Privacy-preserving Trajectory Data Publishing from Adversary with Limited Information

Seongun Choi[1] Toshiro Hikita[1] Rie Shigetomi Yamaguchi[1]

Abstract: Privacy issues obstruct data to be published because anonymizing data to preserve privacy eventually lowers the data usefulness. For trajectory data, it is challenging to remove privacy threat and to retain data usefulness at the same time because of its high dimensionality and sequentiality. Additionally, anonymizing trajectory data has two conflicting goals of k-anonymity and l-diversity. Many researches had limited focus on k-anonymity for trajectory data. In this paper, we consider both k-anonymity and l-diversity by assuming an adversary who has limited information of pre-determined locations. We sanitize data not to allow the adversary to know any individual's sensitive information.

Keywords: trajectory, privacy, data publication

1. Introduction

Location aware devices and location based services(LBSs) are widely spread nowadays. In LBSs, customers provide their location information and the service provider receives the location, then provides adequate service. Those trajectory[*1] data earned from customers are collected by the service provider to be analyzed with data mining tools, so that they could increase their service quality. Furthermore, not only increasing their service quality, those mining results benefit other various applications such as city development, travel recommendation, location based advertising, etc. To make those applications fully utilized, publishing data to other parties are devised. Plus, some parties need to share their data to increase the data size which will make better mining performance. For example, the mining results would be better if hospitals share their patients information. For those reasons, data publication(sharing) is highly motivated.

Data publication is often hindered by privacy problem because data often involves private information of customers (e.g., diagnose in hospital). Trajectory is also considered as private information because it is easy to infer individual's home address, office address, political views, religious inclination, etc from the trajectory. Thus, there have been many researches to publish data while preserving privacy. k-anonymity[1] is proposed to preserve re-identification of individuals by obstructing *identity linkage attack*. k-anonymity assumes adversary who:

- has the published data
- knows that (potential) victim is in the data table
- has side information of the victim.

Succeeding in *identity linkage attack* will allow adversary to re-identify victim and so he will know the victim's sensitive information. k-anonymity guarantees the succeeding in *identity linkage attack* $\leq 1/k$ by making any individual undistinguishable from at leat $k-1$ other individuals. It aggregates k individuals by sanitizing data to have same values. However, even though adversary fails to re-identify victim, somehow he may succeed to know the victim's sensitive information when all the aggregated trajectories share same sensitive values (*attribute linkage attack*). l-diversity[2] and confidence bounding[3] are proposed to secure data from *attribute linkage attack*. Both l-diversity and confidence bounding requires the aggregated individuals to have different sensitive values, so adversary fail to infer one.

Trajectory is considered as private because it is easy to infer private values. Meanwhile, trajectory can also be a clue (quasi identifiers) for adversary to attempt *identity linkage attack* because adversary may have side information of trajectory data. Abul et al.[4] addressed that publishing trajectory data has two conflicting goals because k-anonymity requires similarity on the quasi identifiers and l-diversity requires dissimilarity on the sensitive values. To solve this problem, Terrovitis and Mamoulis[7] divided location points in trajectory into two types; one is observed by adversary and the other is hidden from adversary. However, since they assume multiple adversaries, they failed to retain data usefulness even though they suppressed time stamp. In addition, they did not consider the case that hidden locations are too close.

In this paper, we suggest a new adversary model who has side information of limited locations for solving those problems. We model trajectory as a sequence of (location, time) doublets and define observed doublet and unobserved doublet to classify each doublet to quasi-identifiers or sensitive

[1] Graduate School of Information Science and Technology, The University of Tokyo

[*1] Several location and time doublets form a trajectory

attributes of trajectory. Then, we propose an algorithm that sanitizes the trajectory data to be safe from privacy threat. We introduce related work in Section 2. Problem definition is described in Section 3. We propose our sanitizing algorithm in Section 4 which is discussed in Section 5. We conclude in Section 6 and our future work is described in Section 7.

2. Related Work

2.1 Common Privacy Notions

k-anonymity[1] is a traditional framework which guarantees the succeeding in *identity linkage attack* $\leq 1/k$. The basic assumptions in this framework is that each record in data table is uniquely correspond to an individual and adversary attempts to re-identify an individual with his side information. A record consists of attributes which are classified to follows:

- *explicit-identifiers* such as social ID, passport number, phone number, name[*2] are uniquely correspond to individuals.
- *quasi-identifiers* such as gender, age, zip code are not uniquely correspond to individual, yet, combining several quasi-identifiers may let adversary re-identify individuals.
- *others* are the attributes that are not classified to explicit-identifiers or quasi-identifiers. Others are often divided to sensitive values and non-sensitive values. For example, Heavy diagnose such as cancer, HIV are classified to sensitive. On the other hand, relatively light disease are classified to non-sensitive.

To achieve k-anonymity for the published data, explictidentifiers should be removed before publishing because it uniquely corresponds to the individual. Quasi-identifiers are sanitized to have same values with at least $k-1$ other records, thus a record cannot be distinguished from at least $k-1$ other records. Others are not sanitized for k-anonymity, instead, they are sanitized to avoid *identity linkage attack*. When adversary succeeds to re-identify a victim, then he will know the sensitive values of him. Even though k-anonymity is satisfied, the adversary may learn sensitive values of victim in k-aggregated data if they share same sensitive value. To avoid *identity linkage attack*, l-diversity[2] and confidence bounding[3] are proposed, basically both notions require each of aggregated records to have different sensitive values.

Others also can be a quasi-identifiers. [5] claimed that others should be regarded as quasi-identifiers (so, they should be same to satisfy k-anonymity) to be free from *identity linkage attack*. However, regarding others as quasi-identifiers allow adversary to do *attribute linkage attack* possible. As Abul et al.[4] addressed, quasi-identifiers should be similar to avoid *identity linkage attack* and others should be dissimilar to avoid *attribute linkage attack*.

2.2 Anonymizing Trajectory Data

There were many related works focusing on satisfying k-anonymity for trajectory data[4], [6], [7], [8], [9] using mechanisms of *generalization* or *suppression* to aggregate k trajectories in data table. To the best of our knowledge, most works deemed trajectory as a quasi-identifier except [7]. Abul et al.[4] proposed (k, δ)-anonymity which is motivated from the imprecision of positioning system (e.g., GPS). (k, δ)-anonymity represents the trajectory as a cylindrical volume, where its radius δ represents the possible location imprecision. Thus, trajectory is indistinguishable from $k-1$ other trajectories which are closer than δ. They aggregate trajectories to bounding tube where k trajectories possibly present, and suppressed the trajectories that cannot be aggregated. Nergiz et al.[6] generalized the location points to regions, so that trajectories form k aggregated cylindrical volume. Then they re-construct trajectories from the aggregated cylindrical volume by selecting k atomic points from aggregated regions and linking points.

On the other hand, Terrovitis and Mamoulis[7] focused on suppression to keep data as accurate as possible based on assumption that an adversary holds partial information about trajectory. In [7], trajectory is considered as a sequence of location points and adversary has side information of several places that individual visited. The location points what the adversary don't know is considered as sensitive values. It is possible for adversary to know partial trajectory of individuals, e.g., he can earn side information from credit card company or transportation card, etc. They proposed an algorithm considering multiple adversaries having different side information, however, there are too many things to consider, even though they suppressed time points, it was not applicable in general. Chen et al.[8] proposed a comprehensive notion of $(K, C)_L$-privacy and suggested a local suppression algorithm. $(K, C)_L$-privacy guarantees followings;

- Pr[Succeeding in an *identity linkage attack*]$\leq 1/k$
- Pr[Succeeding in an *attribute linkage attack*]$\leq C$

where an adversary's side information is bounded by at most L continual location-time doublets. Instead of [7], they suppose an adversary who has at most L location-time doublets of every individuals. They considered records have trajectory and a attribute and considered *attribute linkage attack* for the sensitive attribute, however, they did not consider *attribute linkage attack* for the doublets which adversary don't know. Additionally, limiting the number of location-time doublets to L is not realistic considering a capable adversary who can continue stalking every trajectory of potential victim. Later, Ghasemzadeh[9] employed local suppression to achieve (K, L)-privacy for the trajectory data however, did not consider trajectory as sensitive. It is said that employing generalization to the trajectory data loses the data utility more than employing suppression because generalization has to merge all selected child nodes to their parent node, when suppression only removes the selected child node violating privacy[10].

[*2] Person name is socially recognized as explicit-identifier though it is not uniquely connected to individual

$$P(loc, r, T, \hat{T}) = \max_{\hat{tr} \in \hat{T}} \frac{|\{tr | tr \in T(\hat{tr}) \wedge \exists (loc, t) \in tr \ s.t. \ (loc, t) \notin \hat{tr} \wedge (loc, t) \in circle(loc, r)|}{|T(\hat{tr})|} \tag{1}$$

However, trajectory data recorded by the positioning system (e.g., GPS) naturally has imprecision[6] thus, we can say that it is already a generalized version of trajectory. Also, many applications can not be realized without generalization, it can be employed for retaining better data utility.

2.3 Differential Privacy

Recently, ϵ-differential privacy[11] is proposed which guarantees the difference of adjacent data table negligible by adding controlled random noise such as Laplace noise. To make data differentially private while retaining data usefulness, the data size should be big and one record can not change many things to data (*sensitivity*). Data size should be big in order to decrease the influence on one record. Small size of data is much more susceptible to noise. Also, *sensitivity* should be small because it will make a bigger noise. So, it can be employed in very restricted dataset and restricted use. It is not recommended to flat data (e.g., trajectory data) because it will loss its truthfulness by noise. Chen et al.[12] proposed a sanitization algorithm for trajectory data to satisfy ϵ-differential privacy on the purpose of two data mining tasks; count query and maximal frequent sequence mining which are very basic mining tasks.

2.4 Our Contribution

In this paper, we assume adversary who has limited side information and attempts *Identity linkage attack*. The main difference between [7] is two; one is that we define privacy threat that adversary confidently infer individual's sensitive location with distance error $\leq r$. The other is that we engaged generalization approach to remove privacy threat.

3. Security Definition

We model trajectory as a sequence of (location, time) doublets. We assume adversary has limited side information, so we divide doublets to observed doublets and unobserved doublets. Then, we formalize privacy threat and data usefulness.

3.1 Trajectory Model

We consider trajectory tr as a sequence of location, time doublets, tr is represented as $\{(loc_1 t_1) \rightarrow (loc_2 t_2) \rightarrow ... \rightarrow (loc_n t_n)\}$ where $loc_i \in \mathcal{L}$ (location universe) and $t_i \in \mathcal{T}$ (time universe). $(loc_1 t_1)$ means that an individual visited loc_1 at t_1.

We classify each doublet to follows;

- *observed doublet* : A doublet which location is exposed to adversary and observed by adversary. Observed doublets are regarded as quasi-identifiers because adversary may use the information to re-identify the individual.
- *unobserved doublet*[*3] : A doublet which location is not exposed to adversary. Unobserved doublets are regarded as sensitive attributes. Especially, home location, office location, and other frequent visited locations should be treated as sensitive attributes.

based on assumption that adversary has limited side information for trajectory.

3.2 Adversary Model

We consider adversary has knowledge of partial trajectory of individuals' trajectory. In this paper, we consider a specific condition that adversary can access to transportation card company and he is able to certify that card owner is in the published data table. Then, partial trajectories of individuals in the published data, who used transportation, are exposed by the adversary.

It is a reasonable scenario, compare to L continual doublets[8] because it reminds a physical stalker, however it is hard for the stalker to stalk the individual. If we assume that there is a talented adversary who can stalk L continual doublets, then it is easy for him to follow the individual over L doublets. In [7], multiple adversaries with partial side information are considered, however, multiple adversaries require the doublets to be 'quasi-identifiers' and 'sensitive attributes' at the same time. For example, if a is observed by adversary A and not observed by adversary B, then doublets involving a should be similar not to be re-identified by A and should be dissimilar not to be inferred by B.

3.3 Privacy Threat

We consider the attack is succeed when a sensitive location is confidently inferred by adversary. We define a sensitive location which distance to any unobserved location is $\leq r$. It is because, adversary can infer a sensitive location from close locations. If sensitive locations of aggregated trajectories are too close, then it will loss its diversity. Adversary attempts *identity linkage attack* to trajectories and *attribute linkage attack* to aggregated trajectories to learn sensitive locations from individuals in published data.

We regard the attack failed when adversary cannot infer sensitive area of any individual confidently. We formalize the privacy threat here. We define trajectory data table T and adversary's side information table $\hat{T}$. Trajectory data table T consists of several tr which uniquely correspond to individual. Side information table $\hat{T}$ consists of partial trajectories $\hat{tr}$ which is originally from tr. We denote trajectories in data table T and which is involving tr for its partial trajectory, $T(tr)$. Let adversary infers sensitive location of individual in data table T. Sensitive location can be any lo-

[*3] Actually, not every unobserved doublets are sensitive, however, we consider them as sensitive.

Table 1 Example of trajectory data table T

ID#	trajectory
tr_1	$A5 \to b5 \to e7 \to I8$
tr_2	$B2 \to a5 \to e7 \to L9$
tr_3	$G4 \to c6 \to e7 \to J8$
tr_4	$F5 \to d6 \to e7 \to K9$
tr_5	$C2 \to a3 \to c3 \to E4$
tr_6	$D2 \to a3 \to c3 \to H5$

Table 2 Example of adversary's side information table $\hat{T}$

ID#	observed trajectory	count
$\hat{tr}_1$	$b5 \to e7$	1
$\hat{tr}_2$	$a6 \to e7$	1
$\hat{tr}_3$	$c6 \to e7$	1
$\hat{tr}_4$	$d6 \to e7$	1
$\hat{tr}_5$	$a3 \to c3$	2

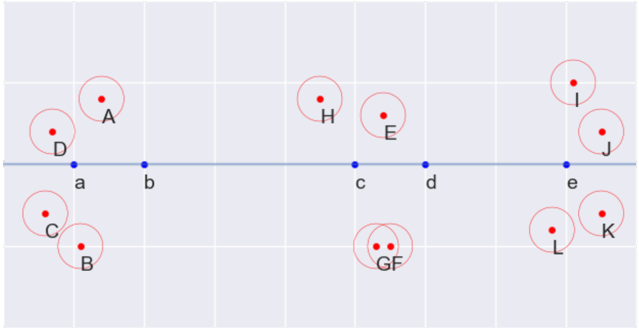

Fig. 1 The map of example locations

cation in radius r circle area from any unobserved locations. The privacy threat is then computed as equation 1, where, $circle(loc, r)$ represents the center loc, radius r circle area. (See Fig. 1 for circle area.) If $P(loc, r, T, \hat{T})$ equals to 1, then adversary confidently infers that the member of aggregated group visited loc, thus individual in the aggregated group is exposed to privacy threat. For side information $\hat{tr}_j \in \hat{T}$, we say loc is insecure when $P(loc, \hat{tr}_j, T) > C$. In some cases, not only location, time is also sensitive, however, in this paper we focus on preserving locations.

We show a simple example of trajectory data table in Table 1. In this example, we assume a simple origin-destination trajectory which is represented as {origin → get on station → get off station → destination}. We denote unobserved locations, here origin and destination, as large capital (A to L), and observed locations, here get on and off stations, as small capital (a to e). Adversary has side information of location universe $\mathcal{L}_{ob} = \{a, b, c, d, e\}$, i.e., if individual pass through locations in $\mathcal{L}_{ob}$, then the log will be reported to adversary. We set privacy parameter $C = 0.5$ and r is shown in Figure 1. The adversary's side information is shown in Table 2. We calculate privacy threat for every location points in the circled area. From Table 2, adversary can uniquely re-identify tr_1, tr_2, tr_3, tr_4, thus their sensitive locations. tr_5 and tr_6 are not uniquely re-identified because adversary cannot distinguish them. Location point loc satisfies $P(loc, r, T, \hat{tr}_5) \leq 0.5$, so tr_5 and tr_6 are safe from privacy threat.

We sanitize trajectory data table not to leak sensitive location to adversary. The sanitized trajectory data table is shown in Table 3. The mechanism for sanitizing is described in Section 4. While sanitizing, $a \in \hat{tr}_2$ and $b \in \hat{tr}_1$ are generalized to location group $\{a, b\}$. This generalization makes $\hat{tr}_1$ identical to $\hat{tr}_2$ and eventually eliminate privacy threat of sensitive locations. However applying same generalization to $c \in \hat{tr}_3$ and $d \in \hat{tr}_4$ doesn't go well because, location G and F are close each other (We say locations are close if there exists any location points which distance to each locations is less than or equal to r.), adversary is able to infer sensitive location of $\hat{tr}_3$ and $\hat{tr}_4$. There is no need to do sanitization to $\hat{tr}_5$ and $\hat{tr}_6$ since they are safe from privacy threat.

Table 3 Example of sanitized trajectory data table T'

ID#	trajectory
tr_1	$A5 \to \{a, b\}5 \to e7 \to I8$
tr_2	$B2 \to \{a, b\}5 \to e7 \to L9$
tr_5	$C2 \to a3 \to c3 \to E4$
tr_6	$D2 \to a3 \to c3 \to H5$

3.4 Data Utility

Data recipient will analyze the anonymized data in various ways. Generally, the data publisher doesn't know what kind of analyzing tasks the data recipients will do. Different data recipients may have different purposes, so we measure the similarity between data table T and the anonymized data table T' is a reasonable information metric.

We measure the utility loss between a location and its generalized location to measure how its similarity damaged. Generalized location is denoted as a group consists of several locations. We measure the utility loss by measuring the distance between the original location and the center of locations in the potential generalized group. Measuring every utility loss in $loc \in tr$, we earn utility loss of trajectory data T. Measuring the every distance between locations $loc \in tr$ and every loc' in its generalized group is as follows;

$$\begin{aligned} UL(T, T') &= \sum_{tr \in T} dist(tr, tr') \\ &= \sum_{tr \in T} \sum_{(loc,t) \in tr} dist(loc, Group(loc)) \end{aligned}$$

where,

$$dist(loc, Group(loc)) = dist(loc, \frac{\sum_{p \in Group} p - loc}{|Group(loc)| - 1}).$$

If such $Group(loc)$ that satisfies equation 1 doesn't exist, then trajectories having violating locations are suppressed. We restrain suppression as possible, at this time, we penalty the maximum utility loss to the suppressed trajectory in T.

4. Algorithm

We sanitize trajectory data to make it safe from privacy threat using generalization and suppression (Section 3.3) while minimizing the utility loss (Section 3.4) as possible. Briefly, we generalize the observed locations to location groups or suppress the observed locations, so that adversary cannot get any further information of individual over C confidently. Since finding the optimal points of generalization and suppression for anonymizing that minimizes the utility loss is NP-hard[13], we use a greedy algorithm which selects the best solution that minimizes utility loss at one time.

4.1 Sanitizing Algorithm

Our sanitizing algorithm is shown in Algorithm 1. At first, pick up the partial trajectories $\hat{T}_v$ which is violating privacy from $\hat{T}$. This process is described in algorithm 2. (line 1) Until $\hat{T}_v$ becomes empty, we randomly select a trajectory $\hat{tr}$ from $\hat{T}_v$, find its nearest trajectory $\hat{tr}^*$ in $\hat{T}_v$. (line 3-4) The closest trajectory will minimize the utility loss of trajectories in $T(\hat{tr})$ after generalization. If such $\hat{tr}^*$ exists, generalize every $\hat{tr}$ in T to group of $\{\hat{tr}, \hat{tr}^*\}$. (line 5-6) As T renewed to generalized version, so shall $\hat{T}$ do. (line 7) The generalized trajectories may steal not be safe from privacy threat. So, we renew $\hat{T}_v$ by algorithm 2. (line 8) As a matter of fact, violation check does not have to be done to whole trajectories, it is more effective just checking the generalized trajectories. if such $\hat{tr}^*$ does not exist, just remove trajectories in $T(\hat{tr}^*)$. (line 10) Also, remove corresponding partial trajectory $\hat{tr}^*$ from $\hat{T}$ and $\hat{T}_v$. (line 11)

Algorithm 1 Sanitizing algorithm

```
Input: trajectory $T$, adversary's info. $\hat{T}$, confidence $C$, radius $r$
Output: Sanitized trajectory data $T$
 1: $\hat{T}_v$:=CheckViolate($T, \hat{T}, C$)
 2: while $T_v$ do
 3:     pick $\hat{tr} \in \hat{T}_v$ randomly
 4:     $\hat{tr}^* :=$ argmin$_{\hat{tr}^* in \hat{T}}$ $UL(\hat{tr}, \hat{tr}^*)$
 5:     if $\hat{tr}^*$ exists then
 6:         generalize every $\hat{tr}$ to $\{\hat{tr}, \hat{tr}^*\}$
 7:         renew $T$, $\hat{T}$
 8:         $\hat{T}_v$:=CheckViolate($T, \hat{T}, C$)
 9:     else
10:         remove $T(\hat{tr})$ from $T$
11:         remove $\hat{tr}$ from $\hat{T}$ and $\hat{T}_v$
12:     end if
13: end while
14: return $T$
```

Algorithm 2 Violation check algorithm

```
Input: $T, \hat{T}, C, r$
Output: observed data $\hat{T}_v$ that violate privacy
 1: $\hat{T}_v \leftarrow \phi$
 2: for all $\hat{tr} \in \hat{T}$ do
 3:     if $\exists loc$ s.t. $P(loc, r, T(\hat{tr}), \hat{tr}) > C$ then
 4:         $\hat{T}_v \leftarrow \hat{tr}$
 5:     end if
 6: end for
 7: return $\hat{T}_v$
```

4.2 Violation Check Algorithm

To verify that trajectory data is safe from privacy threat, which is formulated in Section 3.3, for every $\hat{tr} \in \hat{T}$, we find location loc such that satisfies $P(loc, r, T(\hat{tr}), \hat{tr}) > C$. (line2-3) If such loc exists, it means that loc is threatened to be known by adversary, so we put it in $\hat{T}_v$. (line 4) Doing this verification repeatedly for every $\hat{tr}$, finally return the $\hat{T}_v$.

5. Security Discussion

In this section, we prove that our algorithms remove privacy threats and show the trajectory data table in Table 1 sanitized to Table 3. Then, we discuss our adversary model and more stronger model.

5.1 Proof of our algorithm

We first prove that our algorithms work properly to remove privacy threat. What we show in Theorem 1 is that our algorithm removes privacy threats. Theorem 2 shows us that privacy threatening trajectories are correctly added to $\hat{T}_v$.

Theorem 1. *Algorithm 1 eventually diminishes $\hat{T}_v$.*

Proof. Algorithm 1 changes $\hat{T}_v$ in 2 ways; whether $\hat{tr}^*$ exists or not for randomly given $\hat{tr}$. if $\hat{tr}^*$ exists, then, $\hat{tr}$ is generalized to $\{\hat{tr}, \hat{tr}^*\}$. Thus, $\hat{tr}$ is removed from $\hat{T}_v$ and $\{\hat{tr}, \hat{tr}^*\}$ is added (or it is already added) to $\hat{T}_v$. So, when $\hat{tr}^*$ exists and $\{\hat{tr}, \hat{tr}^*\}$ already exists, the number of $\hat{T}_v$ decreases. Otherwise, when $\hat{tr}^*$ exists and $\{\hat{tr}, \hat{tr}^*\}$ does not exists, the number of $\hat{T}_v$ is not changed. Doing this process several time, there will be no $\hat{tr}^*$ anymore. It is more easy to prove when $\hat{tr}^*$ does not exist. If $\hat{tr}^*$ doesn't exist, $\hat{tr}^*$ is removed from $\hat{tr}^*$, so $\hat{tr}^*$ decreases. □

Theorem 2. *Algorithm 2 adds every partial trajectory that is threatening privacy to $\hat{T}_v$.*

Proof. Algorithm 2 runs repeatedly for every trajectory in $\hat{T}$. If it correctly picks up a trajectory $\hat{tr} \in \hat{T}$ that violates privacy, then it will add every violating traejectory to $\hat{T}_v$. if there exists location loc such that threatening privacy ($P(loc, r, T(\hat{tr}), \hat{tr}) > C$), it will be added to $\hat{T}_v$. Thus, algorithm 2 correctly adds every partial trajectory to $\hat{T}_v$. □

We show that our algorithm 1 sanitizes trajectory data in Table 1 to Table 3, when adversary has knowledge of Table 2. Following the procedure in algorithm 1, we first do violation check. As we already mentioned, $\hat{tr}_1, \hat{tr}_2, \hat{tr}_3, \hat{tr}_4$ are uniquely correspond to trajectory in T, so they are added to $\hat{T}_v$. We randomly pick a trajectory from $\hat{T}_v$, let's decide that we picked $\hat{tr}_1$ for the first. (We proved that $\hat{T}_v$ decreases eventually. The pick up order does not matter.) To minimize the utilily loss, $\hat{tr}_1$ is generalized to the closest location point, b. Thus, $\hat{tr}_1$ becomes $\{\{a, b\}5 \rightarrow e7\}$. Doing same thing to $\hat{tr}_2$, it generalized to $\{\{a, b\}5 \rightarrow e7\}$. For $\hat{tr}_3$, $\hat{tr}_4$, since they have no $\hat{tr}^*$ that generalization with it makes the trajectory safe from privacy threat. Note that $\{\{c, d\}6 \rightarrow e7\}$ steal threatening privacy since both $\hat{tr}_3$ and $\hat{tr}_4$ have close origin location. So, they are suppressed from Table 1. Consequently, trajectory data in Table 1 sanitized to Table 3.

5.2 Stronger Adversary Model

In this paper, we assume that adversary knows partial trajectory of individual in published data. Specifically, we assume adversary has side information of transportaion company, so he knows every individual using transportation. However, adversary may know further information such as individual's school thanks to commuter pass. In such case, it is easier to infer the individual's sensitive location for adversary. To preserve individual whose destination is exposed to adversary, there should be at least $\lceil 1/C \rceil - 1$ trajectories share same detstination and observed locations and different (r far away from each other) origin locations. Simple calculation teaches us that to have same data usefulness to N trajectories, it requires N^2 trajectories for stronger adversary having further information of destination.

5.3 General Trajectory Model

We consider trajectory as discrete sequence of (location, time) doublets, restricted its form to {origin → get on station → get off station → destination}. This is the most simplest trajectory form, some data recipient may want to utilize longer trajectory data. However, long trajectory usually increases its dimensionality, so longer trajectory requires more sanitization, it losses its utility. Some works removed time stamps from trajectory[7] or cloaked time to have one-hour term[8], [9] to lower the dimensionality.

6. Conclusion

Since trajectory data mining has various usages, trajectory data publication will benefit us in many applications. To realize trajectory data publication, it is very important to remove personal privacy in published data. In this paper, we focused on that trajectory data can be a quasi-identifier and also can be a sensitive attribute. Since quasi-identifier and sensitive attribute are having conflict goals, it is challenging to anonymizing trajectory data. In this paper, we assume adversary who has side information of transportation system attempts to know individual's sensitive locations. We defined privacy threat and we used generalization and suppression to anonymize trajectory data, so that published trajectory data be safe from privacy threat. Specifically, we define an utility metrics by measuring the utility loss when location point of trajectory generalized. Then we proposed a greedy algorithm which chooses the optimal solution, minimizes the utility loss, for each randomly selected trajectory that violates privacy. We showed that our algorithm removes violating trajectories and make trajectory data safe from privacy threat. We discussed more strong adversary who has further knowledge of destination, however, it requires the second power of data size to have same data utility. Further, we discussed that general trajectory has higher dimensionality so that it requires more sanitization.

7. Future Work

It is often told that it is hard to anonymize trajectory data because of its high dimensionality. For now, we will apply our algorithm to dataset of Tokyo area 2008 from People Flow Project[14] by changing the radius parameter r and confidence parameter C, and the time stamp cloaking interval. Specifically, we extract origin-destination trip using train from dataset[14]. Later, we will extend our work in two ways.

One is expanding origin-destination trajectory to one-day trajectory. Longer trajectories provide better analyze results, however the data would be more distorted because of its increased dimension. We will evaluate our algorithm to one-day trajectory, and on the other hand, we will working on improving our algorithm; i.e., generalizing time stamp to preserve privacy which will be our second future work. In this paper, we did not do any sanitization process to time stamp in doublets, however, engaging generalization or suppression to time stamp will increase data utlity. For example, in Table 1, tr_3 and tr_4 are suppressed, however, if time is generalized, both $c6$ and $d6$ can be generalized to $\{a, b, c, d\}\{5, 6\}$.

Acknowledgements

This work was supported by JSPS KAKENHI.

References

[1] L. Sweeney: "K-anonymity: A model for protecting privacy", Int. J. Uncertain. Fuzziness Knowl.-Based Syst., **10**, 5, pp. 557–570 (2002).

[2] A. Machanavajjhala, D. Kifer, J. Gehrke and M. Venkitasubramaniam: "L-diversity: Privacy beyond k-anonymity", ACM Trans. Knowl. Discov. Data, **1**, 1 (2007).

[3] K. Wang, B. C. M. Fung and P. S. Yu: "Handicapping attacker's confidence: An alternative to k-anonymization", Knowledge and Information Systems (KAIS), **11**, 3, pp. 345–368 (2007).

[4] O. Abul, F. Bonchi and M. Nanni: "Never walk alone: Uncertainty for anonymity in moving objects databases", Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ICDE '08, Washington, DC, USA, IEEE Computer Society, pp. 376–385 (2008).

[5] Y. Itakura, H. Kikuchi, H. Takaki, K. Takahashi, H. Nakagawa, T. Hikita, K. Hirota, R. Yamaguchi and T. Watanabe: "Beyond the fantasy of "perfect anonymit"", Symposium on Cryptography and Information Security, IEICE (2014).

[6] M. E. Nergiz, M. Atzori and Y. Saygin: "Towards trajectory anonymization: A generalization-based approach", Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, SPRINGL '08, New York, NY, USA, ACM, pp. 52–61 (2008).

[7] M. Terrovitis and N. Mamoulis: "Privacy preservation in the publication of trajectories", Proceedings of the The Ninth International Conference on Mobile Data Management, MDM '08, Washington, DC, USA, IEEE Computer Society, pp. 65–72 (2008).

[8] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai and K. Wang: "Privacy-preserving trajectory data publishing by local suppression.", Inf. Sci., **231**, pp. 83–97 (2013).

[9] M. Ghasemzadeh, B. C. M. Fung, R. Chen and A. Awasthi: "Anonymizing trajectory data for passenger flow analysis", Transportation Research Part C: Emerging Technologies (TRC), **39**, pp. 63–79 (2014).

[10] B. C. M. Fung, K. Wang, R. Chen and P. S. Yu: "Privacy-preserving data publishing: A survey of recent developments", ACM Computing Surveys, **42**, 4, pp. 14:1–14:53 (2010).

[11] C. Dwork, F. McSherry, K. Nissim and A. Smith: "Calibrating noise to sensitivity in private data analysis", Proceedings of the Third Conference on Theory of Cryptography, TCC'06, Berlin, Heidelberg, Springer-Verlag, pp. 265–284 (2006).

[12] R. Chen, B. C. Fung, B. C. Desai and N. M. Sossou: "Differ-

entially private transit data publication: A case study on the montreal transportation system", Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, New York, NY, USA, ACM, pp. 213–221 (2012).

[13] A. Meyerson and R. Williams: "On the complexity of optimal k-anonymity", Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS '04, New York, NY, USA, ACM, pp. 223–228 (2004).

[14] "People flow project, center for spatial information science, the university of tokyo", http://pflow.csis.u-tokyo.ac.jp/?page_id=943 (2016).