Walking without Friends: Publishing Anonymized Trajectory Dataset without Leaking Social Relationships

Kai Zhao, Zhen Tu, Fengli Xu, Yong Li, *Senior Member, IEEE,* Pengyu Zhang, Dan Pei, Li Su, Depeng Jin, *Member, IEEE*

***Abstract*—Trajectory data has been widely collected via mobile devices and publicly released for academic research and commercial purposes. One primary concern of publishing such a dataset is the privacy issue. Previous protection schemes mainly focus on preventing re-identification attack, which utilizes the uniqueness of trajectories. However, the correlation between trajectories, which has not been given much attention to before, could also give rise to serious privacy leakage. Recent studies have proved that it is possible to identify social relationship, de-anonymize trajectories or even infer user's locations by analyzing the correlation between users' trajectories. We identify the serious privacy problem of social relationship leakage caused by what we call social relationship attack and aim to protect social relationship information, which cannot be protected by existing algorithms. We contribute to the design of a new privacy model and an effective system to deal with social relationship attack and re-identification attack simultaneously while maintaining high data utility. We propose a *SlidingWindow* algorithm to merge trajectories according to their *social-aware distance*, which concerns both the spatiotemporal distance and social proximity. Evaluations of two trajectory datasets under different scenarios demonstrate that our system provides more than 1.84 times privacy protection at the cost of only 2.5% data utility loss.**

***Index Terms*—Privacy preserving data publishing, privacy, trajectory, social relationship**

I. Introduction

With the prevalence of mobile devices and localization technologies, mobile user trajectories have been collected through cellular network and applications running on mobile devices. Various mobile user trajectory datasets have been published [1–3] for academic research and industrial engineering. The trajectory information is valuable in many applications, including intelligent transportation [4], urban computing [5] and mobile service provisioning [6–10], etc. Especially for the mobile service, we can provide better service with the knowledge from data. For example, [7] offers services like sharing life experiences and recommending travel and location by analyzing people's trajectories. In the future, when we are in an era of big data, utilizing information from trajectory data to improve mobile service will be more and more important. However, such data often discloses users' privacy, such as telling users' preferred locations, mobility patterns, or even their social relationships. Thus, the primary concern of publishing such dataset is how to preserve users' privacy.

Unfortunately, previous technologies concentrating on preventing re-identification attack do not completely preserve users' privacy. Re-identification attack aims to re-identify individuals from the anonymized dataset with additional information, i.e., spatiotemporal points. Due to the uniqueness of individual trajectory, the attacker might be able to successfully re-identify the individual by matching the additional information with the trajectories in the dataset [11, 12]. [11] shows that four unique spatiotemporal points are enough to identify 95% of the users in a dataset with one million users. To address this problem, k-anonymity, l-diversity and even t-closeness [13–15] are proposed to diminish the uniqueness of mobile users' trajectories. Although these technologies do prevent privacy leakage caused by releasing user's unique trajectory, the correlation between different trajectories could also lead to serious privacy leakage. For example, some news have noticed this privacy issue: "Cell phone tracking can reveal our private associations and relationships with one another", "make note of whenever people being tracked crossed path or spent time together, showing who our friends, associates and lovers are", "infers relationships based on mobile location data", on the CNN [16] and Washington Post [17]. To make matters worse, recent studies [18–25] have proven that by analyzing the correlation among trajectories, it is possible to infer social relationships of mobile users. For example, [18] utilized spatiotemporal patterns in users' physical proximity and calling patterns to accurately infer the friendships of mobile users with 95% accuracy.

Inferring social relationships by utilizing the correlation of trajectories, we name as **social relationship attack**, is a serious privacy breach. We spend most of our time with friends and families. Consequently, it is easier to observe higher correlations of our mobile trajectories with friends and families than strangers'. Different from the online social network, this kind of relationship is hidden in the trajectory dataset and is not voluntarily disclosed by the user. Besides, it is difficult for users to hide any of these friends because they are inherent in trajectories. Therefore, the correlations extracted

Manuscript received September 1, 2018; revised January 3, 2019 and March 17, 2019; accepted March 19, 2019. This work is supported by National Basic Research Program of China (973 Program) (No. 2013CB329105), National Nature Science Foundation of China (No. 61301080, No. 91338203, No. 91338102, and No. 61321061), Beijing Innovation Center for Future Chip, the research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology, and China Ministry of Education-CMCC Research Fund Project no. MCM20160104.

The authors are with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

from the trajectory data have the potential of revealing social relationships. Aware of such serious privacy leakage, it is urgent to protect mobile users from social relationship leakage. However, existing trajectory anonymization algorithms and solutions, which only consider the uniqueness of trajectories, do not consider the correlation of trajectories and as a result, cannot prevent social relationship attack. Therefore, a new privacy model to protect social relationship is desperately needed.

We aim to propose a new privacy model to defend against social relationship attack and re-identification attack when publishing the trajectory data. It is a challenging problem because of the following three reasons. First, different social relationship attack methods use different features of correlation to infer the social relationship. Thus, we do not know what kind of correlation is the underlying reason for social relationship leakage. Second, we need to make a trade-off between reduced uniqueness and increased correlations between trajectories. Preventing social relationship attack by decoupling the correlation between trajectories will increase the uniqueness of trajectories, which makes re-identification attack easier. Finally, techniques to prevent social relationship attack and re-identification attack will decrease the data utility, which means that the value of the released data reduces.

In this paper, in order to prevent both the social relationship attack and re-identification attack caused by the correlation and uniqueness properties of trajectories, we define a new uniform privacy model and formally formulate such privacy preservation problem. By addressing the above three challenges, we propose a computationally efficient system to protect the released data from social relationship attack and re-identification attack. Our major contributions of this paper can be summarized as the following three-folds:

- To the best of our knowledge, we are the first to propose a novel privacy model for trajectory data publishing, which considers both social relationship attack and re-identification attack by satisfying k-anonymity at a specific relationship-preserving level.
- We propose a novel system to generalize trajectories, which effectively reduces the correlation of trajectories and the uniqueness as well. To the best of our knowledge, this is the first system to protect trajectory data from both social relationship attack and re-identification attack, which on the other hand without disrupting data utility.
- We evaluate our system under two real-world scenarios of mobility datasets releasing. The results demonstrate that our system successfully protects social relationships while preserving considerable utility, which provides more than 1.84 times privacy protection at the cost of only 2.5% loss of data utility.

The rest of the paper is organized as follows. §II introduces the limitations of achieving k-anonymity only considering spatiotemporal closeness and describes the attack and privacy model. §III defines the problem. §IV describes our anonymization system for trajectory datasets. §V introduces the datasets and presents the evaluation results. §VI summarizes the related work and finally §VII concludes the paper.

II. Motivation

Existing anonymizing technologies mainly achieve the widely used privacy criterion k-anonymity only considering spatiotemporal closeness, which is not enough to protect data from social relationship attack. In this section, we first introduce the attack model and two state-of-the-art implementations. Second, we briefly introduce k-anonymity and its limitation with an experiment to confirm our judgment. More privacy risk caused by the trajectory correlation is also discussed. Finally, we introduce our privacy model.

A. Attack Model

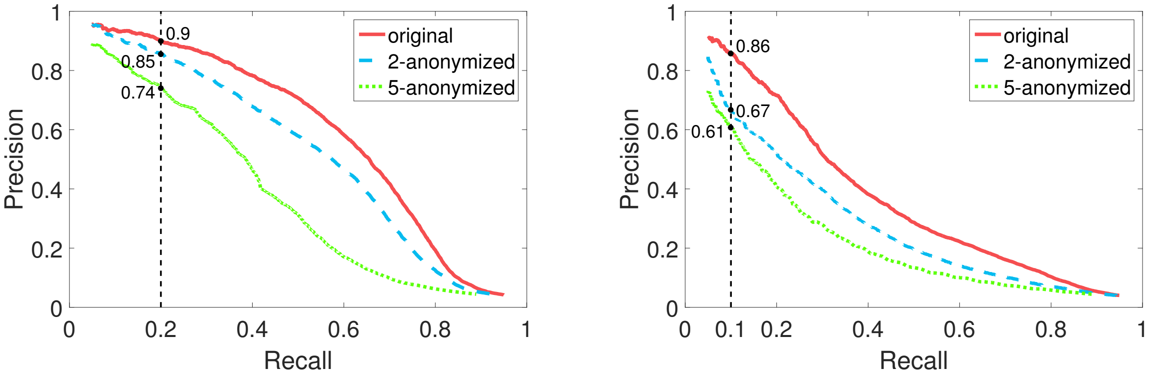
In social relationship attack, the attacker intends to infer real-world social relationships from trajectory data by utilizing properties of mobility trajectory. The attack model is a classification of friends and non-friends, which means it is a binary classification problem. Previous studies have shown two specific and effective attacks.

- **MLI** (Modified *LOCA* Inference): *LOCA* [21] means Location-Oblivious Co-location Attack. It is an algorithm used to infer social relationship from trajectories without physical locations. It first calculates the number of user transitions between locations. More transitions between two locations indicate that they are closer, which is in contrast to the meaning of spatial distance. Then they compute the interactions between user pairs by adding up the location transitions in their trajectories and regard the users with a large number of interactions as friends.
- **CAI** (Context-Aware Inference): It's a method to identify friendship from trajectories by combining spatiotemporal context information with physical proximity [18]. It utilizes two factors to capture most of the spatiotemporal patterns of physical proximity: proximity at work during the daytime labeled "in-role" and off-work proximity in the evening and on weekends labeled "extra-role". Each factor can be used to distinguish friends and non-friends. For example, a pair of users with have high "extra-role" means they have many off-work interactions and are probably friends.

B. Limitations of only Considering Spatiotemporal Closeness

k-anonymity is a state-of-the-art criterion in protecting all kinds of datasets from re-identification attack. It requires that each user in a dataset must be indistinguishable from at least k-1 other users in the same dataset. In the field of trajectory dataset protection, some algorithms achieved k-anonymity only considering spatiotemporal closeness to prevent re-identification attack [26, 27]. The main idea is merging spatiotemporal similar trajectories so that the attacker cannot re-identify any of them and the data utility loss is minimum.

However, the correlation between trajectories reveals social relationships among users even after k-anonymity achieved. While achieving k-anonymity by considering spatiotemporal closeness only, people with social relationships usually have more similar trajectories and they are more likely to be merged. Thus, the similarity of friends' trajectories will not decrease after anonymization.



(a) Algorithm in [21]

(b) Algorithm in [18]

Fig. 1: Observed precision and recall of social relationship classification using two social relationship inference algorithms on original Wi-Fi dataset and k-anonymized Wi-Fi dataset. (k=2,5 and m=8).

We experiment on a Wi-Fi trajectory dataset to confirm that k-anonymity only considering spatiotemporal closeness cannot prevent social relationship attack. The dataset contains users' trajectories in campus and their social relationships. We utilize MLI and CAI to infer social relationship from trajectories on both the original dataset and the k-anonymized dataset. The algorithm to achieve k-anonymity is a simplified version of our algorithm with the social relationship part removed. The details of the algorithm are described in section §IV. We use *precision* and *recall* to measure the results of the attack. A higher value of precision(recall) indicates the attack is more successful. The details of the measurement are described in section §III. The classification results are shown in Figure 1. Although 2-anonymity already satisfies the indistinguishability principle, higher values of k ensure higher privacy levels at the cost of accuracy. [26] shows that the anonymized dataset becomes hardly exploitable when $k > 5$. Thus we choose k to be 2 and 5. From the results, we can observe that even protected with 5-anonymity, there are still 20% pairs of users whose social relationship can be inferred with a precision of 0.74 in the first algorithm and 10% of users whose social relationship can be inferred with a precision of 0.61 in the second algorithm. In addition, the results of 2-anonymized dataset are very close to that of the original dataset, which means k-anonymity doesn't decrease the accuracy of social relationship classification. Thus, the social relationship privacy leakage is still serious after k-anonymized. In summary, anonymization with k-anonymity, which only considers the uniqueness of the trajectories and ignores their correlations, is not enough to prevent social relationship privacy leakage.

In addition to the directly inferring relationship from trajectories, there are more privacy risks from social relationship attack. First, your friends will leak your locations. If the attacker infers that two users are friends, one's trajectory can be utilized to improve the inference of the other's since friends tend to appear together. Many researchers have successfully predicted users' locations with the location information of their friends [28–30]. Friends inferred from trajectory dataset tends to have more co-occurrences and it is easier to predict locations from each other. Second, the correlations of trajectories can be used to de-anonymize trajectories. [31] finds that trajectories can be de-anonymized given an easily obtained social network. The key insight is that the internal networks of relationship in trajectory dataset can be structurally correlated with a social network. 80% of users in the trajectory dataset are identified precisely.

Thus, anonymizing trajectories without wiping off the social relationship is not enough to prevent these attacks.

C. Privacy Model

The ultimate goal of our work is to achieve *Privacy Preserving Data Publishing* (PPDP), *i.e.*, the criterion for the trajectory data publishing situation requiring that the result should be both privacy-preserving and data utility keeping [32]. It insists that each published record corresponds to an existing individual in real life. Thus, our privacy model should be consistent with the goal of PPDP while preventing both social relationship attack and re-identification attack, which can be summarized as the following two aspects.

First, in terms of privacy preserving, the anonymized trajectory data should be able to prevent social relationship attack, which indicates that the accuracy of the above attacks on the anonymized dataset should be low enough such that the attacker can only infer very limited social relationships. Low inference result means the correlations between friends are decoupled and the attacker cannot apply de-anonymizing attacks either. In trajectory data, the key factor of correlation is co-occurrence. In additional, re-identification attack should also be prevented, which means each trajectory should not be distinguishable with other trajectories.

Second, according to PPDP, we need to maintain truthfulness at the record level, i.e., spatiotemporal points in each mobile trajectory must map to locations actually visited by the user at that time. Randomized, perturbed, permuted, or synthetic data does not satisfy this requirement. Therefore, we only use generalization to anonymize the raw trajectory data, which relies on reducing data spatiotemporal precision so as to make points of different trajectories identical.

Overall, we seek a method to reduce social relationship inference accuracy and prevent re-identification attack while maintaining the truthfulness of trajectories.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2907542, IEEE Transactions on Network and Service Management

III. Problem Formulation and Challenges

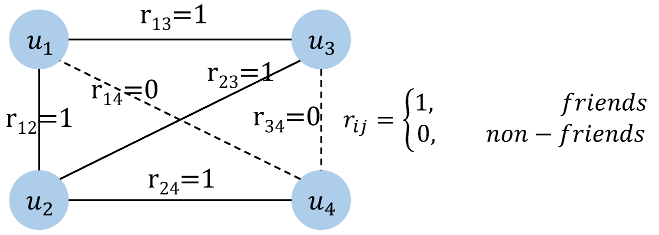
Now, we first formally introduce the privacy-preserving trajectory data publishing problem, and then discuss the challenges need to be solved with respect to preventing social relationship attack.

A. Problem Formulation

The input to this problem has two parts: a set of users' temporal discrete trajectories and the social relationships among them, which needs to be protected. The trajectory dataset, denoted as $\mathbb{D}$, can be denoted as a matrix (Table I) with users on the rows and time slots on the columns. Each trajectory has M time slots during the same period of time. Each element of the matrix is a specific location that the individual visited at that time slot, with l_i^t denoting the t-th location of user i. If there is no record at that time slot, the element is N/A. The social relationships are presented as a graph (an example in Figure 2) with vertices denoting users and edges denoting their relationships (friends or non-friends). We represent the graph as $G = \{U, R\}$, where $U = \{u_1, u_2, ..., u_N\}$ is the set of user vertices and $R = \{r_{12}, r_{13}, ..., r_{NN}\}$ is the set of edges. $r_{ij} = 1$ means u_i and u_j are friends and $r_{ij} = 0$ means they are not friends.

TABLE I: A simple example of trajectory dataset with users u_i and u_j with M time slots.

	T$_1$	**T$_2$**	**T$_3$**	...	**T$_M$**
u$_i$	l_i^1	N/A	l_i^3	...	l_i^M
u$_j$	l_j^1	l_j^2	l_j^3	...	l_j^M



r13=1
u1
u3
r14=0
r23=1
r12=1
r34=0
$r_{ij} = \begin{cases} 1, & friends \\ 0, & non-friends \end{cases}$
u2
r24=1
u4

Fig. 2: An example of social relationship graph with four nodes.

There are co-occurrences between different trajectories. For example, for u_i, u_j and a set of time slots T_{co}, $l_i^t = l_j^t$, $t \in T_{co}$, where T_{co} counts all the time slots that u_i and u_j are in the same location. The larger size of T_{co} means u_i and u_j are stronger correlated. Besides, the semantic information of time and location also delivers different information on users' relationship. The attacker only has access to the trajectory dataset. By combining all the trajectory correlation information, including the number and semantic information of co-occurrences, the attacker can reconstruct a social relationship network R' which may be very close to the real social relationship network R. Our goal is to decouple the correlations between friends and make the obtained network R' by the attacker differs greatly from R so as to prevent social relationship from being inferred. Through our specific generalization method of preserving privacy, we are going to achieve the following goals:

- To prevent social relationship attack, the social relationship inference accuracy should be very low. Inferring social relationship is a problem of binary classification, which can be achieved by some existing classification methods [18, 21]. Our goal is to make $r'_{ij} = 0$ in R' where $r_{ij} = 1$ in R as many as possible. To quantify it, we use the precision of classification result defined as follows,
$$Precision = \frac{TP}{TP + FP}, \; Recall = \frac{TP}{TP + FN}, \quad (1)$$
where TP means $True\ Positive$ representing the size of $\{(i, j)|r'_{ij} = 1, r_{ij} = 1\}$, FP means $False\ Positive$ representing the size of $\{(i, j)|r'_{ij} = 1, r_{ij} = 0\}$ and FN means $False\ Negative$ representing the size of $\{(i, j)|r'_{ij} = 0, r_{ij} = 1\}$. To consider both the precision and the recall, we use the F_1 score which is defined as follows,
$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (2)$$
A smaller value of F_1 score means the better performance of the protection method.
- To avoid re-identification attack, the generalized trajectories should meet the requirements of k-anonymity. To be specific, each trajectory should be indistinguishable with at least other k-1 trajectories. The value of k measures the protection level. Obviously, larger k provides stronger protection.

B. Challenges

Since the social relationship protection problem we formally formulated has not been investigated before, it is challenging to prevent both social relationship and re-identification attacks while preserving data utility as required. Specifically, we face the following three challenges.

First, in order to prevent social relationship leakage, we need to decouple the correlation between friends' trajectories. However, selecting proper features for representing trajectory correlation is not easy. Correlation in trajectory is mainly revealed from co-occurrences of two trajectories, which means they appear at the same location and time. More co-occurrences between two trajectories usually indicate they are more closely correlated. However, spatiotemporal information, containing the location and time a co-occurrence happens, also matters. For example, a co-occurrence at a residential area at 1:00 am contributes more to indicate a social relationship to that happening at a shopping mall at 1:00 pm. The existing methods of social relationship inference are based on different co-occurrence features. Most of them only utilize the spatiotemporal distance between trajectories [19, 21], which do not count the spatiotemporal information of co-occurrences. Some of them use more features such as semantic information of time and location[18]. As a result, there is no prior knowledge to help us choosing correlation features to prevent these different attack methods, which is the first challenge we faced.

Second, preventing social relationship attack is contradictory with preventing re-identification attack, thus preventing these

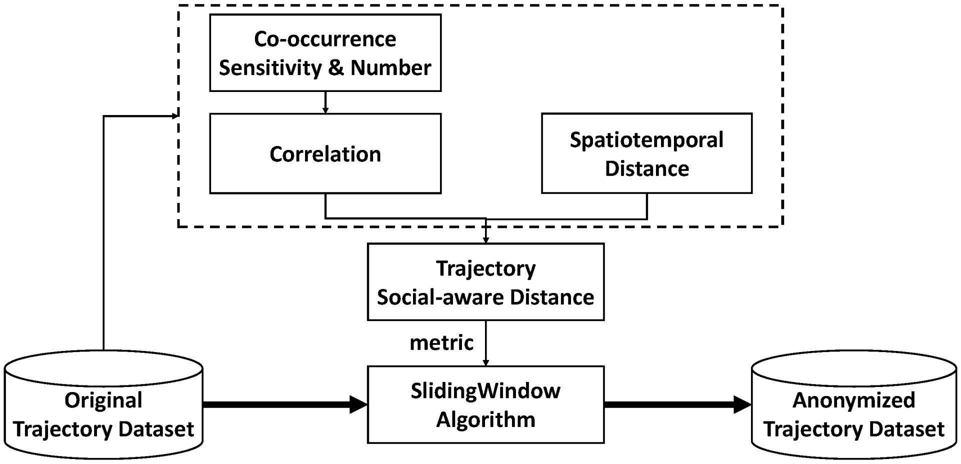

Co-occurrence Sensitivity & Number
Correlation
Spatiotemporal Distance
Trajectory Social-aware Distance
metric
Original Trajectory Dataset
SlidingWindow Algorithm
Anonymized Trajectory Dataset

Fig. 3: The designed system for social relationship protection and trajectory anonymization.

two attacks at the same time is even more challenging. Preventing re-identification attack usually needs trajectories to be similar, which increases the correlation between trajectories and causes more severe social relationship leakage. For example, two friends' trajectories are made similar to avoid being re-identified, but their friendships are more easily to be inferred since their trajectories are stronger correlated due to the similarity. Such operations decrease the probability of attacks that identify users by exploiting uniqueness of trajectories at the cost of increasing the probability of social relationship attack based on the correlations. We need to solve this contradictory by balancing these two factors when designing our system.

Third, techniques to prevent social relationship attack and re-identification attack will decrease the data utility. Generalization methods to diminish trajectory's uniqueness to avoid re-identification attack will reduce the spatiotemporal granularity of trajectory data. Besides, if preventing social relationship attack by avoiding merging friends' trajectories, data utility will be further reduced. Because merging strangers' trajectories causes more utility loss. Existing anonymization methods without considering social relationship attack will be no longer useful. In other words, we have to strike a balance between privacy protection and data utility to ensure the utilities of data while preventing both social relationship and re-identification attacks.

IV. Anonymization Algorithm

We address the above three challenges and design a novel system for social relationship protection and trajectory anonymization. We first present the key ideas of our system at a high level and then introduce the detailed algorithms. Figure 3 shows an overview of our system.

A. System Overview

As mentioned in the first challenge, the **correlation** between trajectories does not have a uniform definition in different attack methods. To identify proper features to represent the correlation between trajectories, we extract the core idea of different attacks, *i.e.*, what is the difference between friends' trajectories and strangers' trajectories. First, since we target at the scenario of individuals in a specific region, *e.g.*, a city, friends usually have more co-occurrences than strangers. Second, friends meet each other publicly or privately while strangers usually meet only publicly. Therefore, we calculate the sensitivity of each co-occurrence and regard it as a weight to the correlation. Sensitivity represents the diversity of users at that time and location, which reveals how private the co-occurrence is. In summary, the definition of correlation includes both the number and sensitivity of two trajectories' co-occurrences. The correlation module in Figure 3 computes the correlation between two trajectories to measure the social distance between them. It takes two inputs: co-occurrence sensitivity and number. The former is the sensitivity of their co-occurrences and the later is the number of their co-occurrences. More details will be introduced in §IV-B1.

The second module, **social-aware distance**, is designed to measure the distance between two trajectories concerning both the spatiotemporal distance and social proximity. As mentioned before, preventing social relationship attack is contradictory with preventing re-identification attack due to their opposite requirements. To solve this challenge, we achieve both protection simultaneously by carefully choosing trajectories to merge. Our goal is to merge spatiotemporally closed trajectories of strangers, which will increase the correlation between them, in other words, decrease the correlation between friends. To this end, we propose a new distance between trajectories: social-aware distance, which considers both the correlation of friends' trajectories and their spatiotemporal distance. Further, the correlations of friends are diverse. Some friends are closely correlated and easily to be identified while others are not. Thus, we also consider the intensity of social relationships, which means giving close social relationships stronger protection. Overall, strangers with close spatiotemporal distance tend to have short social-aware distance. More details about this module will be introduced in §IV-B2.

To deal with the utility loss, we design a **SlidingWindow algorithm** to achieve k^m-anonymity rather than full-

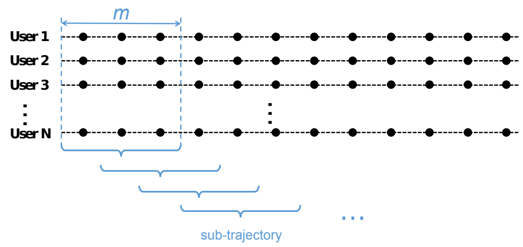


Fig. 4: The procedure of anonymizing the whole trajectory using sliding window.

length anonymization. k^m-anonymity requires that for random continuous m spatiotemporal points of a trajectory, there are at least k-1 other trajectories sharing the same m points. If a dataset achieves k^m-anonymity, attackers with background knowledge of less than m continuous spatiotemporal points are not able to re-identify users. We propose k^m-anonymity as the privacy criterion to maintain more data utility, because merging short trajectories causes less data utility loss than merging long trajectories. The core idea of this algorithm is that each time we achieve k-anonymity in sub-trajectories in an m-length window (Figure 4). With short trajectories to anonymize in each window, the data utility loss can be kept low with some sacrifices of the privacy protection. The SlidingWindow module in Figure 3 is to achieve k^m-anonymity by SlidingWindow algorithm. Its input is the original trajectory dataset and output is k^m-anonymized trajectory dataset, which is the final result of our system. More details about this module will be introduced in §IV-B3.

In summary, we design a novel system to prevent social relationship attack and re-identification attack in Figure 3. The primary process is generalizing trajectories via SlidingWindow algorithm. At the top of the system, we first calculate the correlation between trajectories and combine it with spatiotemporal distance into the social-aware distance, which is utilized as a distance metric in SlidingWindow algorithm.

B. Algorithm

1) Entropy-based Correlation: We first investigate the sensitivity of each co-occurrence and then derive the correlation between trajectories.

Since an individual's movements usually have weekly periodicity, we only consider time slots within one week. Each time slot T_t is mapped to a weekly periodic time slot T_t^w. For example, each Monday 8:00~9:00 during the time span of the dataset is mapped to the same time slot: Monday 8:00~9:00. Suppose there are two users a and b. We denote L_a and L_b as their locations at this time slot, and denote L_θ as the union of L_a and L_b, *i.e.*, $L_\theta = \{L_a, L_b\}$. Thus, the co-occurrence is denoted as (L_θ, T_t^w). So how to quantify the sensitivity of the co-occurrence? Empirically, if there are many different users visiting location L_θ at T_t^w, then this location is not sensitive at this moment. This is very similar to the definition of chaos, which can be well described by *Shannon Entropy*. Since higher sensitivity means lower chaos, we use *Shannon Entropy*'s reciprocal to define the sensitivity. *Shannon Entropy* of a discrete random variable X with possible values $\{x_1, x_2...\}$ and probability mass function $P(X)$ is defined as $H(X)$, which can be explicitly be written as

$$H(X) = -\sum_i P_i \log(P_i). \tag{3}$$

In our situation, the set of users who have visited L_θ at T_t^w is denoted as U_{L_θ,T_t^w}. For each user $u \in U_{L_\theta,T_t^w}$, we calculate the proportion of the number of visits to the total visit times of (L_θ, T_t^w) as follows,

$$P_{L_\theta,T_t^w}(u) = \frac{V_u}{\sum_{u \in U_{L_\theta,T_t^w}} V_u}, \tag{4}$$

where V_u denotes the number of visit times of user u at (L_θ, T_t^w). With each user's proportion of visit times, we calculate *Shannon Entropy* and use its reciprocal to represent the sensitivity, denoted as s_{L_θ,T_t}, as follows,

$$s_{L_\theta,T_t} = \frac{1}{-\sum_{u \in U_{L_\theta,T_t^w}} P_{L_\theta,T_t^w}(u) \log(P_{L_\theta,T_t^w}(u))}. \tag{5}$$

As an example in Figure 5 illustrated, at time slot T_1^w and location L_1, there are three distinct visitors who visited this location three times, twice and once, respectively. Their proportions of visit times are 3/6, 2/6 and 1/6. According to (5), $s_{L_1,T_1} = 0.6309$.

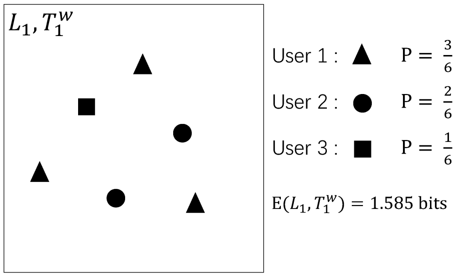

L_1, T_1^w
User 1 : ▲ P = $\frac{3}{6}$
User 2 : ● P = $\frac{2}{6}$
User 3 : ■ P = $\frac{1}{6}$
$\mathrm{E}(L_1, T_1^w) = 1.585$ bits

Fig. 5: An example of co-occurrence sensitivity, where $E(L_\theta, T_t^w)$ denotes the entropy of (L_θ, T_t^w).

Then, we add up the co-occurrence sensitivity to represent the correlation between the trajectories of a and b, defined as S_{ab}, can be expressed as follows,

$$S_{ab} = \sum_{t \in T_{co}} s_{L_\theta,T_t}, \tag{6}$$

where T_{co} represents the time slots of co-occurrence between user a and user b. Thus, S_{ab} includes not only the number but also the sensitivity of the co-occurrence. A large value of S_{ab} indicates that a and b's trajectories are strongly correlated.

2) Social-aware Distance: In order to prevent social relationship attack and re-identification attack simultaneously, we use customized generalization methods, which means merging carefully chosen trajectories. To this end, we propose a new distance metric, *i.e.*, social-aware distance, between trajectories. First, we calculate the spatiotemporal distance that is crucial for maintaining data utility while generalizing trajectories, because merging spatiotemporal close trajectories brings less utility loss. Then, we introduce the correlation between friends' trajectories to the spatiotemporal distance, which is the key point of social relationship protection.

Since the trajectory we discuss is temporal discrete, the spatiotemporal distance between trajectories is the sum of spatial distance of each time slot. Let us consider the t-th time slot (T_t) in the trajectories of user a and b. We utilize L_a and L_b to denote their location sets at this time slot, and n_a and n_b represents the number of locations in L_a and L_b, separately. In the original data, L_a and L_b only contain at most one specific location. However, as the anonymization process continues, a and b may already be a merged trajectory consisting of more than one user, thus L_a and L_b contain more than one location. Formally, the spatial distance between L_a and L_b is computed as follows,

$$d^t_{a,b} = \frac{1}{n_a \cdot n_b} \cdot \sum_{l_a \in L_a, l_b \in L_b} B(l_a, l_b) \ , \tag{7}$$

where $B(l_a, l_b)$ is the Squared Euclidean Distance between l_a and l_b. Thus, the spatiotemporal distance between trajectories is $D_{a,b} = \sum_{t=1}^{M} d^t_{a,b}$.

Based on the spatiotemporal distance, we define the social-aware distance. For each point in trajectory, higher sensitivity indicates that the visitors are less diverse at (L_θ, T_t^w). We don't want to merge these kinds of points of two trajectories when they belong to friends. Thus, we add the sensitivity factor to the spatiotemporal distance of two points. Formally, the social-aware distance of points between user a and b at t-th time slot is $\omega^t_{a,b} = (1+\alpha \cdot s_{L_\theta, T_t}) \cdot d^t_{a,b}$, where α is the weight coefficient of sensitivity. Now, we calculate social-aware distance of trajectories by adding up point distance of all the time slots. However, merging trajectories that belong to close friends will not reduce the correlation between them. Thus, we add the intensity of the social relationship, which is defined to distinguish strong and weak relationships, to the social-aware distance. For all the friend pairs, we calculate the intensity of their relationship given their trajectories' correlation using *Bayes' theorem* as follows,

$$P(F=1|S) = \frac{P(S|F=1) \cdot P(F=1)}{P(S)}, \tag{8}$$

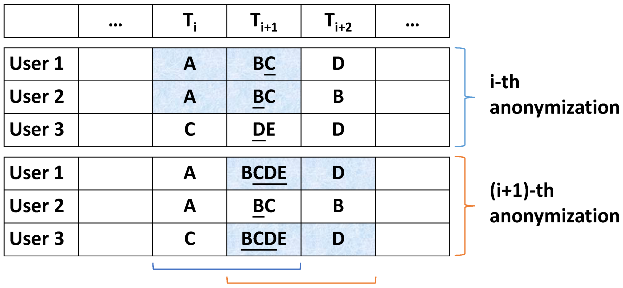
where $F = 1$ means they are friends and 0 means they are not friends, S means the correlation between their trajectories (see Equation 6). $P(S)$ is the proportion of user pairs that have trajectory correlation of S in all user pairs and $P(S|F = 1)$ is the proportion of user pairs that have trajectory correlation of S in all friend pairs. Finally, we compute the trajectory distance as

$$\Omega_{a,b} = (1 + \beta \cdot P(F=1|S)) \cdot \sum_{t=1}^{N} \omega^t_{a,b}, \tag{9}$$

where β is the weight coefficient of relationship intensity.

Suppose a and b are friends. If their relationship is close, their social-aware distance is larger and less likely to be merged into one trajectory. The reason is that we add the intensity of social relationship to trajectory distance and the sensitivity to point distance. In this way, we combine users' spatiotemporal distance and social distance together and then we are able to prevent social relationship attack and re-identification attack simultaneously.

3) SlidingWindow Algorithm: We propose a SlidingWindow algorithm to generalize trajectories according to their social-aware distance. Trajectory generalization will reduce data utility. Especially after adding the trajectory correlation to the distance, spatiotemporal similar trajectories of close friends will have greater social-aware distance, which indicates that they are going to merge with other trajectories and lead to more utility loss. To tackle this challenge we design a SlidingWindow algorithm to apply k^m-anonymity rather than full-length k-anonymity to reduce data utility loss. The key idea of SlidingWindow algorithm is that each time we anonymize continuous m time slots of all the trajectories and then move one time slot ahead, until the end of the trajectories. It looks like moving a sliding window showing in Figure 4. The advantage of SlidingWindow algorithm is that at different windows, the trajectory can be merged with different k-1 other trajectories. On the one hand, it causes less utility loss because short-length trajectories have closer distance than long-length trajectories. On the other hand, it will not increase the correlation between friends' full-length trajectories, because friends are merged at a very limited number of windows and at most windows they merge with non-friends, respectively.



	...	T_i	T_{i+1}	T_{i+2}	...
User 1		A	BC	D	
User 2		A	BC	B	
User 3		C	DE	D	
User 1		A	BCDE	D	
User 2		A	BC	B	
User 3		C	BCDE	D	

i-th anonymization
(i+1)-th anonymization

Fig. 6: An example of anonymizing two adjacent sliding windows. A,B,C,D,E are different locations and location with a underline means it is added after generalization (m=2, k=2).

One problem we need to consider is that anonymizing sub-trajectories in a window will change the former m-1 adjacent windows which have already achieved k-anonymity. Because adjacent windows have overlapping time slots. We solve this problem by only utilizing generalization as the anonymization technique, which means we only add locations rather than delete or change locations. Thus, a generalized location still contains the original location and the sub-trajectories in former windows are still k-anonymized. For example, in Figure 6, we apply 2^2-anonymity on trajectories, which means the length of the window is 2 and the size of each anonymous set should be at least 2. T_i means the i-th time slot. We discuss the three users in two adjacent time slots. At the i-th window, user 1 and user 2 are in the same anonymous set and the merged sub-trajectory is $[A, BC]$. Then, at the next window, user 1 is merged with user 3 and the sub-trajectory of user 1 and user 3 turns to be $[BCDE, D]$, which results in that user 1's location $(BCDE)$ at T_{i+1} is no longer the same with user 2's (BC).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2907542, IEEE Transactions on Network and Service Management

However, as we use only generalization method, they both contain BC. Thus, if the attacker has the external information of the victim's (user 1) locations at T_i and T_{i+1}, *i.e.*, $[A, B]$, he will obtain two users (user 1 and user 2) by matching the records with the dataset, that is to say, he cannot uniquely identify user 1.

Then we introduce the details of anonymizing sub-trajectories in a sliding window. The algorithm framework is detailed in Alg. 1, whose inputs are the sub-trajectory dataset $\mathbb{D}$ and the value of k ,*i.e.*, the target k-anonymity level and output is the anonymized sub-trajectory dataset $\mathbb{G}$. The input dataset $\mathbb{D}$ contains a series of discrete trajectories. Anonymized trajectories in the output dataset $\mathbb{G}$ have the same time accuracy as the input trajectories. But each time slot might have several nearby locations, which is a kind of spatial generalization. There are three steps in the algorithm: #1) calculate the social-aware distance of each sub-trajectory pair according to (9) and store it in a matrix Ω (line 1-3), #2) cluster no less than k nearest sub-trajectories together as k-anonymous sets, whose details are shown in Alg. 2, and #3) merge sub-trajectories in the same k-anonymous set into one sub-trajectory and add it into the generalized dataset $\mathbb{G}$. The details are shown in Alg. 3.

In order to reduce utility loss when merging sub-trajectories, we cluster close sub-trajectories together in Alg. 2. The input is the social-aware distance matrix Ω and the output is the k-anonymous sets. Since we have already calculated the social-aware distance between each pair of sub-trajectories, we propose an agglomerative clustering method to group sub-trajectories with close trajectories. Agglomerative clustering is flexible to stop when the cluster size reaches k. The size of each anonymous set should not be too large, because merging more sub-trajectories tends to cause larger utility loss. In Alg. 2, we first initialize $Clusters$ and $FinalClusters$ to store clusters with size less than k and larger than k, respectively. The $while$ loop runs until $Clusters$ is empty, which means all the clusters' size is no less than k. At each iteration, the two sub-trajectory clusters that have not yet been k-anonymized and have the minimum distance are identified and combined into θ (line 4,5). If the size of θ reaches k, θ is added to $FinalClusters$ (line 8). Otherwise, it is added to $Clusters$ and the distance between θ and other location sets in $Clusters$ is added to Ω (line 12). The distance between two clusters is the average distance between their trajectories.

At the final step, we merge all the sub-trajectories in the same cluster into one sub-trajectory. As shown in Alg. 3, the inputs are a cluster c and sub-trajectory dataset, and the output is the merged sub-trajectory. We initialize the merged sub-trajectory as $Mtraj$ (line 1). Then we merge the sub-trajectories one by one. Since the sub-trajectories are temporal discrete trajectories, we only need to merge locations in each time slot in that window (T) while merging trajectories (line 4-8). If both locations of two sub-trajectories are not empty, the merged location is the union of these locations. $\mathbb{D}[i][t]$ means u_i's location at time slot T_t. However, if the location is empty (N/A), this time slot will be filled with L^* which is the union of all locations. For example, if the dataset is collected in a campus, then L^* is the campus.

Algorithm 1 Anonymizing sub-trajectories

```
Input: Sub-trajectory Dataset $\mathbb{D}$, Anonymity Criterion $k$
Output: Generalized Sub-trajectory Dataset $\mathbb{G}$
1: for $a, b \in \mathbb{D}, a \neq b$ do        ▷ Calculating Social-aware distance
2:     $\Omega(a, b) \leftarrow getTrajDist(a, b)$
3: end for
4: Clusters $\leftarrow Clustering(\Omega)$        ▷ Clustering
5: $\mathbb{G} \leftarrow \{\}$        ▷ Merging sub-trajectories
6: for $c \in$ Clusters do
7:     $add(\mathbb{G}, Merging(c, \mathbb{D}))$
8: end for
```

Algorithm 2 Clustering close sub-trajectories

```
Input: Social-aware distance matrix $\Omega$
Output: $k$-anonymous sets: FinalClusters
1: Clusters $\leftarrow \{\{1\}, \{2\}, \{3\}, ..., \{N\}\}$
2: FinalClusters $\leftarrow \{\}$
3: while size(Clusters) $> 0$ do
4:     $i, j \leftarrow argmin(\Omega)$
5:     $\theta \leftarrow$ {Clusters[$i$], Clusters[$j$]}
6:     $delete$(Clusters[$i$]), $delete$(Clusters[$j$])
7:     if $size(\theta) \geq$ k then
8:         $add$(FinalClusters, $\theta$)
9:     else
10:        $add$(Clusters, $\theta$)
11:        for Clusters[$i$] $\in$ Clusters do
12:            $add(\Omega, getClusDist($Clusters[$i$]$, \theta))$
13:        end for
14:    end if
15:    $delete(\Omega, i)$, $delete(\Omega, j)$
16: end while
```

In summary, we utilize a window sliding from the head of a trajectory to the end to anonymize the whole trajectory step-by-step. At each step, we first calculate the social-aware distance between each pair of sub-trajectories and then cluster the close sub-trajectories together as a k-anonymous set. Finally, trajectories in the same set are merged into one. By this way, anonymized trajectories meet the requirements of both social relationship protection and k^m-anonymity, and preserve more data utility.

Algorithm 3 Merging sub-trajectories in the same k-anonymous set

```
Input: Cluster $c$, Sub-trajectory Dataset $\mathbb{D}$
Output: Generalized Sub-trajectory: Mtraj
1: $Mtraj \leftarrow$ (N/A, ..., N/A)
2: for $i \in c$ do
3:     for $t \in T$ do
4:         if Mtraj[t]!=N/A and $\mathbb{D}[i][t]$!=N/A then
5:             Mtraj[t] $\leftarrow$ {Mtraj[t], $\mathbb{D}$[i][t]}
6:         else
7:             Mtraj[t] $\leftarrow L^*$
8:         end if
9:     end for
10: end for
```

C. Complexity Analysis

Assuming the dataset has N users and the length of the trajectory is M. At each sliding window, the computational complexity is from three steps of the algorithm. First, calculating social-aware distance between every two trajectories includes calculating the distance of m pairs of spatial points. As there are $N^2/2$ pairs of trajectories, the computational complexity is $O(N^2)$. Second, as each step of clustering needs to calculate the distance between the new cluster and the others, the time complexity is $O(N^2)$. Third, merging every two trajectories includes merging m dyads spatial points. Total merging time is M, which means the time complexity is $O(M)$. Overall, the total time complexity is $O(MN^2)$, which is linear in the trajectory length and quadratic in the number of users. This complexity is competitive in the area of trajectory dataset anonymization. For example in [26], the computational complexity of the proposed protection method is $O(N^2M^2)$, which only deals with the re-identification attack problem.

V. Evaluation

We evaluate the performance of our social relationship protection system under two different scenarios: social relationship of students on a campus which covers a small and closed area, and social relationship of mobile users in a city which covers a large and open area. Due to the difference of spatial coverage and user groups, the underlying relationship characteristics are different. In the campus, most of the users are students who have close daily interactions, thus the main relationship between them is friendship. While in a metropolitan city, all the subscribers are citizens and they have complex relationships. Therefore, we use the social tie to represent the relationship among them. In these two scenarios, individual trajectory data is collected from different networks: Wi-Fi network and cellular network. Therefore, we call these two datasets as **Wi-Fi dataset** and **Cellular dataset**, whose key features are summarized in Table II. For both datasets, we have taken the following steps to ensure the ethical considerations to deal with such sensitive data: firstly, all mobile users' identifiers are replaced with random sequences to achieve anonymizations; secondly, we store all the data in a local secure server; thirdly, only the core researchers regulated by the strict non-disclosure agreements have access to the data.

To evaluate the effectiveness of our system in preventing friendship and social tie leakage, we launch the most efficient attack systems of MLI [21] and CAI [18] to evaluate the accuracy of social relationship identification.

TABLE II: Major information and key features of two utilized mobility datasets.

Datasets & Metrics	Wi-Fi Dataset	Cellular Dataset
Source	Wi-Fi network	cellular network
Location	Tsinghua Univeristy, China	Shanghai, China
Time	Nov. 2015 - Feb. 2016	Apr. 2016
Duration	sixteen weeks	one week
User number	10,162	5.90 millions

A. Preventing Friendship Leakage

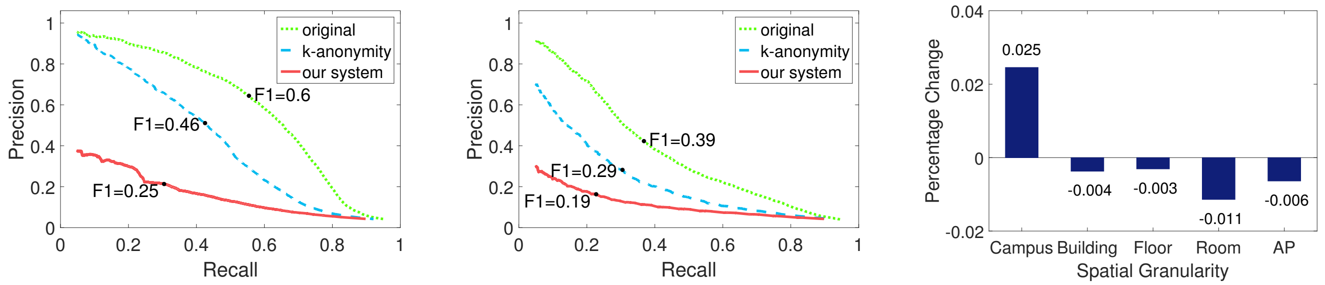
1) Wi-Fi Dataset and Metrics: This dataset is collected by polling the device association and probing logs from 2699 access point (APs) in Tsinghua University via SNMP. The detailed collection method is introduced in [33]. Since each AP in the campus is carefully named using its semantic location (*building-floor-room-AP*), we can obtain the individual's location by analyzing the AP name. In addition, we calculate the number of APs in the minimum area covering two locations and regard it as the relative distance between them. To obtain reliable friendship, we select 612 users whose class and dormitory are both accessible, and regard classmates and roommates as friends, which forms our groundtruth of friendship. In addition, in order to adapt the dataset to our system, we group all records in each hour time slot and extract the most frequently visited location of every mobile user in each time slot to form a temporal discrete trajectory dataset.

Generally speaking, for an anonymization algorithm, the privacy level and data utility are two basic metrics to be considered.

- We utilize parameter k of k-anonymity to indicate the privacy level against re-identification attack and the **Precision-Recall Curve** to show the friendship disclosure risk, with $Precision$ meaning the percentage of accurately inferred friend pairs in all the inferred friend pairs and $Recall$ representing the percentage of accurately inferred friend pairs in all the actual friend pairs.
- For data utility, since the temporal granularity does not change after generalization, we only discuss the spatial utility loss. For Wi-Fi dataset, the location granularity is hierarchical ($building$-$floor$-$room$-AP) and the original spatial granularity is AP level. To better display the output trajectories, if a trajectory point contains multiple possible locations, they will be represented by a higher level location that covers all of them. For example, if one output trajectory point contains several APs in a room then we use this room to represent its location. Thus, through spatial generalization, the granularity of some records may change to room level, floor level, building level or even campus level. Therefore, we use the percentage changes of records with different spatial resolutions to measure the spatial granularity loss.

2) Results and Analysis: We evaluate the performance of our system in friendship leakage risk and data utility preserving. We set k=4 and m=8, and compare with the performances of only achieving k-anonymity. The k-anonymity algorithm is the same as our algorithm without the social component.

We launch MLI and CAI with different extents of protection: raw data without any protection (original), anonymized data protected by k-anonymity (k-anonymity) and anonymized data protected by our system (our system). We plot the Precision-Recall Curves in Figure 7(a) and Figure 7(b), respectively. As shown in Figure 7(a), though both our system and k-anonymity algorithm reduce the friendship identification accuracy compared with the original data, the amount of decreased accuracy is quite different. Under different recall rates, the precision rate of our system is much lower than that of the original data. In addition, the average decreased precision rate is almost 2 times when compared with k-anonymity algorithm. Moreover, we find the maximum F1-scores of Precision-Recall Curves, respectively, which represent the worst situation



(a) MLI (b) CAI (c) Data utility loss

Fig. 7: Friend identification accuracy and data utility loss of our system compared with k-anonymity on Wi-Fi dataset (k = 4, m = 8).

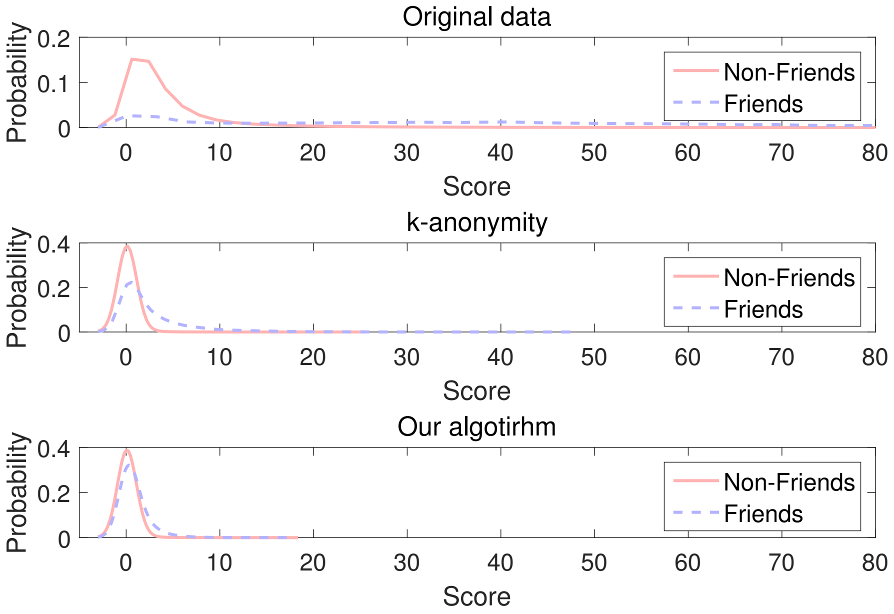


Fig. 8: The probability distribution of friends and non-friends' scores calculated by MLI on the original Wi-Fi dataset and dataset under different protection methods (k = 4, m = 8).

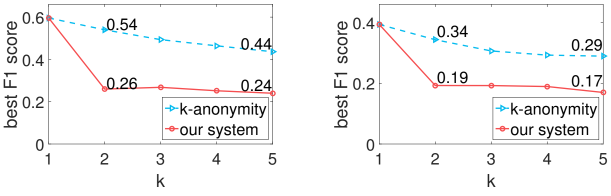
of friendship leakage in these three datasets. The F1-scores shows that our system provides 1.84 (0.46/0.25) times better friendship protection than k-anonymization even in the worst case. Our system also shows great performance against CAI in Figure 7(b).

In term of data utility represented by spatial granularity, our system decreases the granularity slightly compared with k-anonymity. Figure 7(c) shows the percentage change of records with different spatial resolutions. From the results, we can observe that we lose some fine-grained records and add more coarse-grained records. Records at AP level decrease 0.6% and records at room level reduce 1.1% while records at campus level increase 2.5% respectively, all the percentage changes are very small. In other words, we only add 2.5% coarse-grained records, leading to extra loss of data utility. The strategy of decreasing the trajectory similarity of friends and our flexible criterion of k^m-anonymity contribute to the small loss of data utility in preventing friendship leakage.

We also compare the probability density function (PDF) of correlation scores between friends and non-friends under different protection schemes in Figure 8, which shows the underlying reasons for the effectiveness of our system. In original data, the scores of non-friends are highly concentrated and close to zero, while the scores of friends are variant and the distribution has a long tail. The attacker can distinguish friends and non-friends since they have different distributions, causing serious privacy leakage. After k-anonymity protection, the long-tail effect of scores between friends eliminates but its distribution is still quite different from that of non-friends. However, our system makes the distribution of scores between friends and non-friends very similar, thus the attacker cannot tell if two users are friends based on this score.

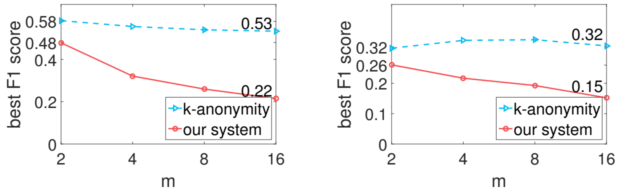
Since our system meets the requirement of k^m-anonymity, both k and m are key factors to influence the results. We evaluate our system's performance in preventing friendship identification when k and m vary, respectively.

Influence of k. k is the minimum size of anonymity set required to prevent re-identification. Larger k guarantees a higher privacy level. In order to evaluate the influence of k on friendship identification accuracy, we launch the same attack to Wi-Fi dataset and compare the results of k-anonymity and our system when k ranges from 1 to 5 as [26] showing



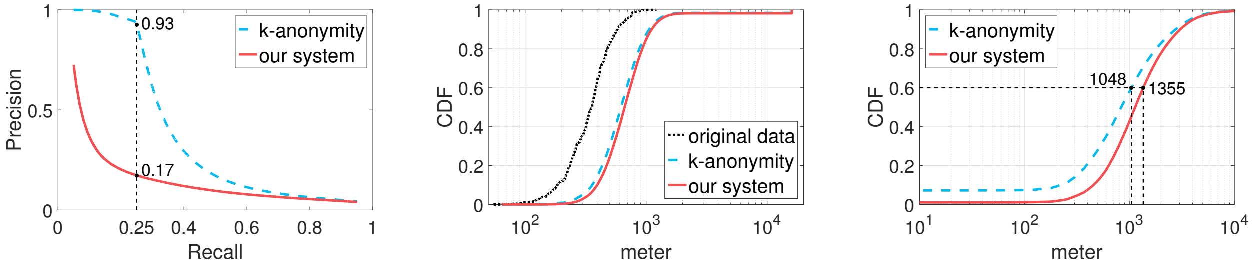
(a) MLI (b) CAI

Fig. 9: The influence of k on friend identification accuracy (m=8).



(a) MLI (b) CAI

Fig. 10: The influence of m on friend identification accuracy (k=2).



(a) MLI (b) Spatial Granularity (c) Offset Distance

Fig. 11: Social-tie identification accuracy and data utility loss of our system compared with k-anonymity on cellular dataset (k = 2, m = 8).

that the anonymized dataset becomes hardly exploitable when $k > 5$. Note that the dataset refers to original data without any protection when k=1. We show the results in Figure 9. Figure 9(a) shows that the performance of our system against MLI is outstanding and stable while that of k-anonymity is greatly influenced by k. When k increases from 2 to 5, the maximum F1 Score of k-anonymity only decreases from 54% to 42%, but the maximum F1 Score of our system remains a low level at 26%. Figure 9(b) shows similar results on another relationship attack CAI, the maximum F1 score of our system, representing friendship identification accuracy, is much lower than k-anonymity with the same parameters. The results indicate that larger k cannot provide extra friendship protection. Thus only achieving k-anonymity is not enough to prevent friendship leakage. That's why a new privacy model and algorithm are needed, and our system just meets such requirement.

Influence of m. m is the number of continuous spatiotemporal points in each sliding window, larger m guarantees higher privacy level. In order to evaluate the influence of m on friendship identification accuracy, we do the same experiment when m ranges from 2 to 16 and the results on AP dataset are shown in Figure 10. Figure 10(a) shows that increasing m can effectively decrease the accuracy of friendship identification. When m ranges from 2 to 16, the maximum F1 Score of our system decreases from 48% to 25% while the maximum F1 Score of k-anonymity decreases little and the accuracy is still above 50% when m is 16. We can find similar results in Figure 10(b). The results indicate that m has a great impact on the performance of our system, the underlying reason may be larger m makes trajectories between strangers more similar and the attacker may mistake many strangers for friends, thus greatly decreasing the accuracy of friendship identification.

In conclusion, our system shows outstanding performance compared with the state-of-art algorithm in the campus scenario. Our system greatly decreases friendship identification accuracy and still preserves the data utility after anonymization.

B. Preventing Social-tie Leakage

1) Cellular Dataset and Metrics: This dataset is collected by a major mobile service provider in Shanghai, one of the major metropolitan in China. When the user accesses the cellular network, i.e., making phone calls, sending texts, or consuming data plan, the connected base station and timestamp are recorded. Each base station covers a polygonal area generated by voronoi diagram. According to the friendship attack (MLI) results on Wi-Fi dataset, we choose the same percentage of pairwise friends with high scores on the cellular dataset and regard them as our groundtruth. We perform the same preprocessing on the cellular dataset with 30 minutes time slots. Since the time duration of the cellular dataset is one week, we use a higher sampling rate than Wi-Fi dataset to obtain more spatiotemporal points in each trajectory. Since many users only have very limited points in the trajectory, we select five thousand active users from the dataset.

For the cellular dataset, the metrics of measuring privacy level is the same with Wi-Fi dataset and as for the utility, we also only consider spatial granularity loss. The location of an original record in the cellular dataset is a base station. As we perform spatial generalization by adding extra base stations, the output location may contain several nearby base stations. Thus, we measure the spatial granularity loss from two aspects: the square root of area of the generalized location and the generalized location deviates from its original position, named as *Spatial Granularity* and *Offset Distance*,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2907542, IEEE Transactions on Network and Service Management

respectively. Obviously the coarser the spatial granularity and the larger offset distance, the larger the data utility loss.

2) Results and Analysis: We launch MLI to the cellular dataset and compare the privacy-preserving level of anonymized data protected by k-anonymity and anonymized data protected by our system. We plot the Precision-Recall Curves in Figure 11(a). It shows our system greatly decreases social-tie identification accuracy and provides considerable protection against social relationship attack when compared with k-anonymity. Under different recall rate, the precision rate of our system is much lower than k-anonymity. When the recall rate grows, the precision rate of our system drops rapidly. When the recall rate is 25%, the precision rate of our system is only 17%, decreasing by 76% compared with k-anonymity, whose precision rate is as high as 93%.

Besides social-tie identification accuracy, we also analyze the changes in spatial granularity. We utilize spatial granularity and offset distance to measure data utility loss, as shown in Figure 11(b) and Figure 11(c) respectively. Apparently, coarser spatial granularity and larger offset distance mean larger data utility loss. From the figures, our system only causes a little more utility loss than k-anonymity algorithm. In Figure 11(b), compared with original data, both k-anonymity and our system cause some spatial granularity loss due to spatial generalization to meet privacy protection requirements. In addition, the CDF of different grained records of our system is very close to that of k-anonymity algorithm, which means we effectively limit extra granularity loss to meet the requirement of social-tie protection. In Figure 11(c), 60% of the records have an offset distance within 1355 m, only 307 m larger than k-anonymity, showing that our generalized location is close to the original location and rather accurate. The results indicate that our system also provides enough protection to social ties, a major correlation between citizens, while still preserves high data utility.

C. Summary

In summary, our system shows more than 1.1 times better performance than the state-of-the-art algorithms that prevent friendship and social ties from leakage. We demonstrate two advantages of our system: 1) **Adaptation**: Wi-Fi and Cellular dataset come from different sources with different spatial resolution. However, a small modification is enough to adopt our system, which provides considerable protection to both the datasets. 2) **Robustness**: two different datasets contain different groups of people and different underlying relationships. In our evaluation, both friendship and social ties are well protected by our system and the attacker can only infer very limited information from our anonymized dataset through two different relationship attacks. These show our system can effectively prevent different kinds of relationship attack and related privacy leakage.

VI. Related Works

In this section, we summarize the relevant works from three perspectives – social relationship attack, mobility data protection and differential privacy.

Social Relationship Attack: The first type of social relationship attack is directly inferring social relationship from trajectories, which has attracted significant attention in the past decade, *e.g.*, [18–25]. [19] infers friendships only from users' co-occurrence in time and space. [22, 23] consider more factors of meeting events, such as location entropy and time intervals. Others [18, 20, 21] utilize semantic information, such as the location types, of co-occurrence to infer social relationship from trajectories. More recent works [24, 25] utilized advanced techniques. [24] presents a novel neural network model which can jointly model both social networks and mobile trajectories. [25] proposes to construct a user graph based on their spatiotemporal interactions and employ graph embedding technique to learn user representations, which can well describe mobility relationship. However, all these studies inferring social relationship based on trajectory correlations which are decoupled by our algorithm. The other type of social relationship attack is utilizing social relationship to do attacks, *i.e.*, [28–31]. [31] de-anonymizes trajectory dataset with a social network utilizing the similar structure of these two networks of users. [28] predicts the location of an individual given the known locations of her friends. In summary, these works all stand in the angle of an attacker, but we are in the perspective of data publisher and try to prevent potential privacy leakage caused by social relationship of users in the trajectory dataset.

Mobility Data Protection: Mobility data privacy has been discussed mainly in two different situations: location-based services (LBS) and trajectory. A large number of studies have targeted at user privacy in LBS. The goal is ensuring that single georeferenced queries are not uniquely identifiable by utilizing techniques like spatiotemporal generalization [34], encryption [35], deception [36], etc. But protecting the whole trajectory is more difficult than a single point. Trajectory anonymization is an indispensable step of trajectory dataset publishing. Most recent studies achieve k-anonymity on both spatial [37–40] and spatiotemporal [26, 41, 42] trajectory. [42] achieves (k, δ)-anonymity in trajectory dataset. It requires the whole trajectory should be hidden in other k-1 trajectories with a spatial threshold δ. However, it violates the principles of PPDP by perturbing or permuting the trajectories. [39] achieves k^m-anonymity, which is different from our definition of k^m-anonymity. It deals with ordered lists of locations rather than trajectories with timestamps. Their solution bases on a frequent item set mining algorithm - *apriori* and the output satisfies that each m-length sublist of locations should be contained by at least k distinct users while our m means the length of the SlidingWindow. Besides, due to its high complexity, it only works on very short spatial trajectories of several samples each, whereas trajectories typically include hundreds of samples per week. [41] proposes a similar criterion $k^{\tau,\epsilon}$-anonymity in continuous spatiotemporal trajectory dataset to protect trajectory from being re-identified. We have the similar idea of merging trajectories partially but achieve it with a different algorithm. These methods only consider the re-identification attack by decreasing the trajectories' uniqueness but ignore their correlations, which is the main contribution of our work. Some studies go beyond k-anonymity, for example,

[43] achieves k-anonymity, l-diversity and t-closeness, which will be considered in our future work.

Differential Privacy: At the last remark, differential privacy [44] is a framework technique for privacy protection. It demands that for a query in a dataset, adding or removing a single record of a user does not cause a significant difference. It is not suitable for trajectory data publishing scenario for two reasons. First, differential privacy focuses on relational data queries and only gives aggregated information [45]. Second, differential privacy can be achieved only by randomized mechanisms [46], *e.g.*, adding noise, which is conflicting to our targeted principle of preserving data truthfulness in PPDP.

VII. Conclusion

In this paper, we identify a serious privacy problem of social relationship leakage due to the correlation between trajectories, and recognize the need for preventing social relationship attack at the scenario of trajectory data publishing. To the best of our knowledge, we are the first to propose a privacy model and an effective system to prevent both social relationship attack and re-identification attack by reducing the correlation of trajectories and their uniqueness as well. Based on two real-world datasets, extensive evaluation demonstrates that our system successfully protects social relationships while preserving considerable utility in different scenarios, which provides more than 1.84 times privacy protection at the cost of only 2.5% data utility loss. We believe that this work opens a new angle of protecting the privacy leakage caused by the correlation between trajectories in mobility data publishing, which paves the way to more advanced privacy preserving mechanisms.

References

[1] S. Isaacman, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger, “Human mobility modeling at metropolitan scales,” in *ACM MOBISYS*, 2012.

[2] M. Seshadri, S. Machiraju, A. Sridharan, J. Bolot, C. Faloutsos, and J. Leskove, “Mobile call graphs: beyond power-law and lognormal distributions,” in *ACM KDD*, 2008.

[3] Y. Wang, H. Zang, and M. Faloutsos, “Inferring cellular user demographic information using homophily on call graphs,” in *IEEE INFOCOM WKSHPS*, 2013.

[4] L. Chen, A. Mislove, and C. Wilson, “Peeking beneath the hood of uber,” in *ACM Conference on Internet Measurement Conference*, 2015.

[5] F. Xu, P. Zhang, and Y. Li, “Context-aware real-time population estimation for metropolis,” in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 1064–1075.

[6] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma, “Geolife2.0: a location-based social networking service,” in *IEEE Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, 2009, pp. 357–358.

[7] Y. Zheng, X. Xie, and W.-Y. Ma, “Geolife: A collaborative social networking service among user, location and trajectory.” *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.

[8] M. Suzuki, T. Kitahara, S. Ano, and M. Tsuru, “Group mobility detection and user connectivity models for evaluation of mobile network functions,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 127–141, 2018.

[9] H. Jiang, S. Yi, L. Wu, H. Leung, Y. Wang, X. Zhou, Y. Chen, and L. Yang, “Data-driven cell zooming for large-scale mobile networks,” *IEEE Transactions on Network and Service Management*, 2018.

[10] D. Ding, M. Zhang, X. Pan, D. Wu, and P. Pu, “Geographical feature extraction for entities in location-based social networks,” in *World Wide Web Conference on World Wide Web*, 2018, pp. 833–842.

[11] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, “Unique in the crowd: The privacy bounds of human mobility,” *Scientific reports*, vol. 3, p. 1376, 2013.

[12] Y.-A. De Montjoye, L. Radaelli, and V. K. Singh, “Unique in the shopping mall: On the reidentifiability of credit card metadata,” *Science*, vol. 347, no. 6221, pp. 536–539, 2015.

[13] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[14] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[15] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in *IEEE ICDE*, 2007, pp. 106–115.

[16] http://edition.cnn.com/2011/11/07/opinion/crump-gps/index.html.

[17] https://www.washingtonpost.com/news/the-switch/wp/2013/12/10/new-documents-show-how-the-nsa-infers-relationships-based-on-mobile-location-data/?utm_term=.657853556800.

[18] N. Eagle, A. S. Pentland, and D. Lazer, “Inferring friendship network structure by using mobile phone data,” *Proceedings of the national academy of sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[19] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, “Inferring social ties from geographic coincidences,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 52, pp. 22 436–22 441, 2010.

[20] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, “Bridging the gap between physical location and online social networks,” in *ACM international conference on Ubiquitous computing*, 2010, pp. 119–128.

[21] R. Pasqua, M. Roy, and G. Tredan, “Loca: a location-oblivious co-location attack in crowds,” in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 535–544.

[22] H. Wang, Z. Li, and W.-C. Lee, “Pgt: Measuring mobility relationship using personal, global and temporal factors,” in *IEEE International Conference on Data Mining (ICDM)*, 2014, pp. 570–579.

[23] R. Cheng, J. Pang, and Y. Zhang, “Inferring friendship from check-in data of location-based social networks,” in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2015.

[24] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, “A neural network approach to jointly modeling social networks and mobile trajectories,” *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, p. 36, 2017.

[25] Y. Yu, H. Wang, and Z. Li, “Inferring mobility relationship via graph embedding,” *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 147, 2018.

[26] M. Gramaglia and M. Fiore, “Hiding mobile traffic fingerprints with glove,” in *ACM Conference on Emerging Networking Experiments and Technologies*, 2015, p. 26.

[27] 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, 2010.

[28] L. Backstrom, E. Sun, and C. Marlow, “Find me if you can: improving geographical prediction with social and spatial proximity,” in *ACM international conference on World wide web*, 2010, pp. 61–70.

[29] A. Sadilek, H. Kautz, and J. P. Bigham, “Finding your friends and following them to where you are,” in *ACM international conference on Web search and data mining*, 2012, pp. 723–732.

[30] A.-M. Olteanu, K. Huguenin, R. Shokri, and J.-P. Hubaux, “Quantifying the effect of co-location information on location privacy,” in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2014, pp. 184–203.

[31] M. Srivatsa and M. Hicks, “Deanonymizing mobility traces: Using social network as a side-channel,” in *ACM conference on Computer and communications security*, 2012, pp. 628–637.

[32] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, “Privacy-preserving data publishing,” *Foundations and Trends® in Databases*, vol. 2, no. 1–2, pp. 1–167, 2009.

[33] M. Zhou, M. Ma, Y. Zhang, K. SuiA, D. Pei, and T. Moscibroda, “Edum: classroom education measurements via large-scale wifi networks,” in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 316–327.

[34] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking,” in *ACM international conference on Mobile systems, applications and services*, 2003.

[35] M. Herrmann, A. Rial, C. Diaz, and B. Preneel, “Practical privacy-preserving location-sharing based services with aggregate statistics,” in *ACM conference on Security and privacy in wireless & mobile networks*, 2014, pp. 87–98.

[36] L. Zhang, Z. Cai, and X. Wang, "Fakemask: a novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[37] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," in *ACM GIS International Workshop on Security and Privacy in GIS and LBS*, 2008, pp. 52–61.

[38] A. Monreale, G. L. Andrienko, N. V. Andrienko, F. Giannotti, D. Pedreschi, S. Rinzivillo, and S. Wrobel, "Movement data anonymity through generalization." *Trans. Data Privacy*, vol. 3, no. 2, 2010.

[39] G. Poulis, S. Skiadopoulos, G. Loukides, and A. Gkoulalas-Divanis, "Apriori-based algorithms for kˆ m-anonymizing trajectory data," *Transactions on Data Privacy*, vol. 7, no. 2, pp. 165–194, 2014.

[40] R. Yarovoy, F. Bonchi, L. V. Lakshmanan, and W. H. Wang, "Anonymizing moving objects: How to hide a mob in a crowd?" in *ACM International Conference on Extending Database Technology: Advances in Database Technology*, 2009, pp. 72–83.

[41] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs, "Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories," pp. 1–9, 2017.

[42] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *IEEE ICDE*, 2008.

[43] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity and t-closeness," *IEEE Transactions on Network and Service Management*, 2018.

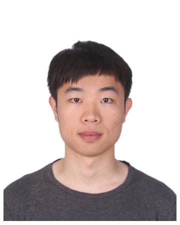
[44] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*. Springer, 2006, pp. 265–284.

[45] A. Monreale, W. H. Wang, F. Pratesi, S. Rinzivillo, D. Pedreschi, G. Andrienko, and N. Andrienko, "Privacy-preserving distributed movement data aggregation," in *Geographic Information Science at the Heart of Europe*, 2013, pp. 225–245.

[46] A. Machanavajjhala, J. Gehrke, and M. Götz, "Data publishing against realistic adversaries," *VLDB Endowment*, vol. 2, no. 1, 2009.

Kai Zhao Kai Zhao received his B.S. degree in Electronic Engineering from Tsinghua University, Beijing, China, in 2016, and he is currently working towards the master degree in electronic engineering department of Tsinghua University, Beijing, China. His research interests includes human mobility, mobility data privacy and recommendation system.

Zhen Tu received her B.S. degree in Electronics and Information Engineering and a second B.S. degree in Economics both from Wuhan University, Wuhan, China, in 2016, and currently she is working towards the master degree in electronic engineering department of Tsinghua University, Beijing, China. Her research interests include mobile big data mining, user behavior modeling, data privacy and security.



Fengli Xu received his B.S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2015, and he is currently pursuing Ph.D. degree in electronic engineering department of Tsinghua University, Beijing, China. His research interests include human mobility, mobile big data mining and user behavior modelling.

Yong Li (M'2009-SM'2016) received his B.S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007, and his Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. During July to August in 2012 and 2013, he worked as a Visiting Research Associate in Telekom Innovation Laboratories (T-labs) and HK University of Science and Technology, respectively. During December 2013 to March 2014, he visited University of Miami, FL, USA as a Visiting Scientist. He is currently a faculty member of the Electronic Engineering at the Tsinghua University. His research interests are in the areas of networking and communications. His research is granted by Young Scientist Fund of Natural Science Foundation of China, Postdoctoral Special Find of China, and industry companies of Hitachi, ZET, etc. He has published more than 100 research papers and has 10 granted and pending Chinese and International patents. He has served as Technical Program Committee (TPC) Chair for WWW workshop of Simplex 2013, served as the TPC of several international workshops and conferences. He is also a guest-editor for ACM/Springer Mobile Networks and Applications, Special Issue on Software-Defined and Virtualized Future Wireless Networks. Now, he is the Associate Editor of EURASIP journal on wireless communications and networking.

Pengyu Zhang received the B.S. and M.S. degrees from Tsinghua University in 2007 and 2010, respectively, and the Ph.D. degree from the University of Massachusetts Amherst in 2015. He is currently a Post-Doctoral Researcher with Stanford University. His research interests are embedded systems, sensing, networking, and wireless Communication. He is the winner of the 2016 School of Computer Science Outstanding Dissertation Award from the University of Massachusetts Amherst, the UbiComp 2016 Honorable Mention Award, and the Mobicom 2014 best paper award runner up.

Dan Pei received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree from the University of California at Los Angeles, Los Angeles, CA, USA, in 2005. He is currently an Associate Professor with Tsinghua University. His current research interests are management and improvement of the performance and security of the networked services, through big data analytics with feedback loop. Right now, he is focusing on improving the mobile Internet performance over Wi-Fi networks and data center networks.

Li Su received the B.S. degree in electronics engineering from Nankai University, Tianjin, China, in 1999 and the Ph.D. degree in electronics engineering from Tsinghua University, Beijing, China, in 2007, respectively. He is currently a Research Associate with the Department of Electronic Engineering, Tsinghua University. His research interests include telecommunications, future Internet architecture, and onchip network.

Depeng Jin (M'2009) received his B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Now he is an associate professor at Tsinghua University and vice chair of Department of Electronic Engineering. Dr. Jin was awarded National Scientific and Technological Innovation Prize (Second Class) in 2002. His research fields include telecommunications, high-speed networks, ASIC design and future internet architecture.