Trajectory Privacy Protection Mechanism based on Salp-Like Swarm Algorithm

Sitong Shi, Jing Zhang*, Yanzi Li, Jianyu Hu
School of Computer Science and Mathematics, Fujian Provincial Key Laboratory of Big Data Mining and Applications, Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou, China.

*Corresponding author. Email: jing165455@126.com

Manuscript submitted November 30, 2021; accepted January 15, 2022.
doi: 10.17706/jsw.17.2.71-86

Abstract: Location-based services have been widely used in daily life, providing diversified services for users. However, users may face the risk of trajectory privacy disclosure while enjoying the convenience of location-based services. Most of the existing trajectory protection schemes cannot match the road network and are vulnerable to attacks based on background information. In this paper, the concept of salp swarm algorithm is introduced to construct salp-like swarm algorithm, which can generate $k-1$ false trajectories that are highly similar to real trajectories. It is difficult for attackers to distinguish them. Besides, a road network matching model is designed in order to match the proposed trajectory privacy protection algorithm with the real road network environment, so that the effect of trajectory privacy protection is improved. Morever, a false location selection mechanism is proposed to find false location points, which not only considers the location and speed of users, but also ensures that the selection of false location points is more in line with the road network environment. The experimental results show that, under the condition of satisfying the same service quality, the trajectory privacy leakage probability of this scheme is reduced by 33% compared with the existing schemes, and it has better privacy protection effect.

Key words: k −anonymous, Location-based services (LBSs), salp-like swarm algorithm (SLSA), trajectory privacy.

1. Introduction

With the rapid development of global positioning technology and mobile devices, location-based services (LBSs) [1] have gained widespread attention and applications in multiple fields. For example, smartphone applications can provide users with diversified location services such as interest point query and recommendation through location information, which makes it more convenient for users to travel [2]. However, while enjoying the convenience of LBSs, users also face a series of security and privacy issues [3]. For example, Location-based social networks will record users' trajectories, if the attacker can reason about the trajectory data, he may get the user's personal interests, habits, behavior patterns and other related privacy information [4]. Therefore, how to ensure that users get high-quality location services and protect users' privacy information from being leaked are the key of LBSs.

Trajectory data not only contains abundant external information such as personal position, but also can calculate the potential information of trajectory by reasoning. The attacker gets the information data of the mobile terminal by mining the trajectory behavior characteristics of the mobile terminal. At present, the commonly adopted privacy protection method is k −anonymous technology [5]. Its basic idea is to generate $k-1$ pieces of false information, which together with the user's real information constitute a track or location-anonymous set. And then send it to the location service providers. However, the existing-anonymous privacy protection algorithms still have the following problems:

1) The existing-anonymous privacy protection algorithms are mainly suitable for Euclidean geometric space, and can't match the actual road network environment. Because in the Euclidean geometric space, the user's movement direction is almost unlimited, while in the road network environment, the user can only move along the path direction [6].
2) Most of the existing anonymous privacy protection algorithms do not consider the construction of anonymous areas. When the anonymous areas are too large, the cost of constructing anonymous areas will be high, and the location points in the anonymous areas are too sparse to confuse. When the anonymous areas are too small, the location points in the anonymous area are too concentrated, and the attacker still has a high probability to calculate the true location. In addition, the setting of the starting points and the ending points are not considered. The starting points and the ending points can contain more information, such as the user's home address and work place.
3) The existing anonymous privacy protection algorithms have the problem that the generated false location points are not real enough. The users' behaviors may be analyzed by attackers through data mining methods, and false location points are easily excluded by attackers [7]. Users' privacy cannot be effectively protected.

To solve the above problems, this paper sets up the road network environment matching, and simulates the group behavior of salp swarm foraging to design the salp-like swarm intelligent algorithm. Besides, the generation of anonymous areas and false location points are improved. Finally, a trajectory privacy protection mechanism based on the algorithm of salp-like swarm is proposed. The main contributions of this paper are summarized as follows:

1) The map network and the road network matching model are constructed, so that the proposed trajectory privacy protection algorithm can match the real road network environment.
2) Combining with the salp swarm algorithm, the salp-like swarm algorithm is proposed, which makes the anonymous areas constructed more reasonable. Furthermore, the starting point set and the ending point set are randomly generated in the same range, then the false trajectories are generated. The generated false trajectories have the same direction and speed as the real trajectory, which can reliably realize the k −anonymity of the trajectory.
3) A false location selection mechanism is proposed, which includes the road network-based movement algorithm and the adaptive position state update algorithm. This mechanism enables the false locations to be generated more reasonably, and the overall privacy protection effectiveness of the algorithm is improved.

2. Related Work

Domestic and foreign scholars have done a great deal of research on the security of trajectory privacy, and put forward a series of trajectory privacy protection methods [8].

The existing trajectory privacy protection technologies are mainly divided into three types: false trajectory method, generalization method and suppression method [9]. The false trajectory method refers to obtaining false trajectories from the historical trajectories of users or the trajectories of neighboring users, and generating certain false trajectories for real trajectories to reduce the exposure probability of real trajectories, thus protecting the privacy of users' trajectories [10]. The generalization method [9] refers to generalizing the location points in the trajectory to a larger anonymous area, so as to reduce the probability that attackers can successfully guess the user's private information and protect the user's real location. The suppression method [9] is to selectively publish the location data on the user's trajectory to suppress the publication of some sensitive or frequently accessed data items.

Based on the false trajectory method, Li et al. and Wang et al. rotate the user's real trajectory according to the selected rotation point, thus generating $k-1$ false trajectories in turn [11], [12]. The generated false trajectories often do not meet the requirements of realistic geographical restrictions. Huo et al. define the problem of selecting anonymous sets of tracks as a graph partition problem, and minimize the partition cost according to the distance between tracks, so as to reduce the information loss [13]. However, they ignore the similarity of trajectories, which makes it easy for attackers to identify users' trajectories by using background information. Yang et al. propose a personalized track association construction method based on the included angle and position coincidence between tracks, and measure the edge weight between tracks according to the distance and direction between tracks, so as to construct a personalized track map model with variable scale [14].

Based on the generalization method, Wang et al. design a privacy protection framework based on $k-$anonymity for continuous query [15]. This method considers the distance limitation, and also establishes a hierarchical structure based on user density and historical records, which accelerates the anonymous processing process in the anonymous server. Chow et al. propose a method to protect the privacy of users under the continuous query [16]. Firstly, two main attacks are identified, namely query sampling attack and query tracking attack. Secondly, determine the shared region attributes and memory attributes. If the technical requirements of location anonymity are met, query sampling attack and query tracking attack can be avoided. Finally, a robust algorithm is proposed, which protects users' privacy under continuous query according to shared area attributes and memory attributes.

Based on the suppression method, Terovitis et al. put forward a method to prevent attackers from inferring other information of users from part of the information obtained [17]. This method suppresses the location information in the part where information leakage occurs. The trajectory database released after processing can ensure that attackers can't reconstruct the user's original trajectory with a probability higher than p.

Although the above methods have achieved certain results in the protection of users' trajectory privacy, there is a problem that the generated false tracks are unreasonable. This paper designs a trajectory privacy protection mechanism based on the salp-like swarm algorithm, and adds the road network model to generate false trajectories according to the real tracks. It can avoid generating false trajectories that are not realistic and greatly improve the protection effect of trajectory privacy.

3. System Model

3.1. System Model

The trajectory anonymity method proposed in this paper adopts the system structure based on the third-party trusted central server. The users send their own trajectory information and query request to the third-party trusted central server through the mobile terminal. After receiving it, the third-party trusted server encrypts the data and sends it to location service provider. The location service provider sends the query results to the central server. Finally, the central server anonymizes the trajectory according to the query request and sends the results to the user. All the privacy data of users' trajectory are stored in the third-party trusted central server, which has high security. The system model is shown in Fig. 1.

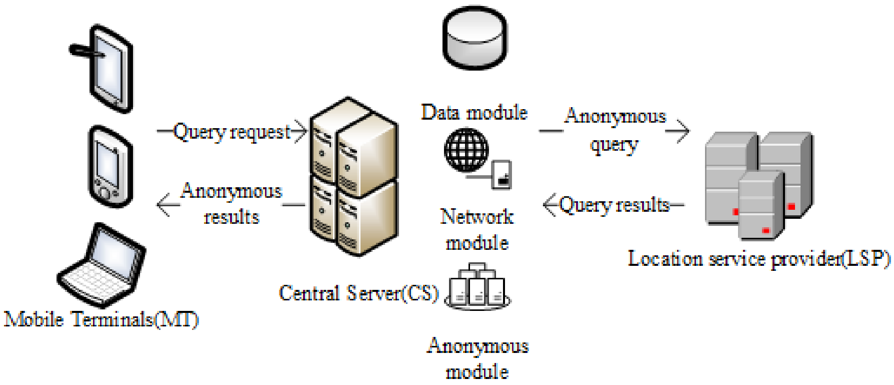


Fig. 1. System model.

Mobile Terminals (MT): Intelligent devices with global positioning technology and wireless network access functions, such as smart phones, tablets and in-vehicle systems, etc. Be able to send location information and query request to the central server.

Central Server (CS): The entity is completely trusted and includes three modules, namely, data module, road network module and anonymous module. The functions of each module are as follows:

Data module: Read and sort out different trajectory data sent by users, and check the validity of the data.

Road network module: Nationalroad network data downloaded from OSM Geofabrik are stored in mysql and redis servers, and intersection points and edges in the road network are taken from redis servers.

Anonymous module: Anonymous processing of query requests and query results according to user requirements.

Location service provider(LSP): It can provide users with location-based service data, such as navigation, interest point query and advertisement recommendation, etc. There are many kinds of LSP, such as Baidu map, Beidou navigation system and Google map. LSP may also disclose users' private information.

3.2. Salp Swarm Algorithm

Salp swarm algorithm is a biological heuristic optimization algorithm proposed by Mirjalili *et al.* [18] for engineering design problems. The main idea is the clustering behavior of salp when sailing and foraging in the ocean. In the deep sea, salps usually form a group called salp chain, then prey and move. The salp chain consists of two types of salp: leader and follower. The leader is salp at the head of the chain, while the rest of salps are regarded as followers. The leader leads the population, and the followers follow each other.

It is assumed that the positions of all salps are stored in a two-dimensional matrix called X, and there is a food source named F in the search space as the target of the population. The leader's next step is to move towards food to a certain extent, only to update its position relative to the food source. Therefore, the formula for updating the position of the leader is shown in formula (1):

$$x_j^1 = \begin{cases} F_j + c_1\left((ub_j - lb_j)c_2 + lb_j\right), c_3 \geq 0 \\ F_j - c_1\left((ub_j - lb_j)c_2 + lb_j\right), c_3 \leq 0 \end{cases} \tag{1}$$

where, x_j^1 is the position of the first salp (leader) in the $j-th$ dimension; F_j is the position of food source in the $j-th$ dimension; ub_j is the upper bound of the $j-th$ dimension; lb_j is the lower bound of the $j-th$ dimension; c_1 is mainly used to control the exploration ability and development ability of the whole population, which is related to the iteration times of the current population. The definition is shown in formula 2.

$$c_1 = 2e^{-\left(\frac{4l}{L}\right)^2} \tag{2}$$

where l is the current number of iterations and L is the maximum number of iterations. The c_2 and c_3 are random numbers generated in the interval [0,1], c_2 determines the length of movement, and c_3 determines the positive and negative direction of movement.

Followers follow the leader to move, and the position of the next iteration of the i −th follower is jointly determined by its own position and the position of the $i-1$ salp in the current iteration. The formula for updating the position of followers is shown in formula (3):

$$x_j^i(t) = \frac{1}{2}\left(x_j^i(t-1) + x_j^{i-1}(t-1)\right) \tag{3}$$

Salp swarm algorithm can explore the optimal solution in the global scope and find the food source. The design based on the road network in this paper is similar to this movement route, but there are differences in the movement mode. Therefore, on this basis, a trajectory privacy protection algorithm based on salp-like swarm algorithm is designed for trajectory privacy protection.

4. Trajectory Privacy Protection Mechanism Based on Salp-Like Swarm Algorithm (TPPM-SLSA)

According to the characteristics of the salp swarm algorithm, this paper designs a trajectory privacy protection mechanism based on the salp-like swarm algorithm. The state of each position point on the trajectory can be regarded as an independent salp. The state of the starting point is regarded as the leader, which directly or indirectly affects the movement state of the followers behind, that is, the state of the next position point. The target is the food source F in the search area, that is, the destination. The resulting $k-1$ salp chains are $k-1$ false trajectories. TPPM-SLSA mainly adopts anonymous region initialization mechanism and false location selection mechanism to protect the privacy of the trajectory. TPPM-SLSA is shown in Fig. 2, and its process is described as follows.

1) Users use mobile terminal equipments to send trajectory parameters and query requests to the central server;
2) After receiving the trajectory parameters and query requests, the central server selects the anonymous area (the detailed process of anonymous area initialization will be described in Section 4.1) according to the user's requirements. At the same time, the starting point set and the ending point set are analyzed and sorted out;
3) The location points constituting the false trajectory are determined (the false location selection mechanism will be described in detail in Section 4.2), and the user-sensitive information is anonymized. Then sent them to the location service provider together;
4) The location service provider processes the query according to the received anonymous request, and then sends the query result back to the central server;
5) The central server sends the filtered anonymous result to the user.

Salp-Like Swarm Algorithm (SLSA): Traversing from the starting point, each point connected with the dividing line is stacked. Search the points on the top of the stack up and down respectively. If the search has been completed , the stack will be released. In the process of searching, prestatusMap is used to record all the motion states until the end point, which is used to trace back the whole trajectory, so as to obtain the virtual trajectory. In the worst case, the algorithm will traverse the upper and lower surfaces of the starting and ending connecting lines instead of the whole graph, which greatly reduces the complexity of the algorithm.

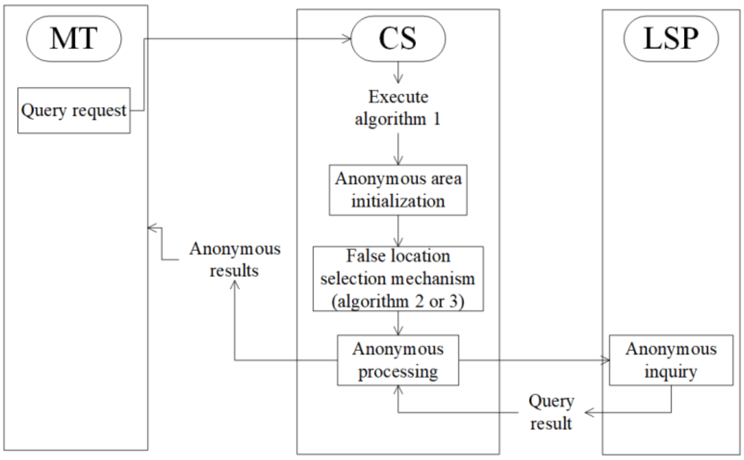


Fig. 2. Workflow.

Algorithm 1: Salp-Like Swarm Algorithm (SLSA)

```
Input: start point, end point.
Output: false trajectory L
begin:
initialize stack and prestatusMap；  ;
startstatusV=(start,vDegree)
Add startstatusV to the stack
while stack is not empty.
    currentstatusV=the top element of the stack popped from the stack,
        if currentstatusV=（end,vDegree），
        return；
    for tempstatusV in nextstatusVs    //Get the next movable point of currentstatusV
        if tempstatusV has no parent node in prestatusVMap.
            put tempstatusV in the stack
        L=(tempExtStatusVs)
        end if
    end for
return L .
end
```

4.1. Anonymous Area Initialization

When protecting track privacy, the generated false trajectories cannot be too close or too far away from the user's real trajectory. When the false trajectory is too close to the real trajectory of the user, it may expose the real trajectory of the real user. However, when the false trajectory is too far away from the user's real trajectory, it may not have a good protection effect on the real track. The position information of the false trajectory should not only meet certain distance constraints, but also have certain similarity with the real trajectory information in direction and speed. The false trajectory formed by connecting the generated false positions should have certain similarity with the user's real trajectory, so as to better protect the user's trajectory privacy.

As shown in Fig. 3., firstly, the center point *C* of the anonymous region is determined according to the real trajectory. A rectangular anonymous area centered on C is generated, which is divided into “well” shaped region. Then, the starting point and the ending point are randomly generated at the regional boundary for path planning. The anonymous area center is determined by formula (4).

$$C = \left(\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2}\right) \tag{4}$$

where C represents the center point of the anonymous region, x_{min} is the minimum longitude of the position point on the real trajectory, x_{max} is the maximum longitude of the position point on the real trajectory, y_{min} represents the minimum latitude of the position point on the real trajectory and y_{max} represents the maximum latitude of the position point on the real trajectory.

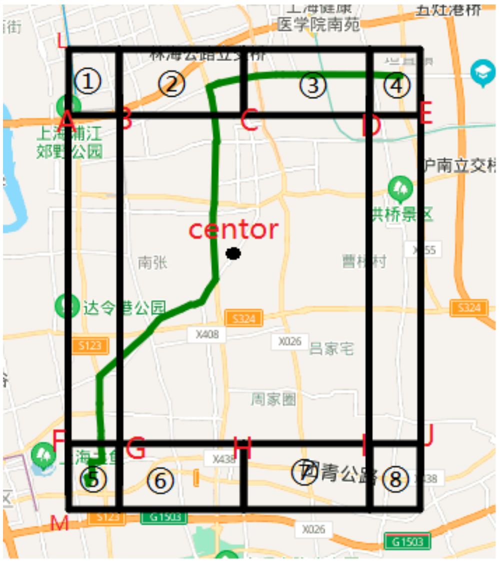


Fig. 3. Anonymous region.

4.2. False Location Selection Mechanism

4.2.1. Road network-based movement algorithm (RNM)

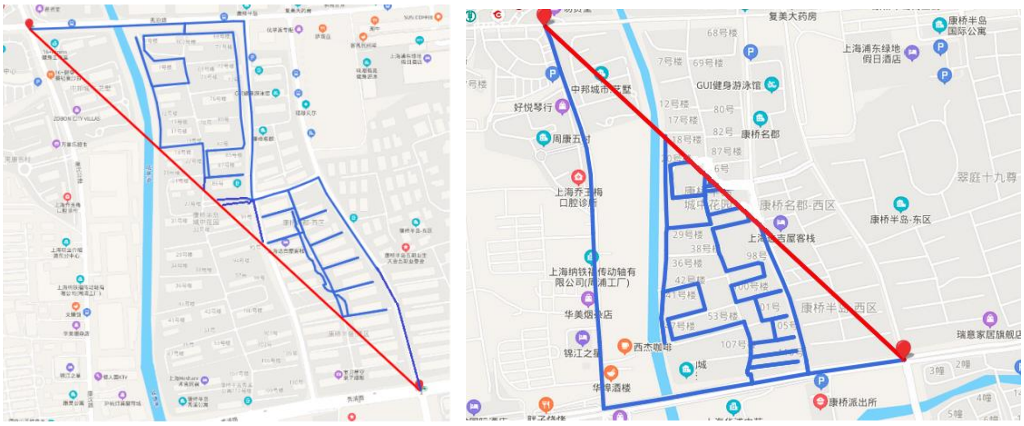
The anonymous region is divided into upper and lower surfaces by the connection between the starting point and the end point of the real trajectory, as shown in Fig. 4. If there is an accessible path from the starting point to the end point in the upper or lower surface, you can move to the end point along the path of this surface. If there is an accessible path from the starting point to the ending point in the anonymous area, but not limited to the upper or lower surface, you can find an accessible path on the upper and lower surfaces along the dividing line. If there is no reachable path from the current location point to the next location point, the algorithm will only traverse the upper and lower surfaces of the dividing line at most, and the complexity will be greatly reduced.

In the process of finding the reachable path, the road network loop may be encountered, which leads to the endless cycle of traversal, so it is necessary to judge the next traversal point. In this section, the position and speed direction of the point are used as the position state of the point, which is used to determine whether to traverse the point next time. In the process of traversal, the algorithm records the point where there is a circular road network. In the next traversal, if the state does not exist, the point will be traversed, and the point will be stored in the traversed set; otherwise, the point will not be traversed.

In this section, the movement algorithm based on road network is designed. When a certain position point and its surrounding connected points are on the same surface, the next route selects the route with the largest difference between the current position and the speed direction. The calculation formula of the movement strategy of the upper and lower surfaces is shown in formula (5):

$$\begin{cases} \theta_i' = \theta - \theta_i \pm 360(-180 \le \theta' < 180)，\mathit{The\ upper\ surface} \\ \theta_i' = \theta_i - \theta \pm 360(-180 \le \theta' < 180)，\mathit{The\ lower\ surface} \end{cases} \tag{5}$$

where, θ represents the included angle between the current speed direction and the horizontal direction, θ_i represents the included angle between the speed direction of the ith route entrance and the horizontal direction, and θ_i' indicates the difference between the speed angle of the ith route entrance and the current position.



(a). The upper surface of the map (b). The lower surface of the map

Fig. 4. Dividing anonymous region.

Algorithm 2: Road network-based movement algorithm(RNBM)

```
Input: current position state CS = (aboveOrbelow, vDegree, location), set of position states of adjacent points NL
Output: next position status NS
begin:
maxDegree=-sys.maxint;   //Find the maximum difference function;
maxDegreeLocation=null;
aboveOrbelow == 0 Or 1;   // Determine whether to move up or down;
  vDegree= the angle between the current speed direction and the horizontal direction;
  for each location in NL;   // Traverse all adjacent points;
        nextVDegree= = the angle between the speed direction of the next point and the horizontal direction;
        if 0 == aboveOrbelow
              tempDegree = vDegree -nextVDegree;   //The upper surface;
        else:
              tempDegree = nextVDegree - vDegree；  //The lower surface;
        if tempDegree > maxDegree:
              maxDegree = tempDegree, // find the maximum included angle;
              maxDegreeNode = tempLocation, // The position point of the largest angle is the desired position;
        NS=(maxDegreeNode,maxDegree)
        end if
end for
return $NS$    //next position status
end
```

4.2.2. Adaptive position state update algorithm (APSU)

The road network-based movement algorithm is only applicable to the situation that the current location point is not on the dividing line and the points connected with it are all on the same side. However, the user's position and speed are constantly changing [19]. In order to adapt to other situations, an adaptive position state update algorithm is designed. It can be divided into the following two situations:

1) If the location point is just on the dividing line, it can be moved up or down. According to the surface movement strategy, the maximum value is obtained, that is, the point that can move on another surface;
2) If the position point is not on the dividing line, the algorithm 1 is used for the adjacent points on the same plane as the current position point, while all the adjacent points on the other plane need to traverse in turn.

Algorithm 3: Adaptive position update state (APSU)

```
Input: current position state $CS = (aboveOrbelow, vdegree, location)$, position state set of adjacent points $NL$
output: position state set of next moment $TNS$
begin:
initialize the resultList; ;
currentDegree = currentStatusV.getDegree(),    //the direction of the current motion state.
  if current location on the line,
      $currentstatus == 0\ Or\ 1$
      nextLocation=algorithm 1,
      resultList.add(nextLocation);
  else
      nextsampleSidesLocation moves with algorithm 1; //sampleSidesLocation is the point consistent with the current location movement strategy.
          For temlocation in otherSidesLocationList;
              resultList.add(temstatus)
        TNS=resultList
          end for
  end if
return TNS.
end
```

5. Experimental Analysis

5.1. Experimental Environment

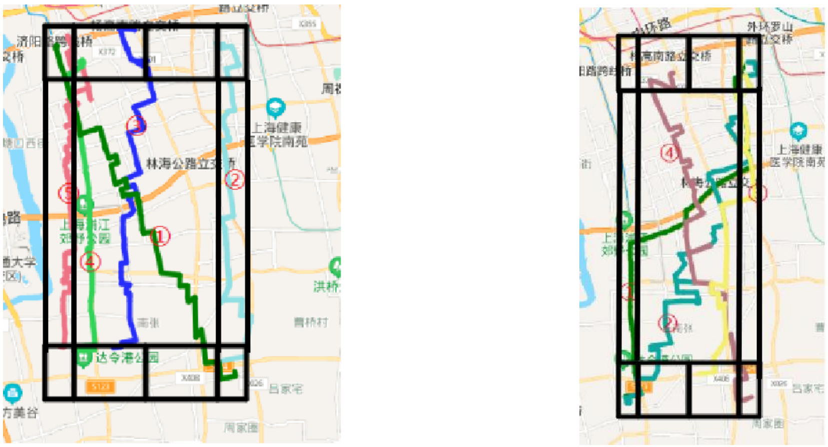
In this section, the scheme is simulated and analyzed, and the simulation platform is implemented by JAVA language. The computer platform used in the experiment is a Windows 10 64-bit computer with Intel Core i5-6300HQ and 8GB memory. Development environment: IntelliJ IDEA 2021 Edition. The real trajectory data used in the analysis are ten trajectories randomly marked in Shanghai. The sum of the total lengths of the tracks is about 53km, the number of sampling positions of each track is between 10 and 35, and the sampling interval is about 200m.

5.2. Experimental Results

In this section, the feasibility of the algorithm, trajectory similarity, trajectory leakage probability and time complexity are analyzed experimentally. At the same time, the SLSA algorithm proposed in this paper is compared with the algorithm proposed in [20] (MPTTA algorithm) and the algorithm proposed in [21] (Random algorithm). To ensure the accuracy of the results, all experimental results are the average of 200 runs.

5.2.1. Feasibility analysis

As shown in Fig. 5., the trajectory is visually displayed. ① represents the real trajectory, ②, ③, ④ and ⑤ represent different false trajectories respectively. It can be seen from Fig. 5. that the false positions generated based on the road network are all on the road, and the generated false trajectories will not deviate from the road. The direction and length of the false trajectory generated in the rectangle area are similar to that of the real trajectory. In addition, there are few overlapping points between the generated false trajectory and the real trajectory, so the attacker can't infer the real trajectory from the false position, and it is not easy to locate the real trajectory.



(a). Example 1 of false trajectories generation (b). Example 2 of false trajectories generation

Fig. 5. Comparison of true and false trajectories.

5.2.2. Trajectory similarity analysis

After determining the starting point set and the ending point set in the anonymous area, the path planning is carried out, so that the false trajectory and the real trajectory have the same speed and direction. Therefore, this paper uses the average error rate of true and false trajectories to measure the similarity of trajectories.

Definition 1: (Error rate): Error rate (E_i) is an index used to evaluate the difference between the real trajectory length and the i-th false track length. The formulation is shown in formula (6):

$$E_i = \frac{|L_{real} - L_i|}{L_i} \tag{6}$$

where, L_{real} represents the length of the real trajectory, L_i represents the length of the i-th false trajectory, and $k-1$ represents the number of false trajectories. The smaller the value of E_i, the smaller the difference between the length of the real trajectory and the length of the i-th false trajectory. The average error rate is expressed as:

$$\bar{E_\iota} = \frac{1}{k-1}\sum_{i=1}^{k-1}\frac{|L_{real} - L_i|}{L_i} \tag{7}$$

Definition 2: (Trajectory similarity ratio): Trajectory similarity (S) is an index used to evaluate the similarity between real and false trajectories. The formulation is shown in formula (8):

$$S = 1 - \bar{E_\iota} = 1 - \frac{1}{k-1}\sum_{i=1}^{k-1}\frac{|L_{real} - L_i|}{L_i} \tag{8}$$

The lower the average error rate of trajectory, the higher the similarity of trajectory, the higher the usability of trajectory data. Trajectory similarity reflects the privacy protection effect of trajectory k-anonymity to a certain extent. The Fig. 6 shows the average error rate of trajectories under different anonymity levels, and Fig. 7 shows the influence of the change of k on the similarity of trajectories. It can be seen that the similarity of trajectories generated by Random algorithm is low. The reason is that the user's action direction and speed are not considered in the design of the scheme, and the false trajectories are randomly generated. The MPTTA algorithm combines the user's attributes with the construction of anonymous sets, and adds the trace graph to simulate the relationship between traces, which improves the similarity of traces. The SLSA algorithm proposed in this paper not only fully considers the position and speed of users when generating false trajectories, but also adds road network matching. The generated false trajectories have a high similarity with the real trajectory, which is stable between 0.85 and 0.9. Comparing with the Random algorithm, the trajectory similarity of this scheme is improved by 35.62%.

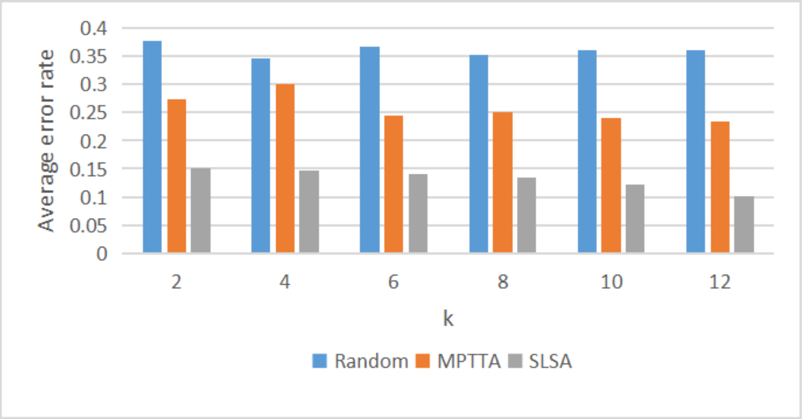


Fig. 6. The influence of the change of k on the average error rate of trajectory.

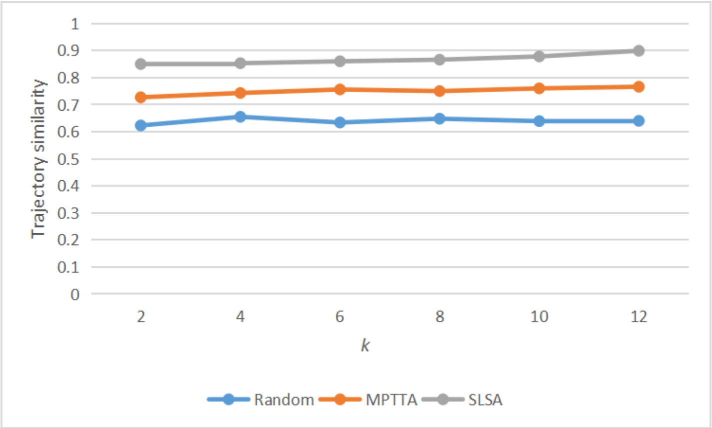


Fig. 7. The influence of the change of k on the trajectory similarity.

5.2.3. Trajectory leakage probability analysis

The trajectory leakage probability is the probability that attackers can identify false trajectories in anonymous sets by calculating the similarity and coincidence between false trajectories and real trajectory.

To achieve the k- anonymity of trajectory, it is necessary to select $k-1$ false trajectories with high similarity with real trajectory from anonymous sets. According to the analysis of trajectory similarity, it can be seen that the smaller the average error rate, the higher the trajectory similarity, the lower the probability of trajectory leakage. In addition, it is also necessary to consider the leakage of the position on the real trajectory. If there are too many overlapping points between the false trajectory and the real trajectory, the attacker can infer the user's real trajectory from the position points. The real trajectory will face the risk of leakage. The definition of trajectory coincidence degree is given below.

Definition 3: (Trajectory contact ratio): Trajectory coincidence ratio (C) is an index used to evaluate the coincidence ratio of real trajectory and false trajectory. The formulation is shown in formula (9):

$$C=\frac{1}{k-1}\sum_{i=1}^{k-1}\frac{n_i}{m} \tag{9}$$

where, m indicates the number of position points on the real trajectory, n_i is the number of overlapping points between the i-th false trajectory and the real trajectory, and $k-1$ reprensents the number of false trajectories. The smaller the value of C, the smaller coincidence degree between the real trajectory and the false trajectory.

Considering the two factors of trajectory similarity and trajectory coincidence, the formulation of trajectory leakage probability is shown in formula (10):

$$P=\frac{1}{S}+C \tag{10}$$

The smaller P, the better the privacy protection effect of users' trajectory.

This section gives the comparison of trajectory leakage probability under different methods. In Fig. 8., at the same anonymous level, the Random algorithm has the highest trajectory leakage probability, followed by MTPPA, and the SLSA algorithm has the lowest. In the trajectories generated by Random algorithm, there are many position points that do not conform to the actual motion law, which are easy to be identified by background information, resulting in poor privacy protection effect. Although MPTTA takes account of users' attributes, it uses historical trajectories and real trajectory to build anonymous sets, which leads to high trajectory coincidence and easy track privacy leakage. The SLSA algorithm is based on the road network environment, and uses the intelligent algorithm of salp-like swarm to generate false trajectories. To a certain extent, the deficiency of anonymous failure caused by the change of real geographical environment is reduced. At the same time, the false trajectory is generated based on the direction and speed of the real trajectory, which makes it difficult to identify through background information, thus reducing the probability of privacy leakage.

5.2.4. Time complexity analysis

This section verifies the comparison of time complexity under different methods. The experimental results are shown in Fig. 9. The Random algorithm has the highest execution efficiency and is nearly linear because it does not consider any other information. However, the faster execution efficiency is at the expense of users' trajectory privacy. In MPTTA algorithm, simulated annealing algorithm is used to solve the approximate optimal anonymous set, and the execution time of the algorithm is relatively long. In SLSA algorithm, the generation of false position only needs to consider two factors, namely position and velocity direction, without considering other factors too much. Therefore, the algorithm has high execution efficiency and relatively short execution time. With the increase of k, the difference between the time complexity of this scheme and MPTTA algorithm is also increasing. When $k = 4$, the time complexity of SLSA algorithm is 33ms lower than that of MPTTA algorithm, and when k =12, the time complexity of SLSA algorithm is 260ms lower than that of MPTTA algorithm.

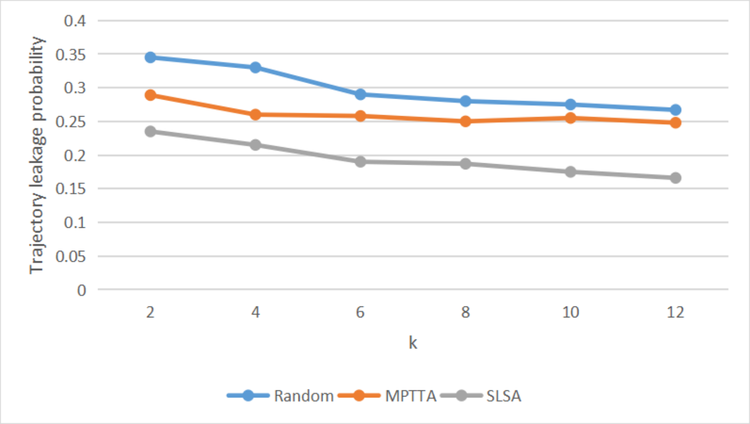


Fig. 8. Comparison of trajectory leakage probability under different methods.

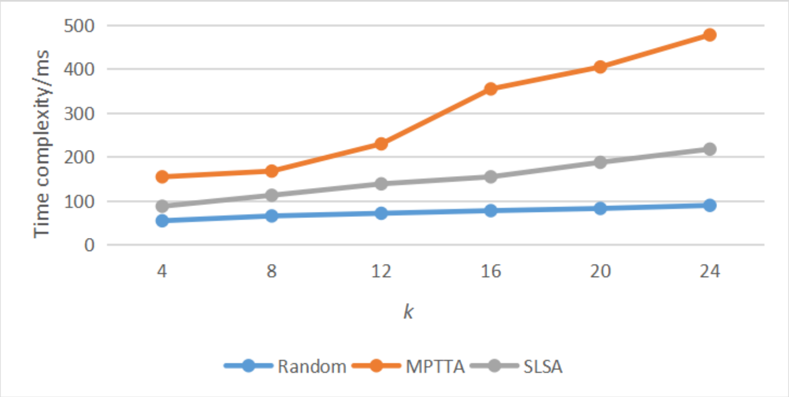


Fig. 9. Comparison of time complexity under different methods.

6. Conclusion

In order to solve the problem of trajectory privacy protection in the road network environment, this paper proposes a trajectory privacy protection mechanism based on the salp-like swarm algorithm. In this paper, the road network module is added to the scheme, so that the proposed trajectory privacy protection algorithm can run in the real road network environment. At the same time, by introducing the concept of the salp swarm algorithm, the salp-like swarm algorithm is imitated and designed, which makes the selection of false location points more reasonable and realizes the privacy protection of the user trajectory k −anonymity level. The experimental results verify the effectiveness and efficiency of the proposed scheme. Under the condition of satisfying the same quality of service, the trajectory privacy leakage probability of this scheme is reduced by 33% compared with the existing schemes. Moreover, the design based on the third-party trusted central server balances the problem between the excessive cost of the anonymous privacy protection algorithm and the privacy protection effect.

In the future work, it is necessary to further study the security of the starting point set and the ending point set of the trajectory to improve the privacy protection effect of users. More users' behavior patterns should be considered to improve the similarity of trajectories and reduce the probability of trajectory privacy leakage.

Funding: This work is supported by the National Natural Science Foundation of China (No.61902069) and Natural Science Foundation of Fujian Province of China (2021J011068).

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Sitong Shi completed the experimental part and analyzed the data; Jing Zhang wrote the paper; Yanzi Li guided the writing; Jianyu Hu checked the paper and experimental data. All authors had approved the final version.

References

[1] Wang, H., & Lu, Z. Y. (2020). Preference-aware sequence matching for location-based services. *GeoInformatica*, *24(1)*, 107-131.

[2] Zhou, J., Cao, Z.F., & Qin, Z. (2020). LPPA: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in VANETs. *IEEE Transactions on Information Forensics and Security*, *15*, 420-434.

[3] Ni, L., Liu, Y., & Liu, Y. (2020). Privacy protection model for location-based services. *Journal of Information Processing Systems*, *16(1)*, 96-112.

[4] Zhang, H., & Cai, Z. L. (2021). A potential friend recommendation algorithm for obtaining spatial information. *Journal of Software*, *16(2)*, 46-54.

[5] Jia, J., & Zhang, F. L. (2014). Nonexposure accurate location k-anonymity algorithm in LBS. *The Scientific World Journal*, 619357-619357.

[6] Fu, Y., & Wang, H. (2019). Virtual trajectory filling algorithm for location privacy protection. *Computer Applications*, *39(8)*, 2318-2325.

[7] Teng, Y. X. (2020). A survey of mining software repositories in social network. *Journal of Software*, *15(2)*, 62-67.

[8] Hasan, A. S. M. T., Qu, Q., & Li, C. M. (2018). An effective privacy architecture to preserve user trajectories in reward-based LBS applications. *ISPRS International Journal of Geo-Information*, *7(2)*, 53-53.

[9] Zhao, J., Zhang, Y., & Li, X. H. (2014). Trajectory privacy protection method based on trajectory frequency suppression. *Journal of Computers*, *37(10)*, 2096-2106.

[10] You, T., Peng, W. C., & Lee, W. C. (2008). Protecting moving trajectories with dummies. *Proceedings of the International Conference on Mobile Data Management.*

[11] Li, F. H., Zhang, C., & Niu, B. (2015). Efficient scheme for user's trajectory privacy. *Journal of Communications*, *36(12)*, 114-123.

[12] Wang, T., Zeng, J. D., & Tian, H. (2017). Trajectory privacy preservation based on a fog structure for cloud location services. *IEEE Access*, *5*, 7692–7701.

[13] Huo, Z., Huang, Y., & Meng, X. F. (2011). History trajectory privacy-preserving through graph partition. *Mobile Location-Based Service,* ACM Press, New York, 71-78.

[14] Yang, J., Zhang, B., & Zhang, J. P. (2015). Personalized trajectory privacy preserving method based on graph partition. *Journal on Communications*, *36(3)*, 1-11.

[15] Wang, Y., Xia, Y., & Hou, J. (2015). A fast privacy-preserving framework for continuous location-based queries in road networks. *Journal of Network & Computer Applications*, *53(C)*, 57-73.

[16] Chow, C. Y., & Mokbel, M. F. (2008). Enabling private continuous queries for revealed user locations. *Proceedings of the Advances in Spatial & Temporal Databases, International Symposium*.

[17] Terovitis, M., & Mamoulis, N. (2008). Privacy preservation in the publication of trajectories*. Proceedings of the Intentional Conference on Mobile Data Management*.

[18] Mirjalili, S., Gandomi, A. H., & Mirjalili, S. Z. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114(6)*, 163-191.

[19] Ghoneim, A. (2018). A new service oriented framework for self-adapting smart applications in mobile environment. *Journal of Software*, *13(3)*, 180-191.

[20] Xu, Z. P., Zhang, J., & Tsai, P. W. (2021). Spatiotemporal mobility based trajectory privacy-preserving algorithm in location-based services. *Sensors*, *21(6)*, 2021-2021.

[21] Hidefoshi, K., Yanagisaway, Y., & Satoh, T. (2005). An anonymous communication technique using dummies for location-based services. *Proceedings of International Conference on Pervasive Services* (pp. 88-97).

Sitong Shi received her B.S. degree from Shanxi Normal University, China, in 2018. She is a graduate student at the College of Computer Science and Mathematics, Fujian University of Technology. Her research interests include network and information security, location privacy protection.

Jing Zhang is a professor and master Supervisor at the School of Computer Science and Mathematics, Fujian University of Technology. She received her Ph.D. degree in mathematics from the College of Mathematics and Informatics, Fujian Normal University in 2015. She was a visiting scholar with Ulster University, UK, during 2018 and 2019. She is the person in charge of Cyberspace Security, Fujian University of Technology. She is also the kernel members in Fujian Provincial Key Laboratory of Big Data Mining and Applications at Fujian university of Technology. She has authored more than 30 papers. Her research interests include network algorithms, network and information security.

Yanzi Li received her B.S. degree from Qingdao University of Technology, China, in 2020. She is a graduate student in the School of Computer Science and Mathematics at Fujian University of Technology. Her research interests cover network and information security, location privacy protection.

JianYu Hu is a master student at School of Computer Science and Mathematics in Fujian University of Technology, Fuzhou, China. His research is focused on network security and big data mining.