency due to cloud computing utilization. 3. The algorithm is not efficient when the distribution of users in the space is not uniform.
Mokbel et al [15]	Centralized (using location anonymizer)	1. Efficient in executing query. 2. Scalable.	1. Trusting a third party represented by location anonymizer which is costly. 2. Privacy leaks.
Chi-yin chow et al [12]	Decentralized	1.Reduce communication overhead. 2. Solving the privacy assault at the core of the cloaked area.	1. Limitation in transmission range and communication resources. 2. Trade-off between privacy and accuracy. 3. Assuming all users are trusted which might be wrong.
Natgunanathan et al [6]	Decentralized (main processing unit (MPU) attached to a patient's body)	1. No need to trust other parties while preserving privacy. 2. Reduce communication overhead.	1. computation overhead in MPU

We can summarize the goal of algorithms that we have found in the literature as to preserve the location privacy of the user or patient who uses location-based services (LBS). An adversary may access private data of users based on their location information queries where an intruder can specify the user habits and interests. Accordingly, the location of the user must be hidden all the time while he is on the move. In this paper the proposed scheme, i.e. LOCACY, overcomes some of the disadvantages of methods mentioned in literature review such as requiring high computation and communication costs, which cannot be supported by constraint devices in addition to reducing the latency due to the use of servers resides in the cloud. Another challenge facing location privacy methods is the need to have reasonable number of users within the same geographical area to preserve anonymity; this challenge has been tackled in LOCACY to make sure to have an adequate number of users in the same area.

3. PROPOSED SCHEME (LOCACY)

Our proposed scheme (LOCACY) is present in details in this section.

3.1 LOCACY Architecture

The proposed scheme services patients in providing location privacy by sending the patient's location to a medical center as a clocked area that hides the patient's location. Although in case of emergency the client's exact location can be sent as an encrypted field, for other non-emergency cases the client can send a cloaked location in order to achieve location privacy. An example of the later case is when the patient is communicating with the medical center in order to get location based services that are related to healthcare. Additionally, the medical center can compare the patient's location with infected areas stored in the medical center database and give notification to the user of the infected areas in the proximity. LOCACY consists of three layers as follows:

- Fog layer: The density servers are placed in this layer in order to update and index the users' locations in the entire area and provide them to the users as per a request. The use of fog servers reduces the latency in communication.
- Communication layer: Facilitates the communication between various system devices. It allows patients' end devices such as smartphones to communicate with fog density servers, medical servers and other LBS servers via a base station.
- Services layer: It provides services to the patient like medical services, and location-based services through medical servers and other LBS servers. Figure 6 illustrates the components of the architecture.

In our scheme, the cloaked area is calculated in the mobile device, which has a grid structure that decomposes the space recursively into cells stored in the mobile device memory. Each grid cell can include several users, where the number of these users is obtained by requesting it from the fog density server. After receiving the number of users surrounding the mobile device, the mobile client generates a cloaked area that fulfills the user privacy requirements (A-min, K) and sends it to the medical center, or the LBS as a part of the request. The proposed architecture avoids using the location anonymizer, which while it acquires a lot of information about the users, it may not always be trusted. In addition, it is costly.

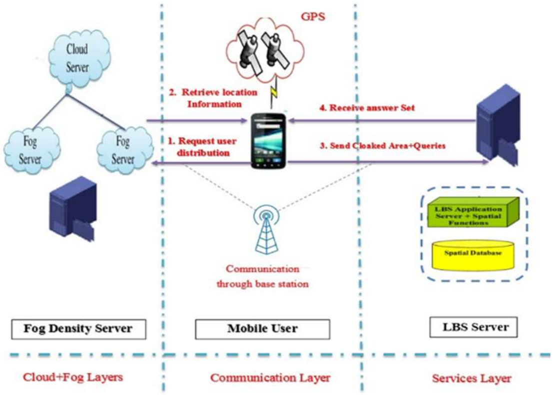

GPS
Cloud Server
Fog Server
Fog Server
2. Retrieve location Information
4. Receive answer Set
1. Request user distribution
3. Send Cloaked Area+Queries
LBS Application Server + Spatial Functions
Spatial Database
Communication through base station
Fog Density Server
Mobile User
LBS Server
Cloud+Fog Layers
Communication Layer
Services Layer

Fig. 6 Proposed System Architecture

3.2 LOCACY Algorithm

To calculate the cloaked area, the algorithm needs two input parameters which are: number of users in the cloaked area called (K), and minimum size of the cloaked area, called (A-min). The density fog server uses a quad tree data structure, which divides the spatial space into h levels, that have 4^h cells. The entire spatial space represents the root of the tree, where each cell has an id and many live users in it. This is the same structure used by Aman algorithm [10] and discussed in the related work section.

Searching the quad tree could be conducted through either bottom-up or top-down approaches. The bottom-up approach begins searching the quad-tree from the leaf, in which the user exists, and goes up until the level that fulfills the k-anonymity is reached. It first finds the cell in which the user exists, then it checks whether the number of users in the cell is more than or equal to K, and the cell size equals to or more than A-min. Satisfying these conditions results in adopting that cell as a cloaked area [15].

In top-down approach. In order to search for the cloaked area, the same logic as the bottom-up search is used, however, this is done in a reverse order. It begins from the root and then moves down until A-min, which satisfies the k-anonymity, is found [9]. The key disadvantage of both approaches in the search for a cloaked area is that several communication rounds are required to locate a cloaked area.

To address the shortcomings of both the bottom-up and the top-down methods, the authors in [10] has suggested an algorithm, named Aman, that efficiently searches for cloaked areas in the quad tree indexing structure. The quest begins at the intermediate approximate level of the quad tree, that is as close to the target area as possible according to the equation (1).

$$Li = Log4\ (\frac{U}{K})\ \ldots\ldots\ldots\ldots\ldots (1)$$

where Li is the estimated level, U is the total number of online users in the entire area, K is the number of users in the cloaked area. Aman algorithm works well when assuming a uniform distribution of users over the considered area. This assumption is not valid in reality since users' distribution follows more realistic models. It is unlikely that people would spread themselves evenly throughout an area. On the other hand, one might expect a Gaussian model in the real world [28]. Therefore, the estimated level in Aman algorithm can be inaccurate and requires many communication rounds to compute the cloaked area.

To tackle the potential drawbacks of the bottom-up, top-down and Aman algorithms used in previous approaches, we present an algorithm, named LOCACY (for LOCation privACY), that starts searching for a cloaked area at a closer intermediate level in the quad-tree to the required cloaked area. This level contains cells with a size equal or a bit more than Amin according to equation (2).

$$Li = log\ 4\ (\frac{A}{A-min})\ \ldots\ldots\ldots\ldots\ldots (2)$$

Equation (2) is inspired by equation (1) used in Aman algorithm [10]. We adjust equation (1) because it is valid only if the online users are evenly distributed in the space. Therefore, we use (A) and (A-min) which represent the entire area and the minimum size of the cloaked area respectively, as inputs instead of U and K. In contrast to Aman algorithm and its dependency on parameters that are affected by the way users are scattered, LOCACY uses A-min and A which do not depend on the distribution of users in the whole space. This means that the proposed equation for LOCACY is valid in all cases of user distribution. LOCACY algorithm, see Algorithm 1, works as follows:

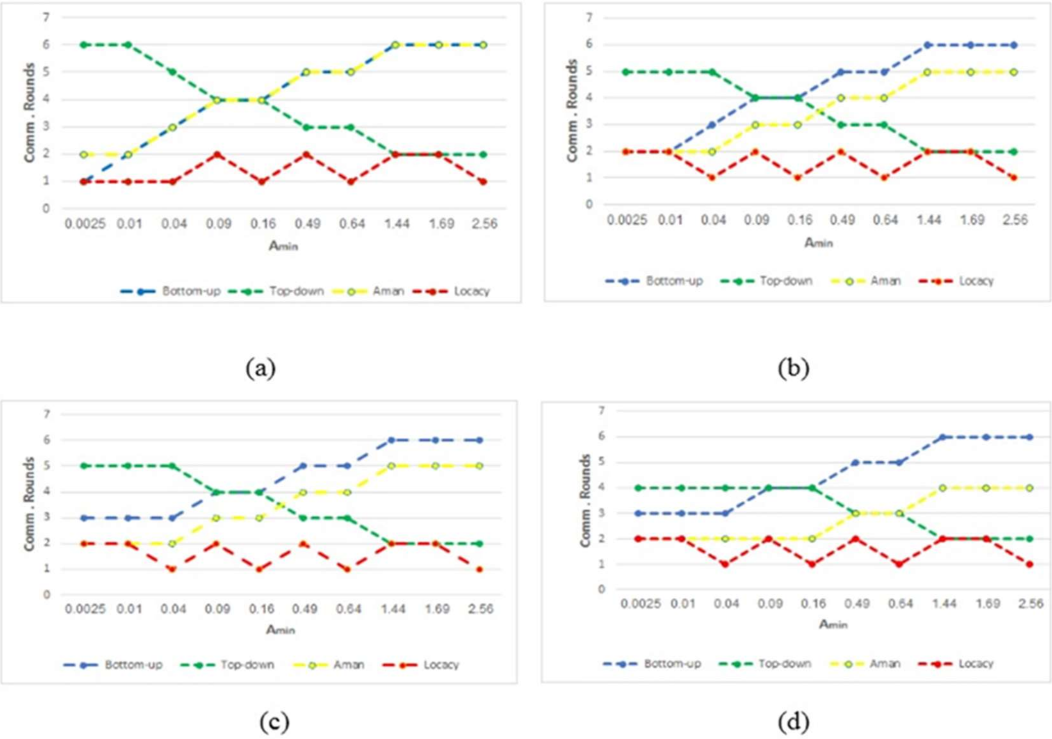
- By using equation (2) and once the approximate level is determined, the client contacts the density fog server and receives

Algorithm 1: Spatial cloaking algorithm for Location Privacy (LOCACY)

```
Input: (A-min), K
Output: cloaked area CA.
Method:
1. Set cloaked area (CA) = root cell.
2. Determine the exact location (x, y) of the user
3. Calculate the estimated level Li using Eq. 4.
4. Determine the Cell (C) where the user exists based on (x, y) location.
5. Get the density data for all cells of the level Li from the fog server.
6. If  C = A-min  and $C_{users} \geq K$  then
7.         Return CA = C                          ( C is the user cell in level Li)
8.  Else If (0.75) C > A-min and $C_{users} > K$
9.          Li =Li + 1
10.             If C ' ≥ A-min and $C'_{users} > K$        (C' is the user cell in level Li+1)
11.             Return CA = C '
12.             Else If C ' < A-min
13.             Check the horizontal neighbor $C_h'$ and vertical neighbor $C_v'$ with the same parent grid
14.             cell.
15.                       If $C' \cup C_h' \geq$ A-min or $C' \cup C_v' \geq$ A-min
16.                                  and If $C'_{users} + C_h'_{users} \geq C'_{users} + C_v'_{users}$ and $C_h'_{users} < K$
17.                                  Return CA = $C' \cup C_{h'}$
18.                                  Else If $Cv'_{users} < K$
19                                   Return CA = C ' ∪ Cv'
20.                                  Else If $C' \cup C_h' \cup C_v' \geq$ A-min and $C'_{users} + C_v'_{users} + C_h'_{users} > K$
21.                                  and $(C_v'_{users} , C_h'_{users}) < K$
22.                                  Return CA =$C' \cup C_h' \cup C_v'$
23.                                  Else
24.                                  Li = Li - 1
25.                                  Return CA= C
26.                                  End If
27.                        End If
28.           End If
29. End If
```

the actual number of existing users in each cell that belong to level Li from the fog density server. (lines 1-3).

- Determine the cell C where the user is located (line 4).
- The algorithm checks the size of cell C against A-min, and the number of users in it against K, if C = A-min and the number of users is higher than K, then the algorithm returns cell C as the cloaked area (lines 6-7).
- Otherwise, if the size of (0.75 C) is greater than A-min and the number of users in C is higher than K, the algorithm steps down one level from Li to Li+1 and then tests the children cells of C to determine the child cell where the user is located. If the size of the child cell equals or higher than A-min, and the number of users in that cell equals or greater than K, then report it as a cloaked area (lines 8-11).
- Otherwise if the number of users is less than K, it tests the combination of child cell C ' in level Li+1 and the horizontal neighbor Ch`, if the size of the combined area is more than A-min, and the number of users in the combine area is more than or equal to K, then record it as a cloaked area. Alternatively, repeat the same test with the vertical neighbor Cv` cell.
- If the active number of users in neighboring cell is not lower than k, we will not be able to merge either of the neighboring cells with the user cell C `, since the intruder can easily discover that the issuer (user) is in cell C ` (lines 12-19).
- If the combination with one of neighboring cells results in less than the required number of users or size of the clocked area, then the algorithm tests the combination of the current user cell C ` and both horizontal and vertical cells. if the size of the combined area is more than A-min and the sum of users in it is more



(a) (b) (c) (d)

Fig. 8 Impact Of A-Min Value On The Number Of Communication Rounds, (A) K=25, (B) K=50, (C) K=100, And (D) K=200.

than or equal to K value, it will be reported as the cloaked area. Otherwise it will move up one level (lines 20-25).

- This method is replicated recursively until an area that satisfies A-min and k users is found.

We use the condition (0.75 C) to be greater than A-min to go down one level because (C in level Li = 4 C ` in level Li+1, so if A-min is less than the combination of the user cell C ` in level Li+1 and horizontal and vertical neighbor cells, i.e. A-min is less than C ' +Ch`+Cv`, then the algorithm goes one level down from Li to Li+1, otherwise it reports C in level Li as the cloaked area as shown in Figure 2.

Using LOCACY in Algorithm 1, we fulfill the location privacy of the client without revealing the exact location to the fog density server, or location-based services (LBS). Moreover, LOCACY works well in regard with the communication cost since it, firstly, uses the fog density server instead of cloud server. Secondly, it goes to a level that contains the cells which are closer in area to the proper cloaked area compared to Aman algorithm. This reduces the required communication rounds between the fog density server and the client device.

4. PERFORMANCE EVALUATION

To evaluate our algorithm (LOCACY), we conducted experiments that simulate the locations of users in a synthetics area. We used experimental method so as to monitor the influence of changing the minimum cloaked area (A-min) on the communication rounds. We compared LOCACY with Aman algorithm in [10], bottom-up algorithm in [15], and the top-down algorithm in [9]. Firstly, we presume that the size of the area covered by the quad tree is 1.6km×1.6km and the total number of users is 10240 distributed using Gaussian random distribution in that area. The area of the minimum cell is 50m × 50m as shown in Figure 7. Communication rounds is considered as a key factor to be measured in the comparison between the rival algorithms. In LOCACY, queries are sent according to the certain value of A-min that is set according to the population density of a specific city or location. Accordingly, we have varied the value of A-min and measure the performance of all algorithms. The experiments have been performed on four distinct numbers of users (K) which are 25, 50, 100 and 200. Each experiment has been repeated 20 times for all algorithms on the same distribution then the average of the results has been calculated. Figure 8 shows the results of these experiments as the value of A-min varies.

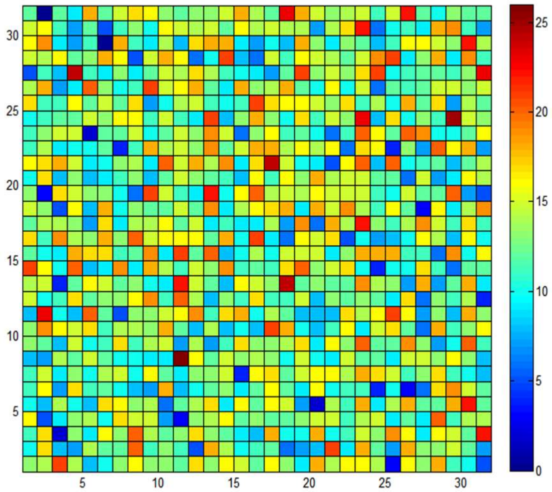


Fig. 7 Gaussian Random Distribution Of The Users

The metric used to evaluate the algorithms is the communication rounds that express the number of rounds between the user mobile device and the user density server. This metric represents the key factor in communication cost and processing time, since the processing time affected by three factors which are: computation time of the algorithms, round trip time (RTT) for the request from client to the server and backward to the client [31], and the number of communication rounds. Since the computation time of the rival algorithms almost the same, and RTT is also equal for all of them, as a result communication rounds play major role in distinguishing between algorithms performance, so that communication rounds represent the best metric to evaluate the communication cost and processing time. For that we focus on communication round to evaluate the rival algorithms.

In this subsection we will evaluate the communication rounds of LOCACY, Aman, Bottom-up and Top-down algorithms with two different values of A-min once as low and once as high. This allows us to experience different assumptions that can affect the performance of these algorithms.

- A-min value is low: Under this assumption (A-min < 0.16 Km^2), the cloaked area is closer to the bottom of the quad-tree, so the Bottom-up algorithm will locate the cloaked area with the minimal communication rounds. While in the Top-down algorithm, more communication rounds are needed to find the cloaked area, since the A-min will be at lowest level in the quad tree. In Aman approach although the algorithm goes to intermediate estimated level based on the total number of users and number of users in A-min. This most likely leads Aman to a higher intermediate level from which it needs to move down in order to reach the proper cloaked area. LOCACY goes almost straight to the level that includes A-min and starts looking for the sufficient number of users, so that the communication rounds would be one or two for all A-min values.
- A-min value is high: In this case (A-min > 0.16 Km2), the cloaked area is closest to the top of the quad tree and the Top-down algorithm can locate the cloaked area with the minimal communication rounds. On the other hand, Bottom-up algorithm needs more communication rounds to find the cloaked area since the A-min is near the top level of the quad tree. In Aman cases, and although the algorithm goes to intermediate estimated level, it needs more communication rounds to find the cloaked area in a higher level that it reaches. LOCACY goes to the level that contains cells of A-min size and starts searching for the required users, so the communication rounds will be one or two for all values of A-min.

As can be noticed from Figure 8-(a), increasing A-min value results in more communication rounds for the bottom-up algorithm which is similar to the behavior of Aman except for the very first value. The number of communication rounds for top-down algorithm decreases as the value of A-min decreases. We can notice the steady behavior of LOCACY with the lowest number of communication rounds. LOCACY outperforms Top-down, Bottom-up and Aman algorithms for most of the A-min values.

In Figure 8 (b-c), increasing A-min value results in less communication rounds for Top-down algorithm and increasing in communication rounds for Bottom-up, and Aman algorithms, but the communication rounds of Aman is less than that of Bottom-up algorithm. LOCACY keeps its steady behavior and achieves the lowest communication rounds.

In Figure 8 (d), the communication rounds decrease for the Top-down algorithm as we increase A-min, while in Bottom-up and Aman algorithms the communication rounds increase as the A-min value increases, Aman algorithm has less communication rounds than Bottom-up in all A-min values. LOCACY has the lowest communication rounds in most of A-min values and get the best performance. LOCACY can be used by a routing algorithm in order to help a user to tackle two main challenges. First, since spatial security needs sufficient users within the same geographical area to preserve anonymity, we need to route the user among a path that satisfies this metric. Secondly the users especially patients with chronic diseases could request to pass through a path that has no infected areas in case of pandemic or epidemic. The solution provided by LOCACY and SecA* for these challenges are illustrated in the following section.

5. CHOOSING SAFE PATH

In order to address the above-mentioned challenges, we developed a Secure version of the A* heuristic search algorithm to provide two main functionalities. The first is to better support the proposed spatial cloaking algorithm, and the second is to allow a mobile patient to avoid infected areas while moving between different places.

The mechanisms should be flexible to accommodate various demands and preferences from users [6].

A* algorithm is one of the well-known methods used in path-finding and is a graph search algorithm that finds a route from a specified source node to a specified target node. A* algorithm heuristic estimated cost from node n to the target. It is relying on testing the best next step in searching for a route. This is done by examining each next step against the heuristic to give a value that can be used to filter the list and therefore determine the next step [29].

A* works by keeping two lists; the open list and the closed list. At the beginning the open list includes the start node when all other nodes that are not considered yet. If there are no nodes in the open list, then there is no possible route toward the destination node. The closed list begins without any node since it will include all visited nodes. The main loop of the algorithm chooses a node named (d) from the open

Algorithm 2 : A* Pseudo, (SecA* code is underlined)

Input: source node, destination node
Output: total path from source to destination
Method:

```
initialize open-list with source node and close-list without any node.
g(source) = 0.
h(source) = heuristic-function (source, destination).
f(source) = g(source) + h(source)
while open-list is not empty
    current = node (d) with least cost taken from open-list.
      check if the number of users in the node d is sufficient.
      check if node d is not in infected area
            If d == destination
                 return
    Remove node d from open-list and add it to close-list.
  Generate child(d), add all child(d) to open-list
            for each q in child(d)
               check if the number of users in node q is sufficient.
                check if node q is not in infected area
            set cost = g(d) + distance (d, q)
            If cost < g(q)    (this path to child q is better than any previous one).
                node q become current node
                g(q) = cost.
                h(q)= heuristic-function (q, destination).
                  f(q) = g(q)+ h(q)
            If q is not in open-list
                  add q to open-list
return failure              (the open-list is empty but the destination is not reached.)
```

list with the minimum estimated value to get to the target.

The SecA* algorithm checks if the number of users in node d is sufficient, if that is true then it checks if node d is not in an infected area. Then If the chosen node is not the target it places all valid neighboring nodes into the open list and repeats the process. The loop ends when either the path to the target is found, or the steps are completed. If the number of users in the node are not sufficient or it is in the infected area, the node is removed from open list to close list. When a path is discovered, all nodes that are generated maintain a reference to their parents. This means that from any node (n), we can backtrack to find a route from that node to the first node [29, 30]. When the steps are finished (open list is empty) without finding route, the algorithm returns failure.

Two functionalities are added to the A* search algorithm that extend its services by adding security measures so that the algorithm can be named as Secure A* (SecA*). The first functionality is the privacy function which verifies the number of users needed by spatial cloaking in each cell. If that number is within the required range, the cell is included in the search scope, otherwise it is excluded. The second functionality is the safety function which checks cells against being part of an infected which mandates the exclusion of these cells.

Experiments Design.

In this section, the efficiency of SecA* in selecting a safe path is experimentally evaluated by simulation. We used experimental method so as to monitor the influence of changing user distribution, and number and location of infected areas on the path length. The simulation is implemented using MATLAB and used to evaluate the algorithm in terms of the distance measured from source to destination. Each result shown in this subsection is an average of 20 runs with various distributions of users in the whole area of the simulation.

First, we describe the graph that represents a certain configuration for the simulation. Blue rectangles represent buildings while the area between and around them represents valid paths. The colored small circles represent the number of users in each cell; yellow circles have (4-6) users, green circles have (6-8) users, blue circles have (8-10) users, and number of users under 4 and above 10 is labeled with red circles, the required number of users is between 4 and 10 users, so the safe path will not pass through red circles. Infected areas are represented by small black boxes filled with an "x" symbol.

In the first experiment, A* search algorithm is executed in two cases, without adding any additional functionality and with adding a privacy functionality. The length of the path is the shortest when the algorithm executed without adding any function to it (original A*), after adding the privacy function, the length of the path increased, because the algorithm selects the path that satisfies the privacy requirement to have sufficient number of users (4 -10 users). The results are shown in Table 2. Figure 9 and Figure 10 illustrate sample cases.

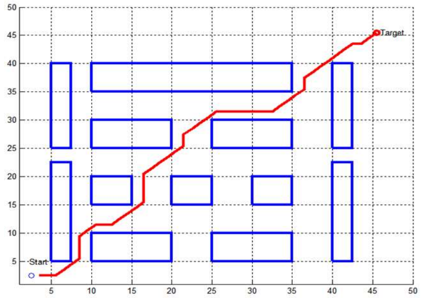
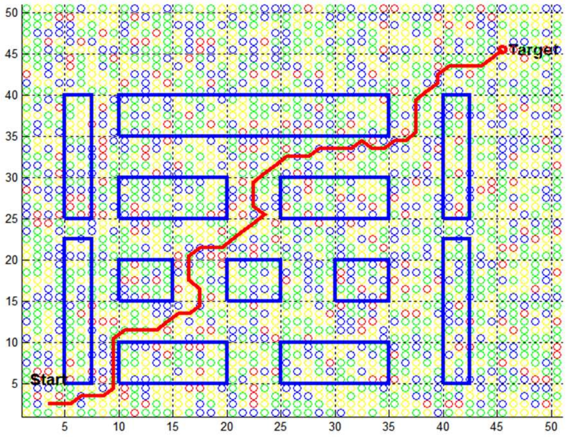


Fig. 9 The Shortest Path For The Original A* Search



Algorithm.

Fig. 10 The Path After Adding Privacy Functionality.

Table 2. The Average Length Of PATH OBTAINED From The First Experiment.

Scenario	Average Length of the path (m)
A* algorithm	6842
SecA*algorithm (privacy functionality)	7677

In the second experiment, the safety functionality is added to A* algorithm. In the first case we assume the same distribution of users, the same number of infected areas, but in different locations, we note that the average length of paths in case of existence of infected areas is longer than the original A* algorithm, because the SecA* algorithm pass through the long path to avoid infected areas, the location of infected area plays major role in specifying the path from source to destination Figure

11 illustrate that. The results in terms of distance are shown in Table 3.

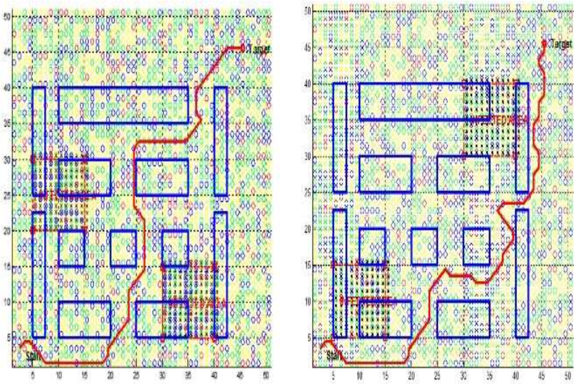


Fig. 11 The Path Of The User For Different Locations Of Infected Areas.

Table 3. The Average Length Of The Paths For Second Experiment Case 1.

Scenario	Average Length of the path (m)
A* algorithm	6842
SecA* algorithm with different locations of infected areas	8560

In the second case, we assume the same distribution of users, fixed location of infected areas, but increasing the number of infected areas. We note that the length of the path is increased while increasing the number of infected areas Figure 12 illustrates sample cases. The change in the length of the path is maximum when we move from one infected area to two infected areas, and minimum when we move from four infected areas to five infected areas which means that after specific number of infected areas, we reach to saturation state as shown in Fig.13

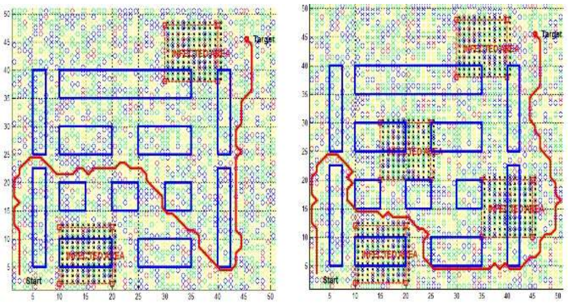


Fig. 12 The Path Of The User For Different Numbers Of Infected Areas.

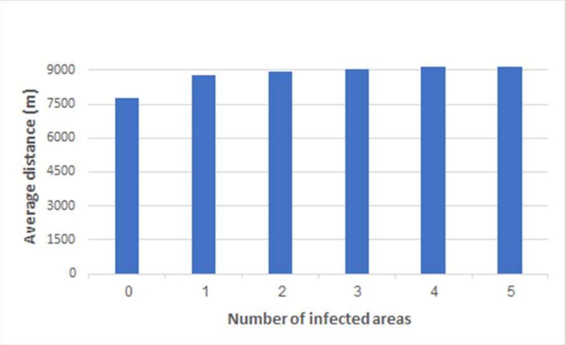


Fig. 13 The Relation Between Number Of Infected Areas And Length Of The Path

In the third case we assume different distributions of users, fixed number of infected areas, with the same location, we note that the path is different according to distribution of the users. The results in terms of distance are shown in Table 4. Figure 14 illustrates sample cases.

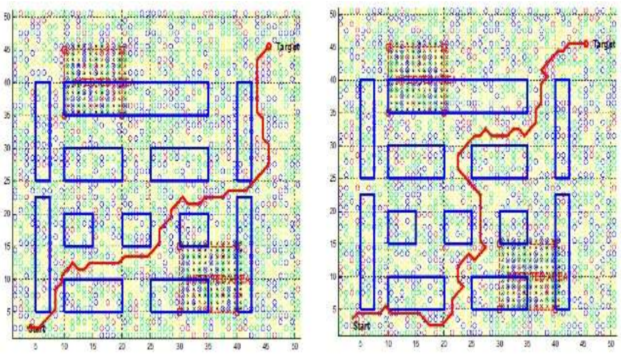


Fig. 14 The Path For Different Distribution Of Users.

Table 4. The Average Length Of Path Obtained From The Second Experiment Case 3

Scenario	Average Length of the path (m)
A* algorithm	6842
SecA* algorithm with different distribution of users	8455

In the third experiment, the algorithm is executed based on the preferences of the user. In the first scenario the user cares about privacy regardless of infected areas or length of the path, Figure 15. In the second scenario the user cares about his/her safety by avoiding the infected areas regardless of privacy level or finding the shortest path, Figure 16. The third scenario assumes that the user cares about shortest distance regardless of privacy or passing through infected areas, Figure 17. The results in terms of distance are shown in Table 5.

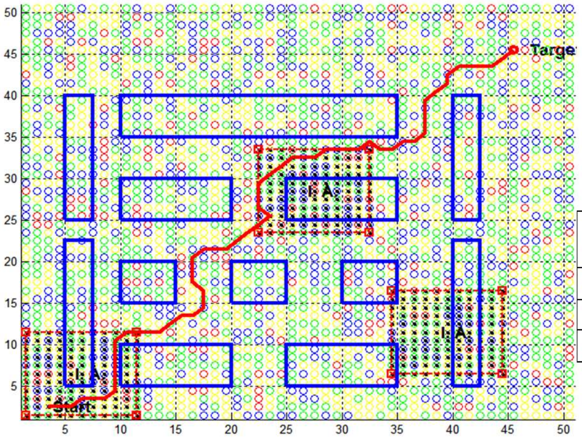


Fig. 15 The User Preference Is The Privacy Regardless Of Infected Area

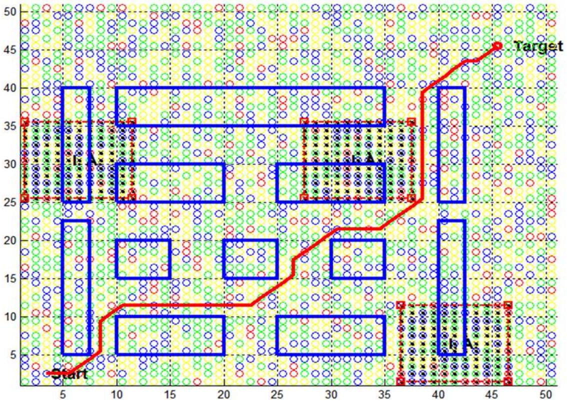


Fig. 16 The User Preference Is Avoiding Infected Area Regardless Of Privacy

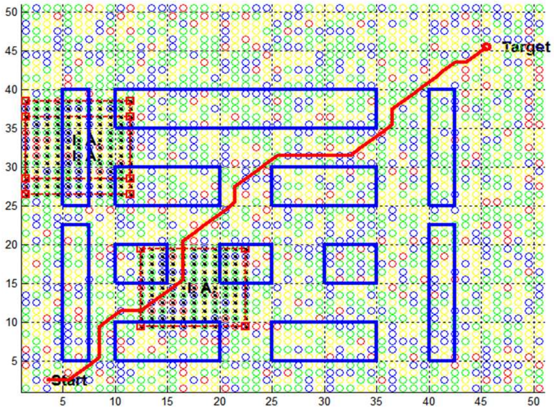


Fig. 17 The User Preference Is The Shortest Path Regardless Of Safety Or Privacy

Table 5. The Average Length Of The Paths For User Preferences.

Preferences	**Average Length of the path (m)**
Privacy functionality	7594
Safety functionality	7252
Shortest path	6842

In the third experiment, SecA* has been simulated according to the user preferences. When the user is focusing on privacy and does not take into consideration the infected areas or finding the shortest path, SecA* will be able to consider routes that pass through infected areas and go through longer paths in order to satisfy the minimum number of user metric (K) related to the privacy factor. Satisfying this metric mandates, the avoiding of many cells that have less than K users and results in the longest path among the three cases considered in this experiment.

In the second case, SecA* tries to avoid infected areas regardless of privacy degree or shortest path finding according to the user preferences. Not taking the shortest path into account ease the job of SecA* in selecting paths that avoid infected areas. Additionally, allowing the path to pass through cells that do not have the required number (K) of users, according to the privacy constraint, allows the selected paths to be shorter than those in the first case.

In the last case, the algorithm chooses the shortest path regardless of the degree of privacy or safety represented by avoiding infected areas. This is the original behavior of A* algorithm which results in the shortest paths of the three cases considered in this experiment. The rationale behind this is the ability of the algorithm to select paths that pass through infected areas and do not satisfy the anonymity constraints in order to find shortest paths.

6. DIFFERENCES BETWEEN LOCACY AND OTHER RIVAL ALGORITMS.

LOCACY proposes an algorithm that has significant advantages comparing with other rival algorithms. Firstly, it uses new technique in searching quad tree by using equation valid in all kind of user distribution in contrast to Aman algorithm in [10], which is valid only when the users are evenly distributed in the space. Moreover, LOCACY uses fog computing instead of cloud computing which reduce the latency. Secondly LOCACY computes the spatial cloaking in device without the need to have third party represented by location anonymizer like in Mokbel et al scheme in [15] which is costly and can become at one point of time compromised. Thirdly in LOCACY there is no need to connect with other users to compute spatial cloaking area like in [12] which assumes that all users are trusted which might be wrong. Moreover, LOCACY outperforms other rival algorithms in term of computation and communication.

7. CONCLUSION

In this paper, we proposed an algorithm, called LOCACY, that optimized the computation of the cloaked area for anonymous location based services. LOCACY reduced the number of communication rounds with a density fog server. It outperforms other rival algorithms in terms of communication and computation costs and achieved average enhancement of 64% better than bottom-up, 55% better than top-down, and 56% better than Aman algorithm. Moreover, a secure version of the A* heuristic search algorithm (SecA*) has been developed to provide two main additional functionalities: the privacy function that better supports the proposed LOCACY spatial cloaking algorithm and the safety function that allows a mobile patient to avoid infected areas while moving between different places. Results of evaluating SecA* algorithm showed that it provides a safe path and improves privacy provision.

REFERENCES:

[1] Gonzalez, L., Wightman Rojas, P., & Labrador, M. (2014). A survey on privacy in location-based services. Ingeniería y Desarrollo, 32(2), 314-343.

[2] Xu, G. (2010). Location cloaking for location privacy protection and location safety protection.

[3] Al Qatawneh, I., Almobaideen, W., Qatawneh, M. (2022). A comparative study on surveillance and privacy regulations (The UAE vs. the USA and the EU), Journal of Governance and Regulation, Vol 11, issue 1, (pp. 20-26).

[4] Asassfeh, M. R, Obeid, N., & Almobaideen, W. (2020). Anonymous Authentication Protocols for IoT based-Healthcare Systems: A survey. International Journal of Communication Networks and Information Security (IJCNIS). Vol. 12, No. 3.

[5] Almobaideen, W., Krayshan, R., Allan, M., & Saadeh, M. (2017). Internet of Things: Geographical Routing based on healthcare centers vicinity for mobile smart tourism destination. Technological Forecasting and Social Change, 123, 342-350.

[6] Nat Gunanathan, I., Mehmood, A., Xiang, Y., Gao, L., & Yu, S. (2018). Location privacy protection in smart health care systems. IEEE Internet of Things Journal, 6(2), 3055-3069.

[7] Gedik, B., & Liu, L. (2008). Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing, 7(1), 1-18.

[8] Niu, B., Li, Q., Zhu, X., & Li, H. (2014, August). A fine-grained spatial cloaking scheme for privacy-aware users in location-based services. In Computer Communication and Networks (ICCCN), 2014 23rd International Conference on (pp. 1-8). IEEE.

[9] Wang, S., & Wang, X. S. (2010, May). In-device spatial cloaking for mobile user privacy assisted by the cloud. In Eleventh International Conference on Mobile Data Management (pp. 381-386). IEEE.

[10] Jadallah, H., & Al Aghbari, Z. (2018). Spatial cloaking for location-based queries in the cloud. Journal of Ambient Intelligence and Humanized Computing, 1-9.

[11] Michael, K., Perusco, L., & Michael, M. G. (2006). Location-based services and the privacy-security dichotomy.

[12] Chow, C. Y., Mokbel, M. F., & Liu, X. (2011). Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica, 15(2), 351-380.

[13] Patil, M. S., & Sutar, S. H. Survey on Location Privacy in Location Based Services

[14] Chow, C. Y., Mokbel, M. F., & Liu, X. (2006, November). A peer-to-peer spatial cloaking

algorithm for anonymous location-based service. In Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems (pp. 171-178). ACM.

[15] Mokbel, M. F., Chow, C. Y., & Aref, W. G. (2006, September). The new casper: Query processing for location services without compromising privacy. In Proceedings of the 32nd international conference on Very large data bases (pp. 763-774). VLDB Endowment.

[16] Chow, C. Y. Spatial Cloaking Algorithms for Location Privacy.

[17] Sharifi, M., & Naghavian, L. (2010). Providing location privacy in pervasive computing through a hybrid mechanism. International Journal of Internet Technology and Secured Transactions, 2(1-2), 160-173.

[18] Qusai Hasan, Sahar Abdelbasit, Hamad Alblooshi, Wesam Almobaideen, Mahmoud Al-Habashneh. (2021). Anonymous Authentication Scheme for Smart Home Environment, 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), (pp. 1-6). IEEE.

[19] Almobaideen, W., Jarboua, H., Sabri, K.E. (2020). Searchable encryption architectures: survey of the literature and proposing a unified architecture, International Journal of Information Privacy, Security and Integrity, Vol 4, Issue 4, (pp. 237-260).

[20] Barik, R. K., Dubey, H., Samaddar, A. B., Gupta, R. D., & Ray, P. K. (2016, December). FogGIS: Fog Computing for geospatial big data analytics. In 2016 IEEE Uttar Pradesh section international conference on electrical, computer and electronics engineering (UPCON) (pp. 613-618). IEEE.

[21] Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2018). Fog computing for Healthcare 4.0 environment: Opportunities and challenges. Computers & Electrical Engineering, 72, 1-13.

[22] Gia, T. N., Jiang, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2015, October). Fog computing in healthcare internet of things: A case study on ecg feature extraction. In 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing (pp. 356-363). IEEE.

[23] Almobaideen, W., Saadeh, M. (2018). Lightweight Authentication For Mobile Users in the Context of Fog Computing, International Journal of Advanced Computational Engineering and Networking, Vol 6, Issue 12, (pp. 2321-2063).

[24] Ardagna, C. A., Cremonini, M., Damiani, E., Di Vimercati, S. D. C., & Samarati, P. (2007, July). Location privacy protection through obfuscation-based techniques. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 47-60). Springer, Berlin, Heidelberg.

[25] Chatzikokolakis, K., ElSalamouny, E., Palamidessi, C., & Pazii, A. (2017). Methods for location privacy: A comparative overview. Foundations and Trends® in Privacy and Security, 1(4), 199-257.

[26] Zhang, X., Kim, G. B., & Bae, H. Y. (2014, October). An adaptive spatial cloaking method for privacy protection in location-based service. In Information and Communication Technology Convergence (ICTC), 2014 International Conference on (pp. 480-485). IEEE.

[27] Gkoulalas-Divanis, A., Kalnis, P., & Verykios, V. S. (2010). Providing k-anonymity in location based services. ACM SIGKDD explorations newsletter, 12(1), 3-10.

[28] Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. Wireless communications and mobile computing, 2(5), 483-502.

[29] Sharma, S. K., & Pal, B. L. (2015). Shortest path searching for road network using a* algorithm. International Journal of Computer Science and Mobile Computing, 4(7), 513-522.

[30] Niu, B., Li, Q., Zhu, X., Cao, G., & Li, H. (2014, April). Achieving k-anonymity in privacy-aware location-based services. In INFOCOM, 2014 Proceedings IEEE (pp. 754-762). IEEE.

[31] Veal, B., Li, K., & Lowenthal, D. (2005, March). New methods for passive estimation of TCP round-trip times. In International workshop on passive and active network measurement (pp. 121-134). Springer, Berlin, Heidelberg.

[32] Zheng, J., Tan, X., Zou, C., Niu, Y., & Zhu, J. (2014, July). A cloaking-based approach to protect location privacy in location-based services. In Control Conference (CCC), 2014 33rd Chinese (pp. 5459-5464). IEEE.