Universitat Oberta de Catalunya

uoc.edu

Citation for published version

Romero-Tris, C. & Megías, D. (2016). User-centric Privacy-Preserving Collection and Analysis of Trajectory Data. Lecture Notes in Computer Science, 9481, 245-253.

DOI

https://doi.org/10.1007/978-3-319-29883-2_17

Document Version

This is the Accepted Manuscript version.
The version in the Universitat Oberta de Catalunya institutional repository, O2 may differ from the final published version.

Copyright and Reuse

This manuscript version is made available under the terms of the Creative Commons Attribution Non Commercial No Derivatives licence (CC-BY-NC-ND) http://creativecommons.org/licenses/by-nc-nd/3.0/es/, which permits others to download it and share it with others as long as they credit you, but they can't change it in any way or use them commercially.

Enquiries

If you believe this document infringes copyright, please contact the Research Team at: repositori@uoc.edu

Universitat Oberta de Catalunya

Research archive

User-centric privacy-preserving collection and analysis of trajectory data

Cristina Romero-Tris and David Megías

Estudis d'Informàtica Multimèdia i Telecomunicació
Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC)
Parc Mediterrani de la Tecnologia. Av. Carl Friedrich Gauss, 5,
E-08018 Castelldefels (Barcelona), Catalonia, Spain
E-mail: {cromerotr, dmegias}@uoc.edu

Abstract. Due to the increasing use of location-aware devices such as smartphones, there is a large amount of available trajectory data whose improper use or publication can threaten users' privacy. Since trajectory information contains personal mobility data, it may reveal sensitive details like habits of behavior, religious beliefs, and sexual preferences. Current solutions focus on anonymizing data before its publication. Nevertheless, we argue that this approach gives the user no control about the information she shares. For this reason, we propose a novel approach that works inside users' mobile devices, where users can decide and configure the quantity and accuracy of shared data.

1 Introduction

Over the last few years, location-aware technologies such as global positioning system (GPS) or location-based services (LBS) have caused the amount of data related to trajectories to significantly increase. On the one hand, mining and analyzing these spatio-temporal trajectory datasets can provide a valuable service (e.g., inferring traffic congestion, tracking infections, etc.). On the other hand, trajectory data often contain information about individuals. Knowledge of mobility data, in some cases combined with quasi-identifiers (gender, age, postal code, etc.), may reveal sensitive data which can threaten privacy (e.g., information about home addresses, lifestyle, religious beliefs, ideology, etc.).

To cope with this problem, there is an emergent field of the literature that focuses on proposing new solutions. For example, Abul et al. [1] propose the (k, δ)-anonymity model, which modifies a location polyline to be represented by a single cylinder of radius δ. Then, k trajectories co-localized inside the same cylinder are indistinguishable from each other. Terrovitis et al. [2] propose an algorithm that suppresses the existence of certain points in the trajectories. The challenge in this case is how to find the optimal set points to delete, with the minimum possible information loss. The authors propose a greedy heuristic that assumes that all the adversarial knowledge is known before data publication. Similarly, Pensa et al. [3] propose to remove frequent sequential patterns. They transform sequences by adding, deleting, or substituting some points of

the trajectory. Yarovoy et al. [4] employ the Hilbert curve [5] in order to map a multi-dimensional space to one dimension. The purpose of this is finding the nearest neighbors at every point of the trajectory. Then, the neighbors are used to create anonymization groups to generalize trajectory data of each member.

All these works are limited to privacy protection on already collected data. The proposed algorithms work on the server side, before its publication. Nevertheless, we argue that trajectory anonymization would rather be performed a step earlier, in the user side. This protects users from an adversary that gains access to the records stored in the database. Moreover, the advantage of this approach is that users are able to configure the quantity and accuracy of shared information before it is stored in the database.

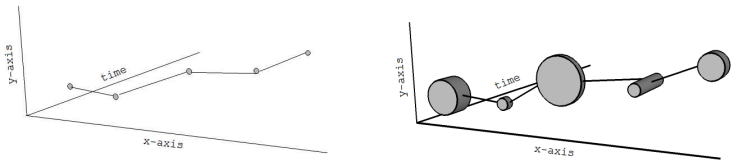
For this purpose, our system relies on a personalized trajectory anonymization method that transforms spatio-temporal points into uncertain points, where the exact location and timing are distorted according to a set of user-defined parameters. Then, users are grouped to execute a protocol and obtain k-anonymity, being k a user-defined parameter according to her privacy requirements.

This paper is organized as follows: Section 2 defines relevant concepts for our system. Section 3 describes our proposal in detail. Privacy is analyzed in Section 4, and Section 5 concludes the paper.

2 Problem definition

This section describes some background tools or concepts that are necessary to understand our system.

Definition 1 (Trajectory). *A trajectory T of length $|T|$ is an ordered list of spatio-temporal points (x_1, y_1, t_1), (x_2, y_2, t_2), ..., $(x_{|T|}, y_{|T|}, t_{|T|})$ where (x_i, y_i, t_i) means that the user was at a physical location with Cartesian coordinates (x_i, y_i) at instant t_i. During the time segment $[t_i, t_{i+1}]$ the user is assumed to move along a straight line from (x_i, y_i) to (x_{i+1}, y_{i+1}). Fig. 1(a) represents the definition of a trajectory with five points. The three-dimentional space represents the time and the Cartesian coordinates of the position (abscissae and ordinates).*



(a) Trajectory (b) Anonymized trajectory

Fig. 1. Schematic representation of a trajectory before and after anonymization

Definition 2 (Uncertain point). *For a specific spatio-temporal point (x_i, y_i, t_i), its anonymized version is another vector $(cx_i, cy_i, r_i, a_i, b_i)$ where (cx_i, cy_i) are the Cartesian coordinates of the center of a circle of radius r_i that contains (x_i, y_i), and $[a_i, b_i]$ is a time interval that contains t_i.*

Definition 3 (Anonymized trajectory). *An anonymized trajectory T' of length $|T'|$ is an ordered list of uncertain point vectors $(cx_1, cy_1, r_1, a_1, b_1)$, $(cx_2, cy_2, r_2, a_2, b_2)$, ..., $(cx_{|T'|}, cy_{|T'|}, r_{|T'|}, a_{|T'|}, b_{|T'|})$. During the time between $[a_i, b_i]$ and $[a_{i+1}, b_{i+1}]$, the user is assumed to move along a line from any point inside the circle defined by (cx_i, cy_i, r_i) to any point inside $(cx_{i+1}, cy_{i+1}, r_{i+1})$.*

Fig. 1(b) represents the trajectory of Fig. 1(a) after being anonymized. The anonymization transforms a point into a circle of variable radius, and an instant into a time interval. Thus, for each spatio-temporal point, a cylinder is obtained.

Definition 4 (Anonymized sub-trajectory). *Given an anonymized trajectory T', an anonymized sub-trajectory s' of size $|s'| \leq |T'|$ is an ordered subset of the vectors composing T'. The conditions to be fulfilled are: (1) in order not to be a single point, the size of the sub-trajectory must be $|s'| > 1$; and (2) the order of the vectors in s' must be the same as in T'.*

Definition 5 (Similar anonymized sub-trajectories). *Having initially two anonymized trajectories: s'=$(cx_{11}, cy_{11}, r_{11}, a_{11}, b_{11})$, $(cx_{21}, cy_{21}, r_{21}, a_{21}, b_{21})$, ..., $(cx_{|s'|}, cy_{|s'|}, r_{|s'|}, a_{|s'|}, b_{|s'|})$, and s''=$(cx_{12}, cy_{12}, r_{12}, a_{12}, b_{12})$, $(cx_{22}, cy_{22}, r_{22}, a_{22}, b_{22})$, ..., $(cx_{|s''|}, cy_{|s''|}, r_{|s''|}, a_{|s''|}, b_{|s''|})$, we define two system parameters θ_L and θ_T that represent the maximum distance to consider two points similar in terms of location and time, respectively. Then, we consider that s' and s'' are similar if these conditions are fulfilled:*

1. $|s'|$=$|s''|$
2. *For $i = 1, 2, \ldots, |s'|$:*
   (a) $\sqrt{(cx_{i1} - cx_{i2})^2 + (cy_{i1} - cy_{i2})^2} < (r_{i1} + r_{i2} + \theta_L)$. *This means that the Euclidean distance betweenboth circles is lower that θ_L.*
   (b) $(((b_{i2} + \theta_T) > a_{i1})$ *and* $((b_{i2} + \theta_T) < b_{i1}))$ *or* $(((b_{i1} + \theta_T) > a_{i2})$ *and* $((b_{i1} + \theta_T) < b_{i2}))$. *This means that the time intervals are separated less than θ_T occurring $|s''|$ before $|s'|$ or $|s'|$ before $|s''|$.*

3 Protocol description

This section describes the proposed system. We assume that User U_i's device already contains her trajectory T (x_1, y_1, t_1), (x_2, y_2, t_2), ..., $(x_{|T|}, y_{|T|}, t_{|T|})$; and that a server $\boldsymbol{\mathcal{S}}$ requests the trajectory information.

Regarding cryptography, users employ a n-out-of-n threshold ElGamal encryption [6], where n users share a public key y and the corresponding unknown private key α is divided into n shares α_i. Using this protocol, a certain message m can be encrypted with the public key y and it can only be decrypted if all n users collaborate in the process.

3.1 Creation of the anonymization group

The process starts when the server $\boldsymbol{\mathcal{S}}$ sends a request to collect trajectory data from users. Then, users that are willing to share their information send a confirmation to the server. Let N be the total number of users who send a confirmation.

The users who want to participate in the process must be included in groups of size n, where n is a predetermined system parameter. In order to prevent $\boldsymbol{\mathcal{S}}$ from grouping users as it wishes, a join coin-tossing protocol adapted from [7] is executed. This protocol assumes that every user U_i already has a personal public key (pk_i) provided by a PKI. The protocol employs two random oracles (which in practice can be computed as pseudo-random functions [8]) $H_1 = 0,1^* \rightarrow 0,1^k$ (where k is the bit-length of the public key), and $H_2 = 0,1^* \rightarrow 0,1^{N \cdot \log N}$. The following steps are executed:

1. Every user U_i generates a random r_i and sends $H_1(IP_i, pk_i, r_i)$ to $\boldsymbol{\mathcal{S}}$, where IP_i is a concatenation of the public and private IP address of U_i.
2. U_i waits a short predefined time.
3. $\boldsymbol{\mathcal{S}}$ sends $H_1(IP_i, pk_i, r_i)$ for $i = 1, \ldots, N$ to all the users.
4. Then, each user U_i computes $h = H_2(H_1(IP_i, pk_i, r_i), \ldots, H_1(IP_N, pk_N, r_N)$ and divides the result h into chunks of size $\log N$, denoted $h_1, \ldots, h_N$.
5. User U_i takes h_i as her identifier.
6. Grouping is carried out by taking groups of n parties according to the sorting. That is, for $i = 1, \ldots, \lfloor N/n \rfloor$, the ith group is formed by users with identifiers $(h_{n \cdot (i-1)+1}, \ldots, h_{n \cdot i})$
7. $\boldsymbol{\mathcal{S}}$ sends the IP addresses of each user to the members of her group.
8. The members of each group send each other their IP addresses, public key and the random r_i they used at the beginning of the protocol.
9. Each group member computes $H_1(IP_j, pk_j, r_j)$ for every user U_j in her group, and verifies that it matches what she received from $\boldsymbol{\mathcal{S}}$. Additionally, she computes H_2 as in Step 4 to verify that all the IP addresses assigned to her group are inside it. If any verification fails, she sends *abort* to the group members and exits the system.

3.2 Trajectory anonymization

In this phase, U_i decides the granularity of spatio-temporal disclosure for every point of T. This means that, for every point (x_i, y_i, t_i), the user will obtain a vector $(cx_i, cy_i, r_i, a_i, b_i)$ based on the values that she chooses for:

- *The radius r_i.* This parameter, expressed in kilometers, is the radius of the circle that contains the Cartesian coordinates (x_i, y_i). A larger radius means higher generalization and hence, higher distortion. Based on the value chosen by the user, we randomly select a point (cx_i, cy_i) that fulfills the equation $(x_i - cx_i)^2 + (y_i - cy_i)^2 \leq r_i^2$.
- *The time gap γ_i.* This parameter, expressed in hours (but working with real numbers), indicates the time difference between a_i and b_i. Therefore, to obtain these values we randomly choose a value v between 0 and γ_i. Then, we compute $a_i = t_i - v$, and $b_i = t_i + \gamma_i - v$.

Repeating this process for all the points (x_1, y_1, t_1), (x_2, y_2, t_2), …, $(x_{|T|}$, $y_{|T|}, t_{|T|})$ in T, we obtain the anonymized trajectory $|T'|$=$(cx_1, cy_1, r_1, a_1, b_1)$, $(cx_2, cy_2, r_2, a_2, b_2)$, …, $(cx_{|T'|}, cy_{|T'|}, r_{|T'|}, a_{|T'|}, b_{|T'|})$.

Note that the user can also completely remove a spatio-temporal point from the list. Therefore, $|T|$ and $|T'|$ might not be equal.

3.3 Sub-trajectory extraction

The sub-trajectory extraction depends on two parameters. The first one is the number of sub-trajectories to extract (τ), and the second one is the maximum number of points that each sub-trajectory should contain, μ. Having τ, μ as system parameters, Algorithm 1 shows how to extract the sub-trajectories:

Algorithm 1 Sub-trajectory extraction algorithm

procedure SUB-TRAJECTORY EXTRACTION
Input: τ, μ, anonymized trajectory $T'[]$ as a table of spatio-temporal points.
Output: Table *subtraj* of anonymized sub-trajectories
subtraj:=new table[τ]
count:=0
Loop: i:=0 to τ by 1
    Loop: j:=0 to $\tau/|T'|$ by 1
        subtraj[i] := T'[*count*]
        count++
    Loop-end: j
Loop-end: i
Loop: i:=0 to τ by 1
    While (size of *subtraj*[*i*] > μ)
        Remove one random element from *subtraj*[*i*]
    While-end
Loop-end: i
end procedure

3.4 Fake sub-trajectory generation

Similarly to the real sub-trajectory extraction, the fake sub-trajectory generation needs two parameters: (1) the number of fake sub-trajectories to generate (τ'); and (2) the maximum number of points that each fake sub-trajectory should contain, μ'. There are many works in the literature that describe how to generate a fake trajectory. The generation of a particular algorithm for this is out the scope of this paper. For our purposes, we employ the method proposed in [9].

3.5 Distribution of sub-trajectories

In this phase, the real and fake sub-trajectories are distributed among the group of users $\{U_1, \ldots, U_n\}$. In order to prevent one malicious member of the group from learning all the sub-trajectories that belong to another user, the group executes a multi-party privacy-preserving protocol composed by three phases: group key generation, anonymous sub-trajectories retrieval, and query submission.

Group key generation

1. Users $\{U_1, \ldots, U_n\}$ generate a large prime p where $p = 2q + 1$ and q is a prime too. Next, they pick an element $g \in \mathbb{Z}_q^*$ of order q.
2. In order to generate the group key, each user U_i performs the following steps:

(a) Generates a random number $a_i \in \mathbb{Z}_q^*$.
(b) Calculates her own share $y_i = g^{a_i} \bmod p$.
(c) Broadcasts a commitment $h_i = \mathcal{H}(y_i)$, where $\mathcal{H}$ is a one-way function.
(d) Broadcasts y_i to the other members of the group.
(e) Checks that $h_j = \mathcal{H}(y_j)$ for $j = (1, \ldots, n)$.
(f) Calculates the group key using the received shares: $y = \prod_{1 \leq j \leq n} y_j = g^{a_1} \cdot g^{a_2} \cdot \ldots \cdot g^{a_n}$

3.6 Anonymous sub-trajectory retrieval

Assuming that each user U_i has $(\tau + \tau')$ sub-trajectories: $s_{i1}, s_{i2}, \ldots, s_{i(\tau+\tau')}$

1. User U_i encrypts real and fake sub-trajectories as plaintext. For each s_{ij}, U_i generates a random number r_{ij} and encrypts s_{ij} with y: $c_{ij}^0 = E_y(s_{ij}, r_{ij}) = (g^{r_{ij}}, s_{ij} \cdot y^{r_{ij}}) = (c1_{ij}, c2_{ij})$.
2. For $i = (2, \ldots, n)$, $j = (1, \ldots, (\tau + \tau'))$ each user U_i sends c_{ij}^0 to the first member of the group (U_1).
3. For $i = (1, \ldots, n-1)$, each user U_i performs the following operations:
   (a) Receives the list of ciphertexts $\left\{c_{11}^{i-1}, c_{12}^{i-1}, \ldots, c_{n(\tau+\tau')}^{i-1}\right\}$.
   (b) Using her share of the group key, partially decrypts the list of ciphertexts using the algorithm described in [6]. The resulting list of ciphertexts is denoted as $\left\{c_{11}^{i-1\prime}, \ldots, c_{n(\tau+\tau')}^{i-1}{}'\right\}$.
   (c) The list of ciphertexts $\left\{c_{11}^{i-1\prime}, \ldots, c_{n(\tau+\tau')}^{i-1}{}'\right\}$ is re-masked using the re-masking algorithm described in [10] with a key $y' = \prod_{w=i+1}^{n} g^{\alpha_w}$. As a result, U_i obtains a re-encrypted version $\left\{e_{11}^{i-1}, \ldots, e_{n(\tau+\tau')}^{i-1}\right\}$.
   (d) Permutes the ciphertexts at random, obtaining $\left\{e_{\sigma(11)}^{i-1}, \ldots, e_{\sigma(n(\tau+\tau'))}^{i-1}\right\}$
   (e) Sends $\left\{c_{11}^{i}, \ldots, c_{n(\tau+\tau')}^{i}\right\} = \left\{e_{\sigma(11)}^{i-1}, \ldots, e_{\sigma(n(\tau+\tau'))}^{i-1}\right\}$ to U_{i+1}.
4. The last user U_n performs the following operations:
   (a) Receives the list of ciphertexts $\left\{c_{11}^{i-1}, \ldots, c_{n(\tau+\tau')}^{i-1}\right\}$.
   (b) Using her share of the group key, partially decrypts the list of ciphertexts using the algorithm described in [6]. At this point, U_n owns the sub-trajectories cleartexts, so she broadcast them to $\{U_1, \ldots, U_{n-1}\}$.

This is the central part of the protocol which has a higher cost and complexity. In this phase, each user performs $(\tau + \tau')$ encryptions, and $n \cdot (\tau + \tau')$ decryptions. Regarding the number of messages, each user U_i sends one long message (containing $n \cdot (\tau + \tau')$ ciphertexts) to U_{i+1}, except for the last user U_n, who sends $n - 1$ short messages (containing each one $n \cdot (\tau + \tau')$ cleartexts).

3.7 Sub-trajectory submission and retrieval

1. Each group member U_i must send $(\tau + \tau')$ sub-trajectories to the server $\boldsymbol{\mathcal{S}}$. More specifically, from the received list, user U_i submits the sub-trajectories found between positions $i \cdot n$ and $i \cdot n + \tau + \tau'$.

2. Upon receiving the $(\tau + \tau')$ answers from the server, each user broadcasts them to the rest of the group members. Then, each user takes the answers that corresponds to her original sub-trajectories.
3. The answer of the server for each sub-trajectory is ϕ, the number of sub-trajectories in the database similar to the one submitted according to Definition 5. Sub-trajectories where $\phi < k$ must be removed from the anonymized trajectory, and hence, they are put in a list L to be used in next step.

3.8 Anonymized trajectory trimming

Using Algorithm 2 the list L of real sub-trajectories is removed from the anonymized trajectory T' of each user. The resulting anonymized trajectory is sent to the server $\boldsymbol{\mathcal{S}}$. The server can store it in its database for future analysis or publication.

Algorithm 2 Anonymized trajectory trimming algorithm

```
procedure ANONYMIZED TRAJECTORY TRIMMING
Input: table of sub-trajectories to be removed L[], anonymized trajectory T'[]
Output: Resulting anonymized trajectory T'
ls :=size of L
Loop: i := 0 to ls by 1
    For every spatio-temporal vector q in L_i
        Remove q from T'
    Loop-end: i
end procedure
```

4 Privacy analysis

In this section, we analyze the system in terms of privacy. First of all, the ElGamal cryptosystem is semantically secure under the Decisional Diffie-Hellman assumption. This means that a dishonest user cannot know if two different ciphertexts will result into the same cleartext after decryption.

Therefore, every time that a ciphertext c_i is transformed by a group member (i.e., remasked and permuted), the attacker can only link the result to c_i by random guessing, the intermediate re-maskings and permutations preventing her from finding the links between them. Hence, the probability of success is $1/(n(\tau + \tau'))$, since there are $n(\tau + \tau')$ ciphertexts involved in the process.

The proposed protocol also relies on the server to help users achieve k-anonymity by answering their requests. Moreover, the server is in charge of creating the groups. The steps presented in Section 3.1 adapted from [7] prevent the server from maliciously grouping users. The security of this protocol is analyzed in [7]. The authors compute the probability of a bad grouping, i.e., having $n-1$ dishonest users together with a single honest party. Assuming that $N >> t$, the authors state that this probability is approximately $(\frac{t}{N})^{n-2} \cdot N$. For example, if one million users participate in the system, and the server controls one thousand, then the probability of a bad grouping is under 10^{-48}.

5 Conclusions and future work

In this paper, we argue that trajectory data would rather be protected in the client-side, before they are stored in the server or disclosed to a third entity. To the best of our knowledge, this is the first work that introduces trajectory anonymization in the user's device, giving users control over the information they send to the server and providing k-anonymity.

However, our work is on an early stage of development and there are some interesting open research problems that need to be addressed in the future. More specifically, experimental results are necessary in order to know how the system behaves for different parameter configurations. In order to do this, we need to implement the system and execute it in a real or simulated environment.

Acknowledgments

This work was partly funded by the Spanish Government through grants TIN2011-27076-C03-02 "CO-PRIVACY" and TIN2014-57364-C2-2-R "SMARTGLACIS".

References

1. O. Abul, F. Bonchi, M. Nanni, Never walk alone: Uncertainty for anonymity in moving objects databases, in: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ICDE '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 376–385.
2. M. Terrovitis, N. Mamoulis, Privacy preservation in the publication of trajectories, in: Mobile Data Management, 2008. MDM'08. 9th International Conference on, IEEE, 2008, pp. 65–72.
3. R. G. Pensa, A. Monreale, F. Pinelli, D. Pedreschi, Pattern-preserving k-anonymization of sequences and its application to mobility data mining, in: PiLBA, 2008, pp. 1–10.
4. R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, W. H. Wang, Anonymizing moving objects: How to hide a mob in a crowd?, in: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '09, ACM, New York, NY, USA, 2009, pp. 72–83.
5. D. Hilbert, Ueber die stetige abbildung einer line auf ein flächenstück, Mathematische Annalen 38 (3) (1891) 459–460.
6. Y. Desmedt, Y. Frankel, Threshold cryptosystems, in: L. N. in Computer Science (Ed.), Advances in Cryptology – CRYPTO'89, Vol. 335, 1990, pp. 307–315.
7. Y. Lindell, E. Waisbard, Private web search with malicious adversaries, in: Privacy Enhancing Technologies, Springer, 2010, pp. 220–235.
8. I. Berman, I. Haitner, From non-adaptive to adaptive pseudorandom functions, in: Theory of Cryptography, Springer, 2012, pp. 357–368.
9. A. Gkoulalas-Divanis, V. S. Verykios, A privacy-aware trajectory tracking query engine, ACM SIGKDD Explorations Newsletter 10 (1) (2008) 40–49.
10. M. Abe, Mix-networks on permutation networks, in: L. N. in Computer Science (Ed.), Advances in Cryptology – Asiacrypt'99, Vol. 1716, 1999, pp. 258–273.