Privacy Preservation of Semantic Trajectory Databases using Query Auditing Techniques

Despina Kopanaki
Dept. of Informatics
University of Piraeus, Greece
dkopanak@unipi.gr

Nikos Pelekis
Dept. of Statistics and Insurance Science
University of Piraeus, Greece
npelekis@unipi.gr

ABSTRACT

Existing approaches that publish anonymized spatiotemporal traces of mobile humans deal with the preservation of privacy operating under the assumption that most of the information in the original dataset can be disclosed without causing any privacy violation. However, an alternative strategy considers that data stays in-house to the hosting organization and privacy-preserving mobility data management systems are in charge of privacy-aware sharing of the mobility data. Furthermore, human trajectories are nowadays enriched with semantic information by using background geographic information and/or by user-provided data via location-based social media. This new type of representation of personal movements as sequences of places visited by a person during his/her movement poses even greater privacy violation threats. To facilitate privacy-aware sharing of mobility data, we design a semantic-aware MOD engine were all potential privacy breaches that may occur when answering a query, are prevented through an auditing mechanism. Moreover, in order to improve user friendliness and system functionality of the aforementioned engine, we propose *Zoom-Out* algorithm as a distinct component, whose objective is to modify the initial query that cannot be answered at first due to privacy violation, to the 'nearest' query that can be possibly answered with 'safety'.

Keywords

Privacy-aware query engine; mobility data; anonymity; semantic trajectories.

1. INTRODUCTION

Nowadays, the ease of collecting and storing data from an increasing variety of devices where positioning technologies (GPS) are embedded along with their enhanced processing capabilities, led inevitably to the desire of revealing useful information from them. Alongside this progress, potential breaches of individuals' privacy came to the light. Space-time 'fingerprints' of each recording moving entity (i.e. trajectories) may prove to be a dangerous tool in the hands of a malicious user. The scientific community has proposed various approaches to protect individual's privacy ([1][2][6][7][8][9][11][16]).

Most of the aforementioned studies, define raw trajectories as sequences of points on a geometric space, focusing on their spatiotemporal nature without complementing raw data with additional information from the application context. However, the increasing need of analyzing mobility data has led to the representation of trajectories with contextual data from external data sources, thus transforming raw trajectories to the so-called semantic trajectories. A semantically-annotated trajectory, in short semantic trajectory, is considered as a sequence of stops (i.e. places where the object remains "static") and moves (i.e. parts of the object's trajectory in between two stops) [12]. This alternative representation of trajectories may pose even greater privacy violation threats. Consider for example a malevolent user who is able to detect places of interest (POIs) where a moving object has stopped (e.g. home, hospital, betting office, etc.). This additional knowledge allows the inference of personal sensitive information of this specific individual. On the one hand, analyzing semantically-enriched movement traces of users can aid decision making in a wide spectrum of applications, on the other hand, the disclosure of such data to untrusted parties may expose the privacy of the users whose movement is recorded. Sharing user mobility data for analysis purposes should be done only after the data has been protected against potential privacy breaches.

Most of the methodologies that have been proposed in the literature, aim at protecting users' privacy by releasing an anonymized version of the original dataset ([1][2][6][7][8][9][11][16]). These approaches assume that in the anonymized dataset a malevolent user will not be able to link a specific user with a movement. In this paper, we employ a more conservative approach to privacy by assuming that data stays in-house to the hosting organization in order to prevent any privacy breach. An auditing mechanism is responsible to control the information that is released to third parties and ensure privacy-aware data sharing.

Gkoulalas-Divanis et al. [5] first proposed an envisioned query engine where subscribed users have gained restricted access to the database to accomplish various analysis tasks. Pelekis et al. [13] then proposed HERMES++, a privacy-aware query engine that can protect the trajectory database from potential attacks, while supporting popular queries for mobility data analysis. Both approaches deal with spatiotemporal trajectory databases.

Inspired by the work previously described and considering the richer representation of semantic trajectories, we design a query-based auditing mechanism that can effectively identify and block a range of potential attacks that could lead to user identification or tracking, for privacy-aware sharing of in-house semantic mobility data. The proposed mechanism provides an answer if *k*-anonymity principle is not violated w.r.t the user's current history. Moreover, we propose an algorithm, called *Zoum-Out*, which modifies the original query that cannot be answered at first due to privacy restrictions, to the most similar query that can be safely answered. The algorithm generalizes space, time and/or semantic dimension of one or more sub-queries w.r.t. to a distortion threshold.

Summarizing, in this paper we make the following contributions:

- We identify various types of attacks and thus privacy violations that malevolent users may try to pursue when querying the original semantic trajectory database.
- We design a query-based auditing mechanism that can effectively identify and block a range of potential attacks that could lead to user identification or tracking.
- We propose *Zoom-Out* algorithm aiming at increasing user friendliness of the proposed mechanism by modifying the (original) query posed that cannot be answered due to privacy violation, to the 'nearest' possible 'safe' query.

The rest of the paper is structured as follows: Section 2 presents related work. Section 3 introduces different types of attacks of a malevolent user. Section 4 provides the auditing mechanism that handles the previously described attacks as well as the *Zoom-Out* algorithm. Finally, Section 5 concludes the paper.

2. Related Work

Methods that have been proposed so far to tackle the issue of privacy-preserving mobility data publication mostly adopt the principle of *k*-anonymity, which was originally proposed for relational databases [15]. *k*-anonymity principle is the most common approach that has been adopted for the anonymization of both relational and mobility data. For mobility data, it states that a dataset must be anonymized so that every trajectory is indistinguishable from at least *k*-1 other trajectories.

Hoh and Gruteser [6] presented a data perturbation algorithm that is based on path crossing. When two non-intersecting trajectories are close enough, it generates a fake crossing in the sanitized dataset to prevent adversaries from tracking a complete user's trajectory. Terrovitis and Mamoulis [16] consider datasets as sequences of places visited by users. Based on the assumption that a malevolent user holds partial information of users' trajectories, a suppression technique is proposed that eradicates the least number of places from a user's trajectory so that the remaining trajectory is *k-ano*nymous.

Abul et al. [1] proposed a *k*-anonymity approach that relies on the inherent uncertainty of moving objects whereabouts where a trajectory is considered as a cylinder. The anonymity algorithm identifies trajectories that lie close to each other in time, employs space translation and generates clusters of at least *k* trajectories. Each cluster of *k* trajectories forms an anonymity region and the co-clustered trajectories can be released. To achieve space-time translation, the authors proposed W4M [2], which uses a different distance measure that allows time-warping.

Nergiz et al. [11] proposed a coarsening strategy to generate a sanitized dataset that consists of *k*-anonymous sequences. The algorithm first generalizes a set of trajectories into a set of sequences of *k*-anonymized regions, reconstructs, consolidates the trajectories of the original dataset into clusters of *k* and anonymizes the trajectories in each cluster. Monreale et al. [9] proposed another anonymization approach that is based on the combination of spatial generalization and *k*-anonymity principle. The geographical area covered by the trajectories belonging to the dataset is partitioned into sub-areas. The original trajectories are then generalized and transformed so as to satisfy *k*-anonymity principle.

Mahdavifar et al. [8] introduced the idea of non-uniform privacy requirements, whereby each trajectory is associated with its own privacy level indicating the number of trajectories it should be indistinguishable from. Kopanaki et al. [7] introduced the problem of Personalized *(K,Δ)*-anonymity where user-specific privacy requirements are used to avoid over-anonymization and decrease information distortion. They proposed efficient modifications to state-of-the-art *(k,δ)*-anonymization algorithms by introducing techniques built upon users' personalized privacy settings and trajectory segmentation.

Recently, in [10] authors faced the problem of anonymizing semantic trajectories. To release a safe version of a semantic trajectory dataset, they propose a method that generalizes sequences of visited places based on a privacy place taxonomy.

On the other hand, in several sharing scenarios data should stay in-house to the hosting organization and the information must remain private. This is the case when a data holder is not willing or is not able due to regulations to publish the entire dataset. Assuming that at least part of the data has to become available to possibly untrusted third parties for analysis purposes, a mechanism is needed in order to ensure that no sensitive information will be released during this process. Along this direction, methodologies have been proposed for disclosure control in statistical databases [3]. These approaches support only count and/or sum queries, since no other information can be made available to the inquirer.

Gkoulalas-Divanis and Verykios [5] first described the design principles of a query engine that protects user privacy by generating fake trajectories. The idea behind [5] is that malevolent users who query the trajectory database should not be able to discover (with high confidence) any real trajectory that is returned as part of the answer set of their query, while they can use the returned data to support their analytic tasks.

In [13] and [14] authors extend and developed a privacy-aware query engine along with a benchmark framework. The proposed engine audits queries for trajectory data to block potential attacks to user privacy, supports range, distance, and *k*-nearest neighbor spatial and spatiotemporal queries, and preserves user anonymity in answers to queries by returning realistic fakes trajectories, while protecting user-specific sensitive locations.

Finally, in [17] authors proposed a data stream management system aiming at preserving users' privacy by enforcing Hippocratic principles. Limited collection, limited use and limited disclosure of data are the main privacy requirements that the system implements.

To the best of our knowledge, this is the first work that proposes a query auditing mechanism for semantically-enriched mobility data, able to provide answers while ensuring that no personal information will be disclosed to untrusted third parties.

3. Privacy Attacks

The main purpose of every attack of a malevolent is to broaden her knowledge about an individual or a situation that interests her. This occurs when the attacker raises her confidence about an event that may be related to an individual who is the 'target' or a situation for which she wishes to acquire more specific knowledge. Usually, a malevolent has prior knowledge, i.e. time, place, type of event and/or semantics (or any possible combination) about an individual.

In our setting, each query may contain one or more sub-queries (i.e. standalone spatiotemporal range queries with key words constrains within another query) . Moreover, *overlapping queries* is a sequence of at least two queries posed by a user, having as a characteristic that the criteria of these successive questions are overlapping. We assume that the queries differ only in one dimension (space / time / semantics) or in the number of the sub-queries that each one contains.

User Identification Attack. In this attack the identity of a user can be revealed by posing overlapping queries in spatial and/or temporal dimension. The attacker poses a query and if the number of trajectories is at least *k*, proceeds with one or more queries modifying each time only the same dimension such that every time the new query contains the previous one.

Let's assume that a user poses query Q_1 that contains *n* trajectories where $n \geq k$ and then Q_2 which returns as an answer $n+m$ trajectories, where $m<k$. The malevolent may conclude that the area corresponding to the difference between Q_1 and Q_2 contains *m* trajectories which is less than threshold *k*, thus privacy violation is occurred.

Consider the example depicted in Figure 1. A user poses a query Q_1: *Find people starting from area **A** between [8.00-8.30am] and, then, stop at **area B** between [9.15-11.30pm].* This query contains two different sub-queries, each one able to provide an answer if posed independently from the other. The same user poses query Q_2: *Find people starting from **area A'** between [8.00-8.30am] and, then, stop at **area B** between [9.15-11.30pm].* The answer of Q_1 contains 7 trajectories while the answer of Q_2 contains 8 trajectories. Thus, the malevolent can easily infer that only person appears in the area *A'-A*. By combining this knowledge with additional information, the malevolent can identify this person.

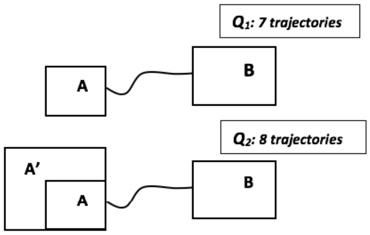

Figure 1: An example of two Overlapping Queries

In the same line, assume the following example that takes into account the semantic dimension. The user poses query Q_1 and the answer corresponds to a specific spatiotemporal area that includes 7 stop episodes. During Q_2 the malevolent maintains the criteria of Q_1 but also adds tag='work'. The output of Q_2 contains 6 trajectories. The malevolent can conclude that one entity was not working.

Sequential Tracking Attack. In this attack the user is tracked down through her trajectory by a set of focused queries. A malevolent poses a sequence of queries, which differ only in the number of the sub-queries that each one contains. Between two consecutive queries, assume that Q_1 contains *n* sub-queries and Q_2 contains $n+m$, where $m \geq 1$. To achieve an attack the malevolent should know that her target participates in *n* sub-queries. Then the malevolent may compare the number of the trajectories that answer the query consisting of *n* sub-queries in relation to $n+m$. If the difference of the number of trajectories that are participating in the answer of the two queries is less than *k*, the malevolent may conclude sensitive information about the target.

Consider the following example. The malevolent is aware of a target home and working address and the goal is to learn if the target slept at her home. Assume that $k=4$. The user poses query Q_1 with a sub-query that for sure contains the target (how many people stayed during the night in area *A* and during the day were working in area *B*). The number of trajectories that fulfil Q_1 is 6. Q_2 contains the same sub-queries with Q_1 along with a new sub-query that asks for those that returned after work back to area *A* during the night. If the answer of Q_2 returns 5 trajectories, then the malevolent may assume that the target did not spend the night at home. If the result again was 6, then she would be certain that the target returned to her home. However, by posing two queries her confidence was increased.

4. Attack Prevention

To prevent the previously described attacks, an auditing mechanism is required:

- to ensure that *k*-anonymity principle is not violated before answering each query;
- to protect sensitive episodes that include sensitive information about entities and should not be disclosed to the attackers;
- to properly modify the original query if *k*-anonymity principle is violated, so as to make it acceptable;
- to allow the data owner to have knowledge about the extent of the data leakage by examining the history of user queries to the database.

4.1 Sensitive Episodes

Sensitive episodes correspond to known locations that contain particularly sensitive information and can expose the identity of a user. We call such locations *sensitive* for a user as no information should be disclosed to the attackers. In order to deal with user-defined sensitive episodes, the auditing mechanism initially does not include the sensitive episodes as part of the answer set of the query. If the number of non-sensitive episodes exceeds threshold *k* then sensitive episodes are incorporated in the answer set.

4.2 *Zoum-Out* Algorithm

When a user poses a query to the database, she is willing to gain knowledge about whether there are semantic trajectories that are answering the query w.r.t. some criteria. If the number of the semantic trajectories composing the result set are less than the anonymity threshold *k*, the query is not safe to be answered. This ensures a first level of privacy protection.

The main idea of the proposed approach is that instead of not providing an answer when *k*-anonymity principle is violated, an auditing mechanism should try to answer the query posed by a user in any case. In other words, the mechanism will provide an answer of the most 'similar' query to the original that fulfills *k*-anonymity principle by relaxing conditions via generalization. Query relaxation enlarges the search range to include additional information. The output of this process is like a generalized query in one or more possible dimensions. Put differently, a user seeks to query an area but the mechanism resolves it for a zoomed-out area that is generalized up to a permissible degree of analysis.

The main goal of implementing such an approach is to increase user friendliness and improve database functionality. A user can gain information without posing consecutive queries expanding the criteria set until an answer is provided. To achieve this the mechanism allows the generalization of one or more criteria, of the sub-queries that constitute the original query. The generalization may occur in the spatial, the temporal or even the semantic dimension.

Let's assume that the answer of query *Q* consists of less than *k* trajectories. The output of the algorithm is a modified query *Q'*. If the modification process is successful, then the execution of query *Q'* will result at least *k* trajectories. The sets of subqueries in *Q* and *Q'* are of the same size. Each sub-query of *Q'* is either the same or generalized w.r.t. the corresponding sub-query contained in *Q*.

An obvious approach would be to apply the aforementioned method only in case where a query cannot be answered marginally w.r.t. to *k*-anonymity threshold. A threshold should then be required based on which the mechanism would be activated every time that the query could not be answered. In such a case where the mechanism is activated only if few trajectories are missing from the answer set, privacy breach may occur. A malevolent user can easily assume that the modified query does contain certain number of additional trajectories within the returned extra area. An obvious solution is to apply *Zoom-Out* algorithm regardless of the number of trajectories that are needed to reach *k*-anonymity principle.

The goal of the algorithm is to modify one or more sub-queries to provide an answer to the user. To enable the algorithm to decide which episode is preferable to be included in the answer of the modified query, the algorithm should be able to compare the distortion (the specific definition of a distortion metric is orthogonal to our approach) that is caused on each sub-query when trying to include two or more candidate episodes. A unit that calculates the distortion caused due to the generalization of one or more dimensions on one or more sub-queries is required. The distortion should be as low as possible to maintain the information that the user required when posing the original query.

Zoom-Out algorithm takes as input a semantic trajectory database, the original query posed by a user that cannot be answered, anonymity threshold *k* and a matrix *H*. The output of the algorithm is the modified query along with the corresponding sub-queries.

The algorithm after the initialization process (lines 1-2) continues with a loop phase where each sub-query of the original query is executed individually and the trajectories that comprise each one are retrieved (line 6). Each trajectory id of these trajectories is inserted into a matrix *(H)* along with the frequency indicating its appearance *(freq)* in all sub-queries (line 7). Thus, *H* is a tuple containing trajectory id (*traj_id*), frequency (*freq*) and the sub-queries (SQ_i). The maximum value that the counter *(freq)* can take in each record is equal to the number of the sub-queries. Consider as an example a query with three sub-queries, the counter for each trajectory in matrix *H* will receive a value ranging from 1 up to 3. If the counter receives the maximum value, the episodes of this trajectory are identified in all sub-queries, thus this trajectory is returned as an answer to the overall query.

Based on matrix *H*, the algorithm detects the trajectory or trajectories with frequency (i.e., number of sub-queries) less than the maximum possible frequency but at the same time with the highest value among the other trajectories in the matrix (line 9). Subsequently, a loop starts that ensures that if no episode is found, the algorithm will search the trajectory that has the subsequent smaller frequency. This loop ends either when permissible episodes can be integrated, or if all the remaining trajectories from the matrix have been investigated and no episode is found (line 15). To define the most appropriate candidate episodes, the algorithm employs a process called *Compute_Distortion_Units*. A metric function is used that calculates the distortion caused in a sub-query in order to be modified so as include an episode from the trajectory. In case we have a distortion unit greater than a distortion limit (user-defined), the episode takes the tag INF and the algorithm proceeds with the next trajectory. Under these conditions the algorithm selects as preferable the episode that has the lowest distortion unit value (line 13).

Algorithm 1. Zoom-Out

```
Input: (1) anonymity threshold k, (2) initial query with sub-queries Q
= <SQ_1, SQ_2,..., SQ_n>, (3) a semantic trajectory database D, (4)
distortion limit dist, (5) array H[tr_id, freq, SQ_1, SQ_2, ..., SQ_n]
Output: Q'= <SQ'_1, SQ'_2,..., SQ'_n>
1.  Q' ← Q; H ← ∅;
2.  N_tr ← Count(Q')
3.  repeat
4.      Something_Changed ← False;
5.      for i=1 to n do
6.          Execute_Query(in SQ'_i out tr_ids)
7.          Fill_Help_Table(in H, tr_ids out H)
8.      end for
9.      Find_Freq_Position(in H, n out i)
10.     episode_found ← False
11.     repeat
12.         Compute_Distortion_Units(in out episode_found, H, i)
13.         Select_best_candidate_episode(in H, dist, i out tr_id,
    ep_id)
14.         i ← i+1
15.     until episode_found or EOF
16.     if episode_found then
17.         Embed_New_Episode(in H, tr_id, ep_id, in out SQ'_i,
    Something_Changed)
18.         N_tr ← Count(Q')
19. until (not Something_Changed) or (N_tr=k)
20. Compute_Random_Number(in Rmin, Rmax out R)
21. Compute_New_Episodes(in Q', R out Q')
22. return Q'
```

As a next step, the sub-query is modified in one or more dimensions to contain the episode that minimizes the distortion (line 17). The repetition ends (line 19) either if *k* trajectories have frequency equal to the number of sub-queries or if no episodes were integrated.

During the generalization process of the sub-queries, a privacy breach may occur. Let's assume that the spatial dimension of the area that the query covers is enlarged so as to contain exactly *k* episodes. The spatial generalization should be the minimum possible in order to keep the distortion caused from this process as low as possible. To achieve this, most of the episodes that are added will appear in the borders of the modified area. The malevolent user thus will be more confident that between the query posed and the modified query will be at least one episode. In order to avoid such a violation, the modified query is expanded on each side by a randomly generated percentage *R* (line 21). Finally, we get as output the final modified query along with its sub-queries (line 22).

4.3 Query Auditing

The main goal of the *Query-Auditing* algorithm is to prevent any privacy violation that may occur. The input of the algorithm is the anonymity threshold *k*, the initial query posed by the user along with the corresponding sub-queries, a semantic trajectory database *D* and the *id* of the user posing the query. The algorithm first executes query *Q* and gets the number of trajectories that belong to the answer set (line 1). Then, the episodes that are considered as sensitive are defined and removed from the answer set (line 2).

Algorithm 2. Query Auditing Algorithm

```
Input: (1) anonymity threshold $k$, (2) initial query with sub-queries, $Q = <SQ_1, SQ_2,…, SQ_n>$, (3) a semantic trajectory database, $D$, (4) user id, $u_{id}$
Output: $F_Q$
1.  $F_Q \leftarrow$ Execute Q
2.  Find all sensitive episodes, remove them from $F_Q$
3.  if $||F_Q||<k$ then
4.     Q ← Zoom-Out (k, Q, D, dist, H[tr_id, freq, $SQ_1$, $SQ_2$, …, $SQ_n$])
5.     if Q = Q then return false
6.  else
7.     for each $SQ_i \in Q$ do
8.        for each $Q_j \in D$ where user_id=$u_{id}$ do
9.           for each $SQ_{jm} \in Q_j$ do
10.             if $SQ_i$ overlaps $SQ_{jm}$ then
11.                if $\left|\|SQ_i\| - \|SQ_{jm}\|\right| \geq k$ then
12.                   $Q_D = Q_D$ U Create_dummy_query ($SQ_i$ - $SQ_{jm}$)
13.                else
14.                   return false
15.          end for
16.       end for
17.    end for
18. Add sensitive episodes to $F_Q$
19. $Q_D \leftarrow Q_D$ U Q
20. return $F_Q$
```

If the number of episodes is less than *k*, *Zoom-Out* algorithm, previously described, is called to modify the original query and try to provide an answer (line 4). If *Zoom-Out* algorithm is not able to modify the query w.r.t. a distortion threshold, no answer is provided to the user and the algorithm ends (lines 5). Contrary, if the original query *Q* or the modified query *F_Q* have equal or more than *k* episodes, the auditing mechanism continues to further investigate the query based on user's history. Note that in this approach, we assume that sub-queries are totally overlapping.

Approaches that have been proposed so far, do not provide any answer when two queries posed by the same user are overlapping to prevent any privacy violation. To increase user friendliness and system functionality, we argue that the previous approach is very conservative and the algorithm should proceed to further examination before denying an answer. Let's assume that *k*=3, a user poses query Q_1: A→B→C→D which is satisfied by 7 semantic trajectories. The same user poses Q_2: A→B and the answer contains 4 trajectories. Since the difference of these two queries that corresponds to query C→D is equal to *k,* no privacy violation can be caused. However, the auditor even though has not directly reply to the query C→D, the information has been inferred. To prevent future violation, the algorithm generates a fake query that is stored in the database and corresponds to the difference of the two queries Q_1-Q_2.

The auditor proceeds by comparing every sub-query of *Q* with all the sub-queries that belong to queries posed from the user in the past. Every time two sub-queries are overlapping, the auditor checks if the number of trajectories belonging to the difference of the corresponding sub-queries is equal or greater than *k* (lines 7-11). If so, a dummy query corresponding to the difference is created (line 12). Otherwise, the algorithm ends and no answer is provided to the user (line 14). Finally, if the query is executed, the sensitive episodes are added to the answer, the query is stored in the database and the answer is returned to the user (lines 18-20).

5. CONCLUSIONS

In this paper, we proposed an envisioned query engine able to provide safe answers to queries posed by users in semantic trajectory databases. Different types of privacy attacks have been addressed and an effective auditing mechanism able to prevent privacy braches have been proposed. Finally, *Zoom-Out* algorithm is able to modify an initially not acceptable query to the closest one that can be safely answered, thus increasing the user friendliness of the engine. As a future work, we plan to finalize the implementation of the proposed query engine and testbed its utility.

6. ACKNOWLEDGMENTS

The publication of this paper has been partly supported by the University of Piraeus Research Center.

7. REFERENCES

[1] Abul, O., Bonchi, F., Nanni, M. (2008). Never walk alone: Uncertainty for anonymity in moving objects databases. *In Proc. of International Conference on Data Engineering*, *ICDE*, pp. 376-385.

[2] Abul, O., Bonchi, F., Nanni, M. (2010). Anonymization of moving objects databases by clustering and perturbation. *Information Systems*, *35*(8), pp. 884-910.

[3] Adam, N.R. and Worthmann, J. C., (1989). Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515-556.

[4] Chow, C. Y., Mokbel, M. F. (2011). Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, *13*(1), pp. 19-29.

[5] Gkoulalas-Divanis, A. and Verykios, V. S., (2008) A privacy–aware trajectory tracking query engine. *SIGKDD Explorations*, 10(1):40-49.

[6] Hoh, B., Gruteser, M. (2005). Protecting location privacy through path confusion. *In Proc. of SecureComm,* pp. 194-205.

[7] Kopanaki, D., Theodossopoulos, V., Pelekis, N., Kopanakis, I., Theodoridis, Y., (2016). Who Cares about Others' Privacy: Personalized Anonymization of Moving Object Trajectories. *In Proc. of the 19th International Conference on Extending Database Technology*, 425-436.

[8] Mahdavifar, S., Abadi, M., Kahani, M., Mahdikhani, H. (2012). A clustering-based approach for personalized privacy preserving publication of moving object trajectory data. *In Proc. of Network and System Security*, pp. 149-165.

[9] Monreale, A., Andrienko, G. L., Andrienko, N. V., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S. (2010). Movement data anonymity through generalization. *Transactions on Data Privacy*, *3*(2), pp. 91-121.

[10] Monreale, A., Trasarti, R., Pedreschi, D., Renso, C., Bogorny, V., (2011). C-safety: a framework for the anonymization of semantic trajectories. *Transactions on Data Privacy*, 4(2), pp.73-101.

[11] Nergiz, M. E., Atzori, M., Saygin, Y. (2008). Towards trajectory anonymization: a generalization-based approach. *In Proc. of the ACM International Workshop on Security and Privacy in GIS and LBS, SIGSPATIAL* pp. 52-61.

[12] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkoulalas-

Divanis, A., Macedo, J., Pelekis, N. and Theodoridis, Y., (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, *45*(4), p.42.

[13] Pelekis, N., Gkoulalas-Divanis, A., Vodas, M., Kopanaki, D., Theodoridis, Y., (2011). Privacy-aware querying over sensitive trajectory data. *In Proc. of the 20th ACM international conference on Information and knowledge management*, pp. 895-904.

[14] Pelekis, N., Gkoulalas-Divanis, A., Vodas, M., Plemenos, A., Kopanaki, D., Theodoridis, Y., (2012). Private-HERMES: a benchmark framework for privacy-preserving mobility data querying and mining methods. *In Proc. of the 15th International Conference on Extending Database Technology*, pp. 598-601.

[15] Sweeney, L (2002) *k*-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge Based Systems*, 10(5), pp. 557-570.

[16] Terrovitis, M., Mamoulis, N. (2008). Privacy preservation in the publication of trajectories. *In Proc. of International Conference on Mobile Data Management MDM*, pp. 65-72.

[17] Wu, H., Xiang, S., Ng, W.S., Wu, W. and Xue, M., (2014). HipStream: A Privacy-Preserving System for Managing Mobility Data Streams. In *2014 IEEE 15th International Conference on Mobile Data Management,* pp. 360-363.