IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 6- Number 2- Spring 2014 (19-28)

TrPLS: Preserving Privacy in Trajectory Data Publishing by Personalized Local Suppression

Elahe Ghasemi Komishani
Department of Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
e.ghasemi@modares.ac.ir

Mahdi Abadi
Department of Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
abadi@modares.ac.ir

Received: January 16, 2013-Accepted: January 15, 2014

***Abstract*—Trajectory data are becoming more popular due to the rapid development of mobile devices and the widespread use of location-based services. They often provide useful information that can be used for data mining tasks. However, a trajectory database may contain sensitive attributes, such as disease, job, and salary, which are associated with trajectory data. Hence, improper publishing of the trajectory database can put the privacy of moving objects at risk. Removing identifiers from the trajectory database before the public release, is not effective against privacy attacks, especially, when an adversary uses some partial trajectory information as its background knowledge. The existing approaches for preserving privacy in trajectory data publishing apply the same amount of privacy protection for all moving objects without considering their privacy requirements. The consequence is that some moving objects with high privacy requirements may be offered low privacy protection, and vice versa. In this paper, we address this challenge and present TrPLS, a novel approach for preserving privacy in trajectory data publishing. It combines local suppression with the concept of personalization to achieve the conflicting goals of data utility and data privacy in accordance with the privacy requirements of moving objects. The results of experiments on a trajectory dataset show that TrPLS can be successfully used for preserving personalized privacy in trajectory data publishing.**

Keywords-trajectory data; privacy preservation; personalized privacy; quasi-identifier; local suppression; information loss; disclosure risk

I. INTRODUCTION

In recent years, with the proliferation of location-aware devices, such as active RFID tags and GPS equipped mobile phones, it is easy to track the location of moving objects over a period of time and generate a collection of spatio-temporal data, also known as *trajectory data* or *moving object data*. These data have been made available in various domains [1].

Real-life applications, such as Geo-marketing, intelligent transportation systems, city traffic planning, location-based advertising, and many more can benefit from trajectory data mining. Trajectory data often contain detailed information about moving objects, and for many applications, these data need to be published with sensitive attributes, such as disease, job, and salary, incurring the concern of breaching moving objects' privacy.

Example 1. A hospital uses an RFID tagging system for the care of its patients, in which patient information are stored in a central trajectory database. The hospital wants to publicly release the trajectory database for data mining tasks. Each trajectory data record is represented as a tuple (ID, Trajectory, Disease), in which "Trajectory" is a sequence of spatio-temporal pairs. For example, the trajectory data record (#7, ⟨b4 → a6 → c7⟩, SARS) indicates that the patient with ID#7 has visited locations b, a, and *c* at timestamps 4, 6, and 7, respectively and has SARS.

With enough background knowledge, an adversary can launch three types of privacy attacks on a trajectory database:

Identity linkage attack: If a trajectory in the trajectory database is very specific, such that not many moving objects can match it, the adversary using some background knowledge may uniquely identify the trajectory data record of the target victim and, therefore, its sensitive attribute values [2-4].

Attribute linkage attack: If a sensitive attribute value occurs frequently with some sub-trajectories, the adversary may identify it from these sub-trajectories even though cannot uniquely identify the trajectory data record of the target victim [2-4].

Similarity attack: If some sensitive attribute values that are distinct but semantically similar occur frequently with some sub-trajectories, the adversary may infer sensitive information from these sub-trajectories even though cannot uniquely identify the trajectory data record of the target victim.

Many approaches have been proposed for preserving privacy in trajectory data publishing [1-13], but most of them do not consider different privacy requirements of different moving objects, resulting in the increasing risk of information loss and privacy breach. Moreover, the majority of them are not resistant to all three identity linkage, attribute linkage, and similarity attacks. To tackle these shortcomings, we present TrPLS, a novel approach that combines local suppression with the concept of personalization for preserving privacy in trajectory data publishing. In general, we can apply local or global suppression on trajectory data records. Global suppression eliminates a moving point from all trajectory data records in the trajectory database, if it makes the privacy breach probability of some trajectory data records so high, while local suppression eliminates the moving point only from trajectory data records with high privacy breach probability and leaves others intact. Hence, local suppression preserves better data utility in comparison with global suppression. As a result, TrPLS apply it on trajectory data records. We show TrPLS is resistant to all aforementioned attacks and evaluate its performance in terms of trajectory information loss and disclosure risk.

The rest of the paper is organized as follows: Section II briefly reviews some related work. Section III gives basic definitions. Section IV presents TrPLS and Section V reports experimental results. Finally, Section VI sums up the discussion and draws the conclusions.

II. Related Work

The methods for preserving privacy in trajectory data publishing can be divided into two categories [1]: (1) clustering based methods that apply the concept of k-anonymity in relational databases and (2) quasi-identifier based methods that assume an adversary uses some partial knowledge of a trajectory as a quasi-identifier to identify its remaining moving points or sensitive attributes.

A. *Clustering*

Abul *et al.* [5] introduce the concept of (k, δ)-anonymity for preserving privacy in trajectory data publishing, which exploits the inherent uncertainty of locations in order to reduce the amount of distortion needed to anonymize trajectory data. Furthermore, they present a method, called NWA, to achieve (k, δ)-anonymity. The method first partitions the trajectory data into equivalence classes with respect to time span and produces a set of clusters, each having a number of trajectories in the interval $[k, 2k-1]$. It then transforms each cluster by means of space translation such that the translation distortion is minimum and all trajectories could be placed in a cylindrical volume of radius $\delta/2$.

Nergiz *et al.* [6] adopt the notion of k-anonymity to trajectories and propose a clustering-based approach for trajectory data anonymization. Moreover, they show that releasing anonymized trajectories may lead to some privacy breaches, and therefore present a randomization based reconstruction algorithm for releasing anonymized trajectory data.

Monreale *et al.* [7] present a method for the anonymization of trajectory data combining the notions of spatial generalization and k-anonymity. The main idea is to anonymize trajectories by replacing exact locations by approximate ones. To do this, they first construct a suitable tessellation of the geographical area into sub-areas and then apply a spatial generalization to the original trajectory data. They further transform the generalized trajectory data to ensure that it satisfies the notion of k-anonymity.

Mahdavifar *et al.* [8] propose a greedy clustering-based approach in which trajectories are anonymized to some extent proportional to the privacy requirements of their moving objects. They first assign a privacy level to each trajectory and then partition trajectories into a set of clusters based on a trajectory similarity criterion. Each cluster is created such that its size is proportional to the maximum privacy level of trajectories within it. They finally anonymize trajectories of each cluster and generate a set of anonymized trajectories containing linked and distorted moving points. Although this approach aims at preserving personalized privacy in trajectory data publishing, but it is not resistance to both attribute linkage and similarity attacks.

B. *Quasi-identifier*

Terrovitis *et al.* [9] assume that an adversary uses some partial trajectory information as its background knowledge to infer unknown moving points. Hence, they iteratively eliminate selected moving points from the original trajectory data until a privacy constraint is satisfied.

Yarovoy *et al.* [10] introduce a notion of k-anonymity by defining an attack graph associated with the original trajectory data and its distorted one. They consider timestamps as the quasi-identifiers and present two different algorithms, namely extreme-union and symmetric-anonymization, to build anonymization groups that provably satisfy the k-anonymity requirement.

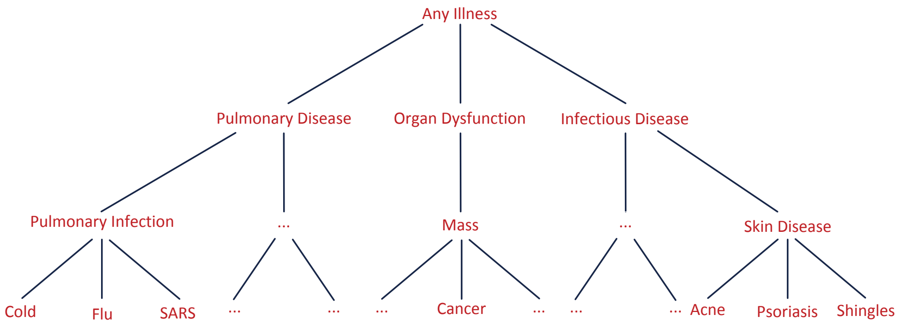

Figure 1. A taxonomy tree for the sensitive attribute Disease

Fung *et al.* [2] adopt a privacy model called LKC-privacy and develop an anonymization framework that employs global suppression to achieve LKC-privacy. The general intuition is to ensure that each sub-trajectory with maximum length L in a trajectory database is shared by at least K trajectory data records and the confidence of inferring any sensitive attribute value is not greater than C. Chen *et al.* [4] present a similar framework that supports both local and global suppressions. The aim is to preserve instances of moving points and frequent sub-trajectories in a trajectory data. These frameworks take into consideration both identity linkage and attribute linkage attacks, but are not resistant to the similarity attack.

Most of the aforementioned methods do not consider different privacy requirements of moving objects. Moreover, the majority of them are not resistant to all three identity linkage, attribute linkage, and similarity attacks.

III. Basic Definitions and Notations

A. *Trajectory Database*

A typical location-aware system generates a sequence of spatio-temporal data records of the general form (id, l, t), each of which indicates that a moving object having the unique identifier id was detected in the location l at time t. For example, in transportation systems, it represents that a passenger with the transportation card number id was present in the station l at time t.

Definition 1 (Trajectory). Let O be a set of moving objects. The trajectory of a moving object $o_i \in O$ is denoted by τ_i and is a sequence of spatio-temporal pairs:

$$\tau_i = \langle (l_i^1, t_i^1), \cdots, (l_i^m, t_i^m) \rangle \ , \qquad (1)$$

where each $(l_i^k, t_i^k) \in \tau_i$ is called a *moving point* and is denoted by p_i^k.

The length of τ_i, denoted by $|\tau_i|$, is defined as the number of its moving points. A trajectory that contains only the first $k \leq |\tau_i|$ moving points of τ_i is denoted by τ_i^k. We define a strict total order relation, $\prec$, between each two moving points $p_i^k = (l_i^k, t_i^k)$ and $p_i^{k} \ = (l_i^{k} \ , t_i^{k} \ )$ in τ_i:

$$p_i^k \prec p_i^{k'} \text{ iff } t_i^k < t_i^{k'} \ . \qquad (2)$$

Definition 2 (Joinable Trajectories). Two trajectories $\tau_i = \langle (l_i^1, t_i^1), \cdots, (l_i^m, t_i^m) \rangle$ and $\tau_j = \langle (l_j^1, t_j^1), \cdots, (l_j^m, t_j^m) \rangle$ are said to be joinable iff $\tau_i^{m-1} = \tau_j^{m-1}$ and $t_i^m < t_j^m$. The joined trajectory is denoted by $\tau_i \bowtie \tau_j$:

$$\tau_i \bowtie \tau_j = \langle (l_i^1, t_i^1), \cdots, (l_i^m, t_i^m), (l_j^m, t_j^m) \rangle \ . \qquad (3)$$

Definition 3 (Sub-trajectory). Let $\tau_i = \langle p_i^1, \cdots, p_i^m \rangle$ and $\tau_j = \langle p_j^1, \cdots, p_j^s \rangle$ be two trajectories. τ_j is said to be a sub-trajectory of τ_i and is denoted by $\tau_j \sqsubseteq \tau_i$, if there exist integers $1 \leq k_1 < \cdots < k_s \leq m$ such that

$$p_j^1 = p_i^{k_1}, p_j^2 = p_i^{k_2}, \cdots, p_j^s = p_i^{k_s} \ . \qquad (4)$$

A trajectory database may contain other attributes that are associated with the trajectory data. These attributes are divided into two categories: *sensitive* and *insensitive*. If moving objects of a location-aware system are patients, sensitive attribute(s) may be their disease. Formally, a trajectory database contains a set of *trajectory data records* in the form of

$$r_i = \langle p_i^1, \cdots, p_i^m \rangle : s_i^1, \cdots, s_i^n : a_i^1, \cdots, a_i^q \ , \qquad (5)$$

where $\langle p_i^1, \cdots, p_i^m \rangle$ is the trajectory, s_i^1 to s_i^n are the sensitive attribute values, and a_i^1 to a_i^q are the insensitive attributes values of a moving object. The trajectory in r_i is denoted by $\tau(r_i)$:

$$\tau(r_i) = \langle p_i^1, \cdots, p_i^m \rangle \ . \qquad (6)$$

The values of each sensitive attribute are usually divided into different categories. We can use a taxonomy tree to represent these values and their categories. To illustrate the concept, Fig. 1 shows a simple taxonomy tree for the sensitive attribute Disease that organizes all diseases as its leaves. Each internal node has been uniquely labeled with a name showing the category of diseases in the node's sub-tree.

In the rest of the paper, for simplicity, we assume that the trajectory database contains only a sensitive attribute and each moving object corresponds to only one trajectory data record. In this case, the sensitive attribute value of each trajectory data record r_i is denoted by $s(r_i)$.

Definition 4 (Taxonomy Tree). Let S be the set of sensitive attribute values. A taxonomy tree for this attribute is a tuple $\Gamma = (V, E, \ell)$, where V and E are the set of nodes and edges, respectively. $\ell: V \rightarrow 2^S$ is a labeling function that assigns a subset of sensitive attribute values to each node in V. There are two types of nodes: internal and leaf nodes. It is assumed that the depth of all leaf nodes is the same.

Let $\ell(v_j)$ be the subset of sensitive attribute values assigned to a node $v_j \in V$, the size of $\ell(v_j)$ is denoted by $|\ell(v_j)|$ and is called the *cardinality* of v_j. It should be noted that the cardinality of all leaf nodes in Γ is always equal to one.

Definition 5 (Node Level). The level of each node $v_j \in V$ in Γ is denoted by $\iota(v_j)$ and is defined as the length of the shortest path from this node to one of the leaf nodes in Γ.

Definition 6 (Covering Node). A node $v_j \in V$ in Γ is strictly covered by a node $v_k \in V$ iff $\ell(v_j) \subset \ell(v_k)$. In this case, v_k is called a *covering node* of v_j. The set of all covering nodes for v_j is denoted by $c(v_j)$.

Example 2. Consider the taxonomy tree in Fig. 1. The level of the leaf nodes, *e.g.*, Cold, Flu, and SARS, is *zero*. The level of the internal nodes Pulmonary Infection, Mass, and Skin Disease is *one* and the level of the internal nodes Pulmonary Disease, Organ Dysfunction, and Infectious Disease is *two*. Also, Pulmonary Infection and Pulmonary Disease are covering nodes for Cold, Flu, and SARS.

Definition 7 (Parent Node). A node $v_k \in V$ in Γ is called the *parent* of a node $v_j \in V$ and is denoted by $\mathscr{p}(v_j)$, iff it is a covering node for v_j and $\iota(v_k) = \iota(v_j) + 1$.

Definition 8 (Spanning Node). Let $s(r_i)$ be the sensitive attribute value of a trajectory data record r_i. A node $v_j \in V$ in Γ is called a *spanning node* for r_i iff $s(r_i) \in \ell(v_j)$.

Definition 9 (Minimal Spanning Node). A node $v_j \in V$ in Γ is called a *minimal spanning node* for a trajectory data record r_i and is denoted by $\mathscr{s}(r_i)$, iff it is a spanning node for r_i and its level, $\iota(v_j)$, is equal to zero.

B. Privacy Level

Different moving objects may have different privacy requirements. Therefore, we assign a privacy level to each moving object to represent its privacy requirements. Let T be a trajectory database and $L = \{\vartheta_0, \dots, \vartheta_{rt-1}\}$ be a totally ordered set of privacy levels, where rt is the level of the root node of Γ. We define $\rho: T \to O$ to be a total function that assigns each trajectory data record in T to a moving object in O and define $\theta: O \to L \cup \{\Upsilon\}$ to be a total function that assigns each moving object in O to a privacy level in $L \cup \{\Upsilon\}$. Therefore, a privacy level is assigned to each trajectory data record in T. It should be noted that if a moving object does not need any privacy protection, its privacy level is defined to be equal to Υ.

Definition 10 (Guarding Node). A node $v_j \in V$ in Γ is called a *guarding node* for a trajectory data record r_i in T and is denoted by $v_j = \mathscr{g}(r_i)$, iff it is a spanning node for r_i and $\theta(\rho(r_i)) = \iota(v_j)$.

Example 3. Consider the trajectory database in Table I. Each trajectory data record has one of three privacy levels Low, Medium, or High, which are respectively equal to three node levels *zero*, *one*, or *two* of the taxonomy tree in Fig. 1. Also, one of trajectory data records does not need any privacy protection that is denoted by None. The sensitive attribute value and privacy level of the trajectory data record r_2 are Cancer and Medium, respectively. Therefore, the node Mass is a guarding node for it.

TABLE I. A TRAJECTORY DATABASE

ID	Privacy Level	Trajectory	Disease
1	Low	$a1 \to b4 \to e5 \to c7$	Flu
2	Medium	$d1 \to b3 \to c7$	Cancer
3	None	$a1 \to b4 \to a6 \to c7$	Cold
4	High	$a2 \to b4 \to e5 \to a6 \to f8$	Cancer
5	Low	$b4 \to a6$	Shingles
6	Medium	$d1 \to a2 \to c7$	Psoriasis
7	Low	$b4 \to a6 \to c7$	SARS

C. Privacy Attacks

Suppose a trajectory database T is to be published for data mining. Explicit identifiers, *e.g.*, name and ID, have been removed. One recipient, the adversary, employing some background knowledge and one of privacy attacks may be able to identify the trajectory data record or sensitive attribute value of a victim in T.

Let r_i be the trajectory data record of a victim $\rho(r_i) \in O$. The adversary's background knowledge about this victim, denoted by ξ_i, contains at most δ moving points:

$$\xi_i = \langle p_i^1, \cdots, p_i^l \rangle,\ l \le \delta\ , \qquad (7)$$

where δ is the maximum length of the adversary's background knowledge.

Using ξ_i, the adversary can identify a set $T(\xi_i)$ of trajectory data records in T matching ξ_i:

$$T(\xi_i) = \{r_k \in T \mid \xi_i \sqsubseteq \tau(r_k)\}\ . \qquad (8)$$

Note that a trajectory data record $r_k \in T$ matches ξ_i iff ξ_i is a sub-trajectory of its trajectory $\tau(r_k)$. For example, in Table I, if $\xi_i = \langle b4, c7 \rangle$, then $T(\xi_i) = \{r_1, r_3, r_7\}$. The adversary can identify and utilize $T(\xi_i)$ to launch three types of privacy attacks: identity linkage, attribute linkage, and similarity attacks.

Definition 11 (Identity Linkage Attack). Given a trajectory data record $r_i \in T$ and a background knowledge $\xi_i \sqsubseteq \tau(r_i)$, if the size of $T(\xi_i)$, denoted by $|T(\xi_i)|$, is small, then the adversary may identify r_i and, therefore, $s(r_i)$.

Definition 12 (Attribute Linkage Attack). Given a trajectory data record $r_i \in T$ and a background knowledge $\xi_i \sqsubseteq \tau(r_i)$, the adversary may identify $s(r_i)$ with confidence $P_c(s(r_i)|\xi_i)$:

$$P_c(s(r_i)|\xi_i) = |T(s(r_i)) \cap T(\xi_i)| / |T(\xi_i)|\ , \qquad (9)$$

where $T(s(r_i))$ is the set of trajectory data records in T that their sensitive attribute value is equal to $s(r_i)$. In fact, $P_c(s(r_i)|\xi_i)$ is the percentage of the trajectory data records in $T(\xi_i)$ containing $s(r_i)$. The privacy of $\rho(r_i)$ is at risk if $P_c(s(r_i)|\xi_i) > \sigma$, where σ is a parameter specifying the amount of privacy disclosure and is called the *privacy breach threshold*.

Definition 13 (Similarity Attack). Given a trajectory data record $r_i \in T$ and a background knowledge $\xi_i \sqsubseteq \tau(r_i)$, the adversary may identify $\mathscr{g}(r_i)$ iff $s(r_k) \in \ell(\mathscr{g}(r_i))$ for all $r_k \in T(\xi_i)$.

Definition 14 (Critical Trajectory). Given a background knowledge ξ_i, a non-empty sub-trajectory

$\tau_j \sqsubseteq \xi_i$ is called *critical* iff the adversary can successfully perform one of the identity linkage, attribute linkage, or similarity attacks using it.

The privacy of the moving object $\rho(r_i)$ is breached when the adversary can associate this object with one of the sensitive attribute values in the set $\ell(\mathscr{g}(r_i))$, where $\mathscr{g}(r_i)$ is the guarding node of r_i.

Definition 15 (Privacy Breach Probability). The probability of privacy breach for $\rho(r_i)$ assuming ξ_i is calculated as

$$P_b(\rho(r_i)|\xi_i) = \frac{1}{|T(\xi_i)|} \sum_{r_k \in T(\xi_i)} P(\mathscr{g}(r_i)|s(r_k)) \ , \quad (10)$$

where

$$P(\mathscr{g}(r_i)|s(r_k)) = \begin{cases} 1 & s(r_k) \in \ell(\mathscr{g}(r_i)) \ , \\ 0 & \text{otherwise} \ . \end{cases} \quad (11)$$

Definition 16 (Critical Trajectory Data Record). A trajectory data record $r_i \in T$ is critical iff $P_b(\rho(r_i)|\xi_i)$ is greater than σ, where σ is the privacy breach threshold.

TABLE II. AN ANONYMIZED TRAJECTORY DATABASE

ID	Privacy Level	Trajectory	Disease
1	Low	$a1 \to b4 \to c7$	Flu
2	Medium	$d1 \to c7$	Cancer
3	None	$a1 \to b4 \to a6 \to c7$	Cold
4	High	$b4 \to a6$	Cancer
5	Low	$b4 \to a6$	Shingles
6	Medium	$d1 \to c7$	Psoriasis
7	Low	$b4 \to a6 \to c7$	SARS

IV. PRESERVING PRIVACY BY PERSONALIZED LOCAL SUPPRESSION

In this section, we present TrPLS, an approach that combines local suppression with the concept of personalization for preserving privacy in trajectory data publishing. TrPLS first applies the algorithm STR to make a set $\mathcal{A}^\delta$ of all sub-trajectories with a given maximum length δ. It then gives $\mathcal{A}^\delta$ as input to the algorithm MPSTD, which identifies critical trajectory data records and eliminates a number of moving points from them such that there is no critical trajectory data record in the anonymized trajectory database and the amount of information loss is minimized.

Fig. 2 shows the pseudo-code of STR. It takes a trajectory database T and the background knowledge threshold δ as input and returns a set $\mathcal{A}^\delta$ of all sub-trajectories with the maximum length δ as output. Let $\mathcal{A}_i$ be an ordered set of sub-trajectories of length i. The algorithm first initializes $\mathcal{A}^\delta$ to the empty set and $\mathcal{A}_1$ to the set of all sub-trajectories of length one (Lines 1–2). It then computes a subset $T(\tau_j) \subseteq T$ for each sub-trajectory $\tau_j \in \mathcal{A}_1$ and adds τ_j to $\mathcal{A}^\delta$ (Lines 3–6). Also, in each iteration of nested loops (Lines 10–18), if each two sub-trajectories $\tau_j, \tau_k \in \mathcal{A}_i$ are joinable and the intersection $T(\tau_j)$ and $T(\tau_k)$ is non-empty, it adds the joined sub-trajectory $\tau_j \bowtie \tau_k$ to $\mathcal{A}_{i+1}$ and $\mathcal{A}^\delta$ (Lines 12–16). The above steps are repeated until i is greater than δ or $\mathcal{A}_i$ is empty (Lines 8–20).

Example 4. The sets $\mathcal{A}_1$ and $\mathcal{A}_2$ of sub-trajectories of length one and two in Table I are as follows:

$$\mathcal{A}_1 = \{a1, d1, a2, b3, b4, e5, a6, c7, f8\}$$

$$\mathcal{A}_2 = \{a1b4, a1e5, a1a6, a1c7, d1a2, d1b3, d1e5, d1c7, a2b4, a2e5, a2a6, a2c7, a2f8, b3c7, b4e5, b4a6, b4c7, b4f8, e5a6, e5c7, e5f8, a6c7, a6f8\}$$

```
Algorithm STR
input:
  $T$: Trajectory database;
  $\delta$: Background knowledge threshold;
output:
  $\mathcal{A}^\delta$: Set of sub-trajectories;

1.  $\mathcal{A}^\delta := \emptyset$;
2.  $\mathcal{A}_1 := \{\tau_j \mid \tau_j \sqsubseteq \tau(r_k)$ for some $r_k \in T \wedge |\tau_j| = 1\}$;
3.  for each $\tau_j \in \mathcal{A}_1$ do
4.    Compute $T(\tau_j)$ using equation (8);
5.    $\mathcal{A}^\delta := \mathcal{A}^\delta \cup \{\tau_j\}$;
6.  end for
7.  $i := 1$;
8.  while $i \le \delta$ and $\mathcal{A}_i \ne \emptyset$ do
9.    $\mathcal{A}_{i+1} := \emptyset$;
10.   for $j := 1$ to $|\mathcal{A}_i|$ do
11.     for $k := j + 1$ to $|\mathcal{A}_i|$ do
12.       if $\tau_j^{i-1} = \tau_k^{i-1}$ and $T(\tau_j) \cap T(\tau_k) \ne \emptyset$ then
13.         $T(\tau_j \bowtie \tau_k) := T(\tau_j) \cap T(\tau_k)$;
14.         $\mathcal{A}_{i+1} := \mathcal{A}_{i+1} \cup \{\tau_j \bowtie \tau_k\}$;
15.         $\mathcal{A}^\delta := \mathcal{A}^\delta \cup \{\tau_j \bowtie \tau_k\}$;
16.       end if
17.     end for
18.   end for
19.   $i := i + 1$;
20. end while
21. return $\mathcal{A}^\delta$;
```

Figure 2. Algorithm for generating a set of sub-trajectories

Fig. 3 shows the pseudo-code of MPSTD. It takes the trajectory database T, the set $\mathcal{A}^\delta$ of all sub-trajectories with the maximum length δ, the background knowledge threshold δ, and the privacy breach threshold σ as input and returns an anonymized trajectory database $T^\mathcal{G}$ as output. Let $\mathcal{T}_c$ be a set of critical sub-trajectories. The algorithm first initializes $\mathcal{T}_c$ to the empty set (Line 1). For each sub-trajectory $\tau_j \in \mathcal{A}^\delta$, it then finds a subset $\mathcal{B}_j$ of trajectory data records in $T(\tau_j)$ whose guarding node is not covered by the guarding node of any other trajectory data record in $T(\tau_j)$ (Line 3), makes the set $\mathcal{C}_j$ of critical trajectory data records in $\mathcal{B}_j$ (Line 4), and adds τ_j to $\mathcal{T}_c$ only if $\mathcal{C}_j$ is non-empty (Lines 5–7). It subsequently repeats the following steps until $\mathcal{T}_c$ is empty (Lines 9–29): For each sub-trajectory $\tau_j \in \mathcal{T}_c$, it computes the personalized suppression score $\psi(\tau_j, \mathcal{T}_c)$ (Lines 10–12):

$$\psi(\tau_j, \mathcal{T}_c) = \max_{p_j^k \in \tau_j} \varphi(p_j^k, \tau_j, \mathcal{T}_c) \ , \quad (12)$$

where $\varphi(p_j^k, \tau_j, \mathcal{T}_c)$ is the personalized suppression score of a moving point $p_j^k \in \tau_j$ with respect to $\mathcal{T}_c$:

$$\varphi(p_j^k, \tau_j, \mathcal{T}_c) = \begin{cases} \dfrac{|\mathcal{T}_c(p_j^k)|}{|T(\tau_j)|} \displaystyle\sum_{r_i \in T(\tau_j)} \theta(\rho(r_i)) & p_j^k \in \tau_j \ , \\ 0 & \text{otherwise} \ , \end{cases} \quad (13)$$

where $|\mathcal{T}_c(p_j^k)|$ is the number of sub-trajectories in $\mathcal{T}_c$ containing p_j^k and $|T(\tau_j)|$ is the number of trajectory data records in T matching τ_j. The algorithm then selects a sub-trajectory τ_z containing p_z^q with the maximum $\varphi(p_z^q, \tau_z, \mathcal{T}_c)$ from sub-trajectories in $\mathcal{T}_c$ (Lines 13–14) and makes the set $\mathcal{C}_z$ of critical trajectory data records in $T(\tau_z)$ (Line 15). It next selects r_i with the maximum privacy level $\theta(\rho(r_i))$ from critical trajectory data records in $\mathcal{C}_z$ and adds r_i to the set $\mathcal{D}_z$ (Lines 18–19). It subsequently eliminates p_z^q from $\tau(r_i)$ and again makes the set $\mathcal{C}_z$ of critical trajectory data records in $T(\tau_z)$ (Lines 20–23). The above steps are repeated until $\mathcal{C}_z$ is empty (Lines 17–24). Eliminating p_z^q from trajectory data records may result in the generation of new critical sub-trajectories. Hence, it identifies these sub-trajectories using the algorithm MCST and adds them to $\mathcal{T}_c$ (Lines 25–27). It finally sets $T^{\mathcal{G}}$ to T (Line 30).

```
Algorithm MPSTD
input:
    $T$: Trajectory database;
    $\mathcal{A}^\delta$: Set of sub-trajectories;
    $\delta$: Background knowledge threshold;
    $\sigma$: Privacy breach threshold;
output:
    $T^{\mathcal{G}}$: Anonymized trajectory database;
1.  $\mathcal{T}_c := \emptyset$;
2.  for each $\tau_j \in \mathcal{A}^\delta$ do
3.      $\mathcal{B}_j := \{r_k \in T(\tau_j) \mid \ell(g(r_k)) \not\subset \ell(g(r_i))$ for all
            $r_i \in T(\tau_j)\}$;
4.      $\mathcal{C}_j := \{r_k \in \mathcal{B}_j \mid P_b(\rho(r_k)|\tau_j) > \sigma\}$;
5.      if $\mathcal{C}_j \neq \emptyset$ then
6.          $\mathcal{T}_c := \mathcal{T}_c \cup \{\tau_j\}$;
7.      end if
8.  end for
9.  while $\mathcal{T}_c \neq \emptyset$ do
10.     for each $\tau_j \in \mathcal{T}_c$ do
11.         Compute $\psi(\tau_j, \mathcal{T}_c)$ using (12);
12.     end for
13.     $\tau_z := \arg\max_{\tau_j \in \mathcal{T}_c} \psi(\tau_j, \mathcal{T}_c)$;
14.     $p_z^q := \arg\max_{p_z^k \in \tau_z} \varphi(p_z^k, \tau_z, \mathcal{T}_c)$;
15.     $\mathcal{C}_z := \{r_k \in T(\tau_z) \mid P_b(\rho(r_k)|\tau_z) > \sigma\}$;
16.     $\mathcal{D}_z := \emptyset$;
17.     while $\mathcal{C}_z \neq \emptyset$ do
18.         $r_i := \arg\max_{r_k \in \mathcal{C}_z} \theta(\rho(r_k))$;
19.         $\mathcal{D}_z := \mathcal{D}_z \cup \{r_i\}$;
20.         $T := T - \{r_i\}$;
21.         $\tau(r_i) := \tau(r_i) - \langle p_z^q \rangle$;
22.         $T := T \cup \{r_i\}$;
23.         $\mathcal{C}_z := \{r_k \in T(\tau_z) \mid P_b(\rho(r_k)|\tau_z) > \sigma\}$;
24.     end while
25.     for each $r_i \in \mathcal{D}_z$ do
26.         $\mathcal{T}_c := \mathcal{T}_c \cup mcst(T, \tau(r_i), p_z^q, \delta, \sigma)$;
27.     end for
28.     $\mathcal{T}_c := \mathcal{T}_c - \{\tau_z\}$;
29. end while
30. $T^{\mathcal{G}} := T$;
31. return $T^{\mathcal{G}}$;
```

Figure 3. Algorithm of personalized local suppression

Eliminating a moving point from a trajectory data record by personalized local suppression may generate new critical sub-trajectories. Identifying all of these critical sub-trajectories requires expensive computational cost. An intuitive way to identify new critical sub-trajectories is to recall MPSTD. However, it is very costly. Instead, we apply the algorithm MCST to reduce the computational cost of identifying all new critical sub-trajectories. It significantly restricts the whole space of sub-trajectories to a very small set of sub-trajectories that are affected by personalized local suppression.

Fig. 4 shows the pseudo-code of MCST. It takes a trajectory database T, a trajectory τ_z, a moving point p_z^q, the background knowledge threshold δ, and the privacy breach threshold σ as input and returns a set $\mathcal{T}_c$ of new critical sub-trajectories as output. The algorithm first initializes $\mathcal{T}_c$ to the empty set and $\mathcal{A}_1$ to the set $\{\langle p_z^q \rangle\}$ (Lines 1–2). Then, for each sub-trajectory $\tau_j \in \mathcal{A}_i$, if τ_j is a critical sub-trajectory, it adds τ_j to $\mathcal{T}_c$ (Lines 5–9). Finally, it makes $\mathcal{A}_{i+1}$ from sub-trajectories $\tau_j \sqsubseteq \tau_z$ of length $i+1$ containing moving point p_z^q (Line 10). The above steps are repeated until i is greater than δ or $\mathcal{A}_i$ is empty (Lines 4–12).

```
Algorithm MCST
input:
    $T$: Trajectory database;
    $\tau_z$: Trajectory;
    $p_z^q$: Moving point;
    $\delta$: Background knowledge threshold;
    $\sigma$: Privacy breach threshold;
output:
    $\mathcal{T}_c$: Set of sub-trajectories;

1.  $\mathcal{T}_c := \emptyset$;
2.  $\mathcal{A}_1 := \{\langle p_z^q \rangle\}$;
3.  $i := 1$;
4.  while $i \leq \delta$ and $\mathcal{A}_i \neq \emptyset$ do
5.      for each $\tau_j \in \mathcal{A}_i$ do
6.          if $(P_b(\rho(r_k)|\tau_j) > \sigma)$ for some $r_k \in T(\tau_j)$ then
7.              $\mathcal{T}_c := \mathcal{T}_c \cup \{\tau_j\}$;
8.          end if
9.      end for
10.     $\mathcal{A}_{i+1} := \{\tau_j \sqsubseteq \tau_z \mid p_z^q \in \tau_j \wedge |\tau_j| = i+1\}$;
11.     $i := i + 1$;
12. end while
13. return $\mathcal{T}_c$;
```

Figure 4. Algorithm for identifying new critical sub-trajectories

Example 5. Consider the trajectory database in Table I with $\delta = 2$ and $\sigma = 0.50$. After eliminating the moving point $e5$ from the trajectory data record r_1, we only need to check the sub-trajectories $\langle e5 \rangle$, $\langle a1, e5 \rangle$, $\langle b4, e5 \rangle$, and $\langle e5, c7 \rangle$ for identifying new critical sub-trajectories. Since both $\langle e5 \rangle$ and $\langle b4, e5 \rangle$ are critical, $\mathcal{T}_c$ becomes $\{\langle e5 \rangle, \langle b4, e5 \rangle\}$.

Theorem 1. The anonymized trajectory database $T^{\mathcal{G}}$ is resistant to all three identity linkage, attribute linkage, and similarity attacks.

PROOF. Let $r_i \in T^{\mathcal{G}}$ be a trajectory data record. Since $T^{\mathcal{G}}$ has been made anonymous, therefore, $P_b(\rho(r_i)|\xi_i) \leq \sigma$ for all background knowledge ξ_i

with the maximum length δ. Therefore, according to Definition (15), the adversary cannot correctly identify $g(r_i)$, and subsequently, $s(r_i)$ with confidence greater than σ, even though the size of $T(\xi_i)$ is small. Thus, we conclude that $T^{\mathcal{G}}$ is resistant to all three identity linkage, attribute linkage, and similarity attacks.

A. Complexity Analysis

As mentioned earlier, TrPLS eliminates moving points from critical trajectory data records with respect to the privacy level of their moving objects. Given a trajectory database T, it first applies the algorithm STR to generate a set $\mathcal{A}^{\delta}$ of all sub-trajectories with the maximum length δ, where δ is the background knowledge threshold. The number of sub-trajectories in $\mathcal{A}$ is $O(n^{\delta})$, where n is the number of distinct moving points in T. Hence, the worst-case time complexity of STR is $O(n \cdot |T| + n^{\delta})$, where $|T|$ is the number of trajectory data records in T. Subsequently, it makes a set $\mathcal{T}_c$ of remaining critical sub-trajectories in T and computes the personalized suppression score of each moving point in $\mathcal{T}_c$. It next eliminates the moving point with maximum personalized suppression score from some critical trajectory data records and applies the algorithm MCST to update $\mathcal{T}_c$ with new critical sub-trajectories. In the worst case, MCST has a time complexity of $O(n^{\delta} \cdot |T|)$. Therefore, the time complexity of personalized local suppression is bounded by $O(n^{2\delta} \cdot |T|^2)$. Accordingly, we conclude that the time complexity of TrPLS is $O(n^{2\delta} \cdot |T|^2)$.

V. Experiments and Analysis

In this section, we evaluate the performance of TrPLS in terms of (1) information loss and (2) disclosure risk for moving objects with different privacy levels. We also experimentally compare its performance with that of other related work in [2-4].

A. Experimental Results

1) Trajectory Dataset: We used a trajectory dataset, called City80K [4], for the experiments. It is a dataset simulating the trajectories of 80,000 citizens in a metropolitan area with 26 city blocks in 24 hours. Each trajectory data record in the dataset contains a sensitive attribute with five possible values. We randomly assigned each trajectory data record to one of five privacy levels None, Low, Medium, High, or Very High, so that trajectory data records with lower privacy levels are more than those with higher privacy levels. We also generated a taxonomy tree of depth 6 and 108 leaf nodes. All experiments were conducted on a PC with a 2.8GHz Intel Core 2 Duo CPU and 4GB of RAM.

2) Trajectory Information Loss: The main goal of TrPLS is to maintain an anonymized trajectory database $T^{\mathcal{G}}$ as close to its original trajectory database T as possible. Hence, we evaluate the information loss for moving objects with different privacy levels due to the personalized local suppression. Let $r_i \in T$ be an original trajectory data record and $r_i^{\mathcal{G}} \in T^{\mathcal{G}}$ be its corresponding anonymized trajectory data record. The average information loss of all trajectory data records is defined as

$$\bar{\mathcal{I}}_{\tau} = \frac{1}{|T|} \sum_{r_i \in T} \frac{|\tau(r_i)| - |\tau(r_i^{\mathcal{G}})|}{|\tau(r_i)|} \,, \qquad (14)$$

where $|\cdot|$ is the length of a trajectory.

Tables III and IV show the effect of σ on $\bar{\mathcal{I}}_{\tau}$ for $\delta = 2$ and $\delta = 3$, respectively, where σ and δ are the privacy breach and background knowledge thresholds. On the whole, with decreasing δ and increasing σ, $\bar{\mathcal{I}}_{\tau}$ slightly decreases due to the decrease in the number of critical sub-trajectories resulting from the decrease in the number of critical trajectory data records.

TABLE III. Effect of σ on the Average Information Loss of Trajectory Data Records for $\delta = 2$

Privacy Level	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
None	0.0000	0.0000	0.0000	0.0000	0.0000
Low	83.8340	0.2955	0.1974	0.1738	0.1734
Medium	99.8852	21.8390	0.2578	0.2348	0.2337
High	99.9885	40.3225	1.4339	0.2340	0.2340
Very High	99.9885	40.1276	14.0198	0.2723	0.2723

TABLE IV. Effect of σ on the Average Information Loss of Trajectory Data Records for $\delta = 3$

Privacy Level	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
None	0.0000	0.0000	0.0000	0.0000	0.0000
Low	89.0524	4.2397	0.2144	0.1738	0.1734
Medium	99.9792	30.9204	1.8432	0.5165	0.3136
High	99.9996	44.8198	6.6221	0.5804	0.3325
Very High	99.9996	49.3583	18.4179	0.8618	0.3920

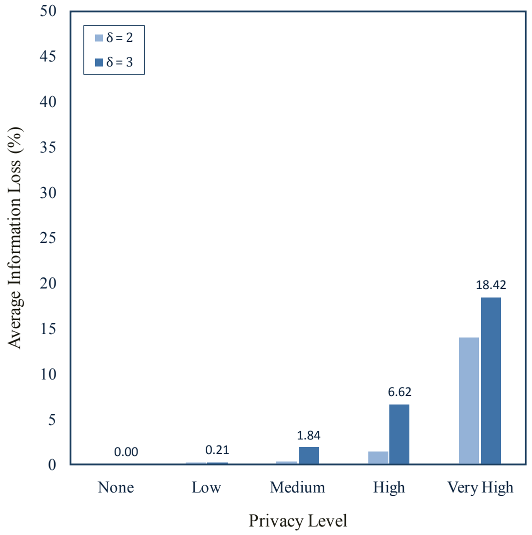

Figure 5. Effect of δ on the average information loss of trajectory data records for $\sigma = 0.4$

Fig. 5 shows the effect of δ on $\bar{\mathcal{I}}_{\tau}$ for $\sigma = 0.4$. As can be seen, trajectory data records with higher privacy levels have more information loss. Since these trajectory data records need more privacy protection, therefore, more moving points are eliminated from them by the personalized local suppression.

3) Disclosure Risk: We use the disclosure risk as a metric to measure the privacy breach probability of moving objects. Given an anonymized trajectory data record $r_i^{\mathcal{G}} \in T^{\mathcal{G}}$, let $s(r_i)$ be the sensitive attribute value of its original trajectory data record $r_i \in T$ and $\xi_i \sqsubseteq \tau(r_i)$ be the adversary's background knowledge. The probability of disclosure of $s(r_i)$ assuming ξ_i is calculated as

$$P(s(r_i)|\xi_i) = \begin{cases} \frac{1}{|T^{\mathcal{G}}(\xi_i)|} \sum_{r_k^{\mathcal{G}} \in T^{\mathcal{G}}(\xi_i)} P(s(r_i)|s(r_k^{\mathcal{G}})) & \xi_i \sqsubseteq \tau(r_i^{\mathcal{G}}) \\ 0 & \text{otherwise} \end{cases}, \quad (15)$$

where $P(s(r_i)|s(r_k^{\mathcal{G}}))$ is the probability of disclosure of $s(r_i)$ assuming the sensitive attribute value $s(r_k^{\mathcal{G}})$ of a trajectory data record $r_k^{\mathcal{G}} \in T^{\mathcal{G}}(\xi_i)$:

$$P(s(r_i))|s(r_k^{\mathcal{G}})) = \begin{cases} 1 & s(r_i) = s(r_k^{\mathcal{G}}) \\ 0 & \text{otherwise} \end{cases}. \quad (16)$$

The adversary may use any sequence of moving points with length not greater than δ as its background knowledge to perform a privacy attack. Therefore, the probability of disclosure of $s(r_i)$ should be calculated for different lengths of ξ_i.

The average disclosure risk of trajectory data records is defined as

$$\bar{\beta}_d = \frac{1}{|T^{\mathcal{G}}|} \sum_{r_i^{\mathcal{G}} \in T^{\mathcal{G}}} \frac{1}{|\mathcal{K}|} \sum_{\xi_i \in \mathcal{K}} P_d(s(r_i))|\xi_i) , \quad (17)$$

where

$$\mathcal{K} = \{\xi_i \mid \xi_i \sqsubseteq \tau(r_i) \wedge |\xi_i| \leq \delta\} . \quad (18)$$

Tables V and VI show the effect of σ on $\bar{\beta}_d$ for $\delta = 2$ and $\delta = 3$. The results suggest that with decreasing σ, the average disclosure risk decreases. This is because more moving points are eliminated from trajectory data records.

TABLE V. EFFECT OF σ ON THE AVERAGE DISCLOSURE RISK OF TRAJECTORY DATA RECORDS FOR $\delta = 2$

Privacy Level	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
None	20.37	20.18	20.23	20.23	20.23
Low	2.43	19.95	20.01	20.03	20.03
Medium	0.03	16.41	20.00	20.00	20.00
High	0.00	14.16	19.64	20.02	20.02
Very High	0.00	14.21	16.39	20.01	20.01

TABLE VI. EFFECT OF σ ON THE AVERAGE DISCLOSURE RISK OF TRAJECTORY DATA RECORDS FOR $\delta = 3$

Privacy Level	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
None	20.56	20.48	20.74	20.37	20.37
Low	1.35	19.27	20.57	20.19	20.19
Medium	0.01	14.19	19.81	20.13	20.13
High	0.00	12.49	17.82	20.12	20.12
Very High	0.00	11.43	15.38	19.43	20.13

B. Comparison

We cannot directly compare TrPLS with previous related work on privacy preserving in trajectory data publishing, because none of them consider the personalized privacy. Instead, we consider equal conditions with KCL-Global [2, 3] and KCL-Local [4], and present a new variant of TrPLS, called KCL-TrPLS. KCL-TrPLS is similar to TrPLS but with this difference that it does not use the taxonomy tree and apply the k-anonymity. However, KCL-Global, KCL-Local, and KCL-TrPLS are not resistant to the similarity attack. Note that C and L are equivalent to σ and δ in TrPLS, respectively. KCL-Global and KCL-Local use City80K [4] as the trajectory dataset and consider one of five possible values of its sensitive attribute as sensitive and the others as non-sensitive, which in KCL-TrPLS, they correspond to sensitive attribute values with the privacy levels Low and None, respectively. Therefore, approximately 80 percent of the trajectory data records in City80K do not need any privacy protection. In the following experiments, we show that KCL-TrPLS would significantly lower information loss in the context of trajectory data.

For the purpose of fair comparison, we use the same trajectory information loss metric as that defined in [4], to measure the percentage of moving points that are lost due to suppressions:

$$\mathcal{I}_\tau(T, T^{\mathcal{G}}) = \frac{N(T) - N(T^{\mathcal{G}})}{N(T)} , \quad (19)$$

where $N(T)$ and $N(T^{\mathcal{G}})$ are the numbers of moving points in the original and anonymized trajectory databases T and $T^{\mathcal{G}}$, respectively.

1) Effect of K: We vary the parameter K from 10 to 50 while fixing $C = 0.6$ and $L = 3$ (which are equivalent to taking $\sigma = 0.6$ and $\delta = 3$ in TrPLS) to compare the effect of K on KCL-Global [2, 3], KCL-Local [4], and KCL-TrPLS, the results of which are shown in Fig. 6. Clearly, KCL-TrPLS can significantly reduce the information loss for higher value of K.

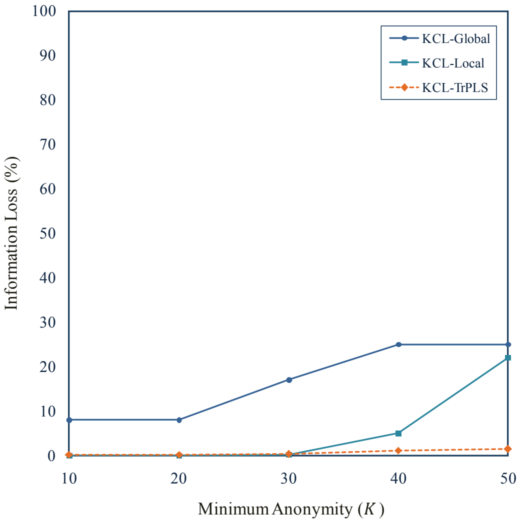

Figure 6. Effect of K on the information loss of KCL-Global, KCL-Local, and KCL-TrPLS for $C = 0.6$ and $L = 3$

2) Effect of C: Fig. 7 shows the effect of C on the information loss of KCL-Global [2, 3], KCL-Local [4], and KCL-TrPLS, while fixing $K = 30$ and $L = 3$. When C is small, the information loss is high for KCL-Global and KCL-Local. However, for different values of C, KCL-TrPLS results in substantially low information loss. As a result, KCL-TrPLS totally has low information loss. This is because it eliminates critical moving points only from critical trajectory data records, while KCL-Global [2, 3] and KCL-Local [4] may eliminate critical moving points from non-critical trajectory data records in addition to critical trajectory data records.

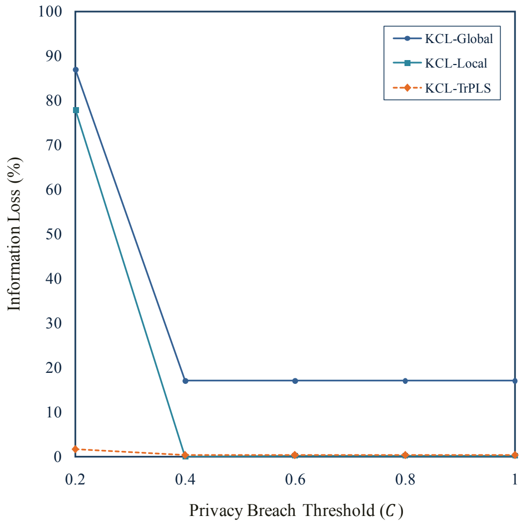

Figure 7. Effect of C on the information loss of KCL-Global, KCL-Local, and KCL-TrPLS for $K = 30$ and $L = 3$

VI. Conclusion and Discussion

In this paper, we presented TrPLS, an approach that combines local suppression with the concept of personalization for privacy preserving in trajectory data publishing. It eliminates moving points from critical trajectory data records with respect to the privacy level of their moving objects, such that there is no critical trajectory data record in the anonymized trajectory database and the amount of the information loss is minimized. We used a trajectory dataset simulating the trajectories of 80,000 citizens in a metropolitan area and evaluated the performance of TrPLS in terms of information loss and disclosure risk. We also experimentally compared its performance with that of other related work in [2-4]. The results of experiments show that TrPLS can significantly reduce the information loss. TrPLS not only is able to provide personalized privacy preserving in trajectory data publishing, but also it is resistant to all three identity linkage, attribute linkage, and similarity attacks.

Acknowledgment

This work was supported by ICT Research Institute (ITRC) under contract number 500/19230 and identification code 91-02-03.

References

[1] F. Bonchi, “Privacy preserving publication of moving object data,” in *Privacy in Location-Based Applications*, C. Bettini, S. Jajodia, P. Samarati, X. S. Wang (Eds.), LNCS, vol. 5599, pp. 190–215, Springer-Verlag, Heidelberg, August 2009.

[2] B. C. M. Fung, M. Cao, B. C. Desai, and H. Xu. “Privacy protection for RFID data,” in *Proceedings of the 24th Annual ACM Symposium on Applied Computing*, Honolulu, HI, USA, March 2009, pp. 1528–1535.

[3] N. Mohammed, B. C. M. Fung, and M. Debbabi, “Preserving privacy and utility in RFID data publishing,” Technical Report 6850, Concordia University, Montreal, Canada, September 2010.

[4] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, “Privacy-preserving trajectory data publishing by local suppression,” *Information Sciences*, vol. 231, pp. 83–97, May 2013.

[5] O. Abul, F. Bonchi, and M. Nanni, “Never walk alone: Uncertainty for anonymity in moving objects databases,” in *Proceedings of the 24th IEEE International Conference on Data Engineering*, Cancun, Mexico, April 2008, pp. 376–385.

[6] M. E. Nergiz, A. Maurizio, Y. Saygin, and B. Güç, “Towards trajectory anonymization: A generalization-based approach,” *Transaction on Data Privacy*, vol. 2, no. 1, pp. 47–75, April 2009.

[7] A. Monreale, G. Andrienko, N. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel, “Movement data anonymity through generalization,” *Transactions on Data Privacy*, vol. 3, no. 2, pp. 91–121, August 2010.

[8] S. Mahdavifar, M. Abadi, M. Kahani, and H. Mahdikhani, “A clustering-based approach for personalized privacy preserving publication of moving object trajectory data,” in *Network and System Security*, L. Xu, E. Bertino, and Y. Mu (Eds.), LNCS, vol. 7645, pp. 149–165, Springer-Verlag, Heidelberg, November 2012.

[9] M. Terrovitis and N. Mamoulis, “Privacy preservation in the publication of trajectories,” in *Proceedings of the 9th IEEE International Conference on Mobile Data Management*, Beijing, China, April 2008, pp. 65–72.

[10] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and 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*, Vancouver, Canada, March 2009, pp. 72–83.

[11] G. Gidófalvi, X. Huang, and T. B. Pedersen, “Privacy-preserving data mining on moving object trajectories,” in *Proceedings of the IEEE International Conference on Mobile Data Management*, Mannheim, Germany, 2007, pp. 60–68.

[12] E. Kaplan, T. B. Pedersen, E. Savaş, and Y. Saygın, “Discovering private trajectories using background information,” *Data & Knowledge Engineering*, vol. 69, no. 7, pp. 723–736, July 2010.

[13] O. Abul, F. Bonchi, and M. Nanni, “Anonymization of moving objects databases by clustering and perturbation,” *Information Systems*, vol. 35, no. 8, pp. 884–910, December 2010.

Elahe Ghasemi Komishani received her B.Sc. degree in computer engineering from University of Mazandaran in 2010 and the M.Sc. degree in computer engineering from Tarbiat Modares University in 2013. Her main research interests are network security, privacy preserving, and data mining. She did her thesis on personalized privacy preserving publication of trajectory data.

Mahdi Abadi received his B.Sc. and M.Sc. degrees in computer engineering from Ferdowsi University of Mashhad in 1998 and Tarbiat Modares University in 2001, respectively. He also received his Ph.D. degree from Tarbiat Modares University in 2008, where he worked on the network vulnerability analysis. Since 2009, he has been an assistant professor in the Department of Electrical and Computer Engineering at Tarbiat Modares University. His main research interests are network security, intrusion detection and prevention, malware detection, evolutionary algorithms, and data mining.