Understanding Location Privacy of the Point-of-Interest Aggregate Data via Practical Attacks and Defenses

Wei Tong, *Member, IEEE*, Yinggang Tong, *Graduate Student Member, IEEE*, Chang Xia, Jingyu Hua, *Member, IEEE*, Qun Li, *Fellow, IEEE*, and Sheng Zhong, *Senior Member, IEEE*

Abstract—Location-based services have significantly affected mobile users' everyday life, and location privacy has become essential. Some applications (e.g., location-based recommendation, mobility analytics) do not need the raw location data, and the service providers adopt aggregation to protect users' location traces. However, some works show that even these aggregation data may disclose users' location privacy when additional prior knowledge is available to an adversary. We consider the location privacy problem in the presence of *Location Uniqueness*, a property by which some geographical locations can be re-identified based on the aggregated point-of-interest information. We first study whether existing protection mechanisms are adequate for defending against this type of attack. Then we present two practical attacks for inferring users' actual locations based on the POI aggregates. A secure POI aggregate release mechanism is proposed for defending against this type of re-identification attack and achieving differential privacy at the same time. We conduct extensive experiments on real-world datasets. The results show that the existing protection mechanisms cannot provide sufficient protection against location re-identification attacks. The proposed attacks can significantly improve the inference performance, and the proposed protection mechanism achieves satisfactory performance.

Index Terms—Location privacy, location re-identification, location uniqueness, POI aggregate

1 Introduction

Nowadays, our life has been flooded by Location-based Services (LBSs), and location privacy has also been extensively studied in the past dozen years. Some LBSs do not require users' geographic locations but only leverage the knowledge of Points-of-Interest (POIs) near a user, e.g., recommendation and advertising. These systems only require the aggregation information of the POIs near a user, instead of the geographic locations of these POIs or the user's actual location, which seems privacy-friendly for users' locations in the view of previous location privacy studies that aim to protect users' geographic locations directly.

However, a recent study shows that only providing the types of POIs near a user in a city may also reveal the user's actual location [1]. They propose a notion of *location uniqueness*, which implies that many locations in a city are unique regarding the combinations of POIs around them. Based on the property of *location uniqueness*, they find that users' geographic locations can be inferred based on the nearby POIs regarding the distribution of their types and successfully show that many locations in a city have the property of *location uniqueness*. Their work reveals this vital phenomenon and shows that the property of location uniqueness can significantly affect users' location privacy. Nevertheless, there is still a gap that we need to mind to perform a practical attack based on the property of location uniqueness. Furthermore, protecting location privacy when publishing aggregate POI data in the presence of location uniqueness is also an urgent problem that has not been well studied.

In this paper, we study the practical attacks and defense for location privacy in the presence of location uniqueness. Specifically, we first consider the scenario in which the users may initiate multiple successive LBS requests and extend the concept of location uniqueness to *trajectory uniqueness* in this context. Then, we try to design a practical attack that can significantly improve the precision of the inferred locations compared with the existing re-identification attacks. We want to explore whether we can construct fine-grained attacks on users' locations by exploiting the property of location uniqueness. Our goal is to re-identify users' locations into significantly smaller areas, which allows the attacker to locate the target user practically in the real world. Furthermore, based on the studies of the practical attacks, we also investigate how to protect users' location privacy in the

- *Wei Tong, Yinggang Tong, Chang Xia, Jingyu Hua, and Sheng Zhong are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. E-mail: weitong@outlook.com, tyg@smail.nju.edu.cn, {changxia656569, huajingyu2012}@gmail.com, zhongsheng@nju.edu.cn.*
- *Qun Li is with the Department of Computer Science, College of William & Mary, Williamsburg, VA 23185 USA. E-mail: liqun@cs.wm.edu.*

Manuscript received 5 Oct. 2021; revised 17 Apr. 2022; accepted 13 June 2022. Date of publication 20 June 2022; date of current version 13 May 2023.
This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1005900, in part by NSFC under Grant 62002159, in part by NSFC under Grant 61872176, in part by NSFC under Grant 61972195, in part by the Leading-Edge Technology Program of Jiangsu NSF under Grant BK20202001, and in part by Jiangsu Shuang-Chuang Program under Grant JSSCBS20210035. The work of Qun Li was supported in part by US NSF under Grant CNS-1816399 and the Commonwealth Cyber Initiative.
(Corresponding authors: Sheng Zhong and Jingyu Hua.)
Digital Object Identifier no. 10.1109/TDSC.2022.3184279

presence of location uniqueness without much sacrificing the performance of POI-based services.

We advance the location inference attack from three aspects: 1) we develop a re-identification attack which can infer users' location when they initiate multiple successive LBS requests and find that the success rate of the re-identification can be significantly improved when users continuously use the services by leveraging the knowledge of *trajectory uniqueness*; 2) we propose an iterative positioning scheme for location re-identification, which can significantly shrink the area where the users are in; 3) we also show that the POIs with some certain types have the property of uniqueness as well, and we resort to machine learning methods to learn these POIs even though they have been sanitized in the results for privacy-preserving consideration. This observation also reveals that some straightforward ways, e.g., merely sanitizing the POI frequency list, may not be able to protect the location privacy effectively.

An experimental study has been conducted to investigate whether previous methods like geo-indistinguishability, spatial k-cloaking, and sanitization can successfully protect the location privacy of aggregated POI data in the presence of location uniqueness attacks. The study is performed on the datasets of two representative metropolises: New York City and Beijing. Our results show that these methods can hardly mitigate the re-identification attacks or could be easily broken by more advanced attack techniques.

To protect location privacy in the presence of location uniqueness, we employ the notion of differential privacy and have designed an optimization-based POI type distribution publishing mechanism that can protect the location privacy under differentially private guarantee and significantly defend against the location re-identification attacks. The proposed defense can not only provide protection when releasing single POI type distribution but also can be applied to the cases where the user continuously uses the service, and the adversary may acquire multiple POI aggregates of a user. Using segment clustering and the Gaussian mechanism, we find a way to carefully characterize the correlation of POI frequencies of two locations for a successive use of the LBS service with differential privacy.

The contributions of this paper can be summarized as:

- First, we revisit the concept of location uniqueness and have conducted experimental studies to evaluate the existing protection mechanisms (sanitization, geo-indistinguishability, and spatial k-cloaking) against the existing location re-identification attack. For the sanitization method, we also show that the learning-based model can easily break the protection.
- Second, we present two practical variants of the location re-identification attack. We advance the existing location re-identification attack from two aspects: extending it to a more general case where users may initiate multiple successive LBS requests and significantly improving the success rate of the attacks by leveraging the information of subsequent queries, being able to locate a specific user in a more precise area.
- Third, we have proposed a differentially private defense mechanism for releasing the POI type frequency vectors, which provides a provable privacy guarantee of the location privacy and satisfactory performance in defending against the re-identification attack. Furthermore, for the cases of multiple POI aggregates release, the proposed defense can also defend against the location re-identification attack and achieve a reasonable bound of the accumulated privacy loss even when the user frequently uses the LBS service.
- Fourth, extensive evaluation has been conducted on real-world data traces, which are extracted for the publicly available geo-information service OpenStreetMap [2], T-drive dataset [3], and Foursquare dataset [4]. The results show that the proposed practical attacks provide better attack performance. The results also show that our proposed differentially private mechanism can effectively defend the re-identification attacks with a reasonable cost of utility.

The rest of this paper is organized as follows. The next section presents the preliminaries, and we evaluate the existing defense mechanisms in Section 3. We present our practical variants of the location re-identification attack in Section 4 and evaluate them in Section 5. Our differentially private POI aggregate release mechanism is presented in Section 6. The evaluation of the proposed defense is presented in Section 7. We review the related work in Section 8 and conclude this paper in Section 9.

2 Preliminaries

2.1 System Model

LBS Architecture. We consider a typical LBS architecture in which there are three types of entities: mobile users, the geo-information service provider (GSP), and LBS applications. A mobile user reports its geographic location to the geo-information service provider and gets the geographic information (e.g., POIs, road networks), then it sends the geographic information to the LBS application service providers and enjoys the LBS services. The geo-information service provider stores the geographic data and shares it with the mobile users and LBS applications via a set of query interfaces. The LBS applications provide LBS services and perform various analyses based on the user-location-based geographic data.

Aggregate POI Data. Same as the previous works, e.g., [1], [5], we assume that the LBS applications cannot access mobile users' geographic locations directly. Instead, when a mobile user wants to use the LBS applications, it sends its location to the geo-information service provider and gets the geographic data, and then reports the geographic data aggregates to the LBS applications (e.g., POI-based services). Furthermore, we assume that the geo-information service provider only provides one query interface: retrieving the POIs within a specific range of a location. The LBS architecture adopted in this paper is illustrated in Fig. 1. Note that the POI aggregates may be generated by the users or the GSP and sent to the LBS applications.

Applications. In the existing works that deploy the POI frequency distribution as an ingredient for their studies, most of them generate the POI aggregates based on the users' past activities (e.g., check-in data). For example, Yu

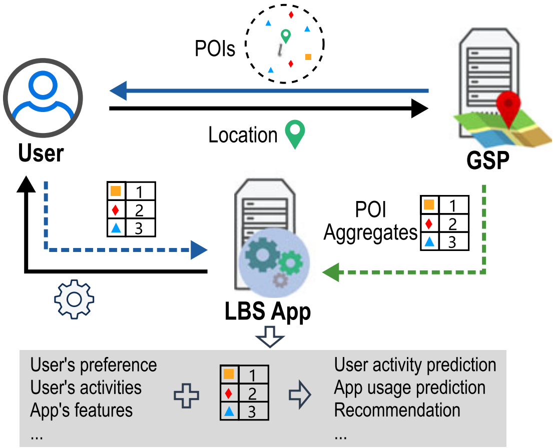


Fig. 1. POI-aggregate-based LBS architecture.

et al. [5] use the POI frequency distribution as a latent feature representation of a location reported by the user and incorporate it with other features, e.g., the app's latent feature, the functionality-based feature, and the preference-based feature, to develop a model for app usage prediction. Similarly, in [6], the (normalized) POI frequency distribution is also generated as a kind of feature based on the check-in data. By combining the POI aggregates and users' online activity data, Fan *et al.* [6] have designed a generative model for online activity prediction. Although most of the existing works use the POI aggregates in an offline manner, they can be generated based on the reported locations instantly or based on users' past activities during a period of time, and both the users and the GSP could store the generated POI aggregates for applications' real-time or future use.

2.2 Problem Formulation

POI Aggregate Data. We formally define the POI aggregate data at first. The user reports its location l and a given query range r to a GSP; then, the GSP generates the set of POIs within the specified query range, denoted by $P_{l,r}$, and returns it to the user. This process can be achieved by the operation:

$$P_{l,r} \leftarrow \mathsf{Query}(\mathbb{G}; l, r), \tag{1}$$

where $\mathbb{G}$ is a geo-information database, which could be a publicly available geospatial database or a private geo-information database managed by the GSP.

In the context of POI type aggregates, the user or GSP does not directly reveal the actual location l or set of POIs $P_{l,r}$ to LBSs or data analysts. Instead, the POI type distribution $F_{l,r} = (n_1, n_2, \ldots, n_M)$ is aggregated by users and released to the POI-based services (e.g., recommendation), where n_i is the frequency of POI type t_i in the result, M is the number of different types of POIs in the city. The POI type distribution can be generated by operation:

$$F_{l,r} \leftarrow \mathsf{Freq}(\mathbb{G}; l, r). \tag{2}$$

Location Re-Identification Problem. As it has been stated in [1], the location re-identification problem is to re-identify the location l based on the distribution $F_{l,r}$. What makes this possible is the property of location uniqueness in the city [1]: given the query range, a location could be re-identified because it has a unique combination of POIs compared to other locations in a city.

Formally, the re-identification process can be formulated as:

$$\Phi \leftarrow \mathsf{Infer}(F_{l,r}, \mathcal{P}), \tag{3}$$

where $\mathcal{P}$ is the prior knowledge of an adversary, $\Phi = \{\phi^1, \phi^2, \ldots, \phi^{|\Phi|}\}$ is a set of re-identified areas in which the location l could be. An adversary that wants to re-identify a target's location precisely, i.e.,

$$Minimize \sum_{i}^{|\Phi|} \mathsf{size}(\phi^i), \tag{4}$$

$$subject\ to\ F_{l,r}, \mathcal{P}. \tag{5}$$

The above straightforward definition of the problem may lead to a result in that the number of areas is large, and they are very far away from each other, although the sum of area sizes is small. In most cases, that result makes it impractical for the attacker to obtain the target's other information (e.g., school, home address, office) from the re-identified location. Then we define the location re-identification problem with two stages. The first stage is the same as the definition in [1], i.e., identifying a continuous area in which the target is:

$$Maximize\ \Pr[|\Phi| = 1] \tag{6}$$

$$subject\ to\ F_{l,r}, \mathcal{P}. \tag{7}$$

Let ϕ^* be the area identified in the first stage, and the second stage is to locate the target precisely in ϕ^*:

$$Minimize\ \mathsf{size}(\phi^*) \tag{8}$$

$$subject\ to\ F_{l,r}, \mathcal{P}. \tag{9}$$

2.3 Threat Model

We assume that the adversary is semi-honest, which means it is interested in inferring users' locations based on the informed information but does not deviate from the protocol specification.

Abilities. The adversary could be a third-party entity that uses the POI type aggregates involved in the POI-based services. We assume that the adversary can access a set of prior knowledge $\mathcal{P}$, which includes public geo-information of a city and the operation Freq to get POI type frequency of any location with the required query range. Such prior knowledge can be obtained from some publicly available geo-information service providers, e.g., OpenStreetMap [2]. Besides, same to [1], we also assume that the adversary can obtain: 1) the user's identification corresponding to the reported $F_{l,r}$, which makes it possible for the adversary to link the re-identified location to a particular user; 2) user's query range r. These two types of information are essential information for any location-based services and are usually included in the meta-data with queries.

Goals. The adversary tries to re-identify a user's location based on $F_{l,r}$ and the prior knowledge by implementing an inference mentioned in (3). Ideally, $|\Phi|$ should be 1, and the size of the only element ϕ^* in the set should be as small as possible. Particularly, same to [1], we define the case where $|\Phi| = 1$ as a successful attack, and $|\Phi| \neq 1$ means that the attack fails. Therefore, we adopt two metrics to evaluate the inference method Infer: 1) success rate of attacks, which equals the ratio between the number of successful attacks to the number of all attacks; 2) when an attack is successful, the area of ϕ^* is used to measure the precision of the inference.

2.4 Region Re-Identification.

For the sake of completeness, we review the location re-identification method in [1] in this part. Specifically, the attack runs by the following steps:

1) Counting the overall POI frequency in the entire city, denoted by F;
2) Sorting $F_{l,r}$ by F, and denoted by t_l the most infrequent POI type in F which satisfies $n_l > 0$;
3) Finding all POIs with type t_l in the city, and denoted by P_{t_l} the resulted set of POIs;
4) Pruning the set of locations P_{t_l} with following rule:
   - For each $p_{t_l} \in P_{t_l}$, get
   $$F_{p_{t_l},2r} \leftarrow \mathsf{Freq}(\mathbb{G}; p_{t_l}, 2r);$$
   - For $i = 1, 2, \ldots, M$, if exist any i such that $F_{p_{t_l},2r}[i] < F_{l,r}[i]$, remove p_{t_l} from the candidate set P_{t_l}.
5) After the above pruning process, if there is only remaining one location $p^*_{t_l}$ in the set P_{t_l}, the location l has the property of uniqueness. The adversary can infer that location l is in the range of $p^*_{t_l}$ with radius r.

Their method is based on the property that the circle that is centered at l with radius r is completely covered by the circle centered at p_{t_l} with radius $2r$ if p_{t_l} is a POI in the distance r of location l. By using this method, the adversary can re-identify a location by POI type distribution with no false negative, but the success rate is affected due to the gap between $F_{p_{t_l},2r}$ and $F_{l,r}$. Also, the adversary can only determine that location l is in the range with distance r of $p^*_{t_l}$, which means the size of ϕ^* is πr^2, which seems to be an infeasible range for attacks on location privacy. For convenience, we refer to this attack as *region re-identification* or *Cao et al.'s attack* in the following parts of this paper.

2.5 Privacy Model

Differential privacy (DP) has become a very important standard for data privacy protection in recent years. For the sake of completeness, we first review the definition of DP [7] below.

Definition 1. *A randomized mechanism $\mathcal{M} : \mathcal{X}^d \rightarrow \mathcal{Y}$ is (ϵ, δ)-differentially private if for any two neighboring datasets $\mathbb{D}_1, \mathbb{D}_2 \in \mathcal{X}^d$, and all $\mathcal{S} \subset \mathcal{Y}$,*

$$\Pr[\mathcal{M}(\mathbb{D}_1) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(\mathbb{D}_2) \in \mathcal{S}] + \delta, \tag{10}$$

where ϵ and δ are privacy parameters.

Then, we review the Gaussian mechanism, which we will adopt as a component in our private defense mechanism.

Definition 2. *Gaussian mechanism adds a noise $\mathcal{N}(0, \sigma^2)$ to $f(\mathbb{D})$, where f is a function with sensitivity Δ. If*

$$\sigma \geq \sqrt{2\ln(1.25/\delta)}\Delta/\epsilon, \tag{11}$$

then, the mechanism achieves (ϵ, δ)-differential privacy.

Lemma 3 (Post-processing [7]). *Let $\mathcal{M} : \mathcal{X}^d \rightarrow \mathcal{Y}$ be a randomized mechanism satisfying (ϵ, δ)-differential privacy. Let $\mathcal{A} : \mathcal{Y} \rightarrow \mathcal{Y}'$ be an arbitrary deterministic or randomized mechanism. If $\mathcal{M}' : \mathcal{X}^d \rightarrow \mathcal{Y}'$ is sequential apply of $\mathcal{M}$ and $\mathcal{A}$, then $\mathcal{M}'$ is (ϵ, δ)-differentially private.*

Theorem 4 (Sequential composition). *Let $\mathcal{M} : \mathcal{X}^d \rightarrow \mathcal{Y}$ be a sequential application of k randomized mechanisms that satisfy (ϵ, δ)-differential privacy. Then $\mathcal{M}$ achieves $(k\epsilon, k\delta)$-differential privacy.*

2.6 POI Datasets

The POI datasets are extracted from a publicly available geo-information service, OpenStreetMap [2]. We choose New York and Beijing as the targets of our analysis. Beijing dataset contains 10,249 POIs with 177 different types; New York City (NYC) dataset contains 30,056 POIs with 272 different types.

3 Evaluating the Existing Protection Methods Against Location Re-Identification

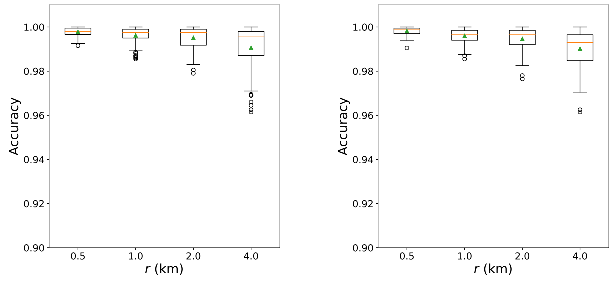
We now measure the region re-identification attack against three protection mechanisms: sanitization and geo-indistinguishability, and spatial k-cloaking.

3.1 Sanitization

A straightforward solution that seems can be applied to protect location privacy in the presence of location uniqueness is to sanitize the frequencies of the POI types, especially for those infrequent POI types. Below we describe a sanitization strategy for the protection, which removes the information of frequencies of POI types that are infrequent in the city. Our results show that the aggressive sanitization strategy can significantly reduce the success rate of the region re-identification attack in some cases. However, we note that the effectiveness of the sanitization relies on two major assumptions: 1) the frequencies of POI types are mutually independent, i.e., the location of any POI in the city is not correlated with the locations of other POIs; 2) the adversary cannot distinguish whether a POI frequency distribution is generated based on a real location in the city or arbitrarily synthesized by the defense. We can find these two assumptions are too strong in practice, and we have demonstrated that the defense can be easily compromised if the attacker has some prior knowledge by presenting a learning-based inference method.

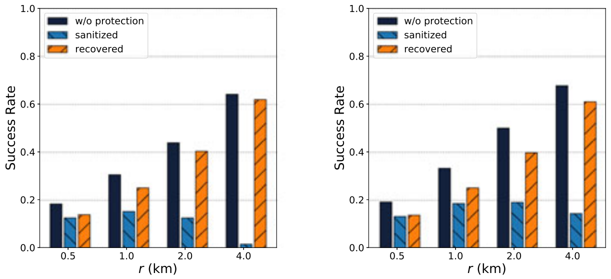
Defense Strategy. Based on the overall POI frequency in the entire city, F, the sanitizer chooses a set of POI types T^S which satisfies that any POI type $t_i \in T^S$, $F[i] <= S$. When trying to report the POI type distribution $F_{l,r}$ of the location l in range r, the user sets $F_{l,r}[i] = 0$ if $t_i \in T^S$.

Prediction Against Sanitization. We assume that the adversary knows two types of prior knowledge:



(a) Beijing (b) New York City

Fig. 2. The accuracy of prediction models.



(a) Beijing (b) New York City

Fig. 3. Performance of the sanitization.

- Whether a specific POI type is sanitized or not sanitized, for example, the attacker may collect the historically reported frequencies for inferring such information.
- The attacker has a collection of POI frequency distributions of a bunch of locations. This can be easily achieved by using open-source geo-information databases.

For each sanitized POI type t^S, we train a prediction model based on the reported frequencies of POI types. Formally, the prediction model is formulated as

$$\mathsf{Pred}(\mathbf{x}^{-S}) \rightarrow n^S.$$

where $\mathbf{x}^{-S} = (n_1 n_2 \dots n_{|T^{-S}|})$ is the feature vector of prediction sample, in which n_i is the corresponding frequency of POI type t_i. We should clarify that t_i is a type in the set T^{-S}, which is the set of POI types that are not sanitized; n^S is the target of the prediction model, which is the frequency of sanitized POI type t^S.

We adopt the support vector machine (SVM) classification [8] with radial basis function (RBF) kernel as an installation of the prediction model. Our experiments are implemented by using Scikit-learn machine learning package [9]. In the training process, random locations are generated in the corresponding city ,[1] and by adopting Freq operation, the POI type distributions are generated from these locations. We compose a training dataset with 10,000 samples and a validation dataset with 2,000 samples for training based on the generated POI type distributions. All samples in the prediction model are normalized by being centered to mean and scaled with unit standard deviation.

Results. In our experiments, we adopt a very aggressive sanitization behavior when implementing the strategy. Specifically, we set the sanitization threshold $S = 10$ (i.e., the POI types whose frequencies are no more than 10 will not be reported), which results in 138 (90, resp.) POI types removed from the POI frequency distribution in New York City (Beijing, resp.).

Fig. 2 shows the classifiers' performance for different query range (r). In this set of experiments, we evaluate the defense strategy with user locations that are randomly generated in corresponding cities. We can observe that for both Beijing and New York City, in the cases of typical query ranges of 0.5 km, 1.0 km, 2.0 km, and 4.0 km, the average validation accuracy of trained classifiers for all targets is larger than 95%. Specifically, for the Beijing, the means of accuracies are 0.998 (± 0.002), 0.996 (± 0.004), 0.995 (0.005), and 0.991 (± 0.010) for the above four query ranges, respectively. For New York City, the means of accuracies are 0.998 (± 0.002), 0.996 (± 0.003), 0.995 (0.005), and 0.990 (± 0.008) for the above four query ranges, respectively.

Fig. 3 shows that the sanitization can mitigate a major part of the attacks, and reduces the success rate from 0.184, 0.306, 0.440, and 0.642 to 0.126, 0.153, 0.126, and 0.016, respectively. For New York City, the success rates decrease from 0.192, 0.333, 0.501, and 0.678 to less than 0.2 for four cases, respectively, when the defense is applied. However, we observe that the prediction models can recover the sanitized types, and achieve an almost success rate compared with the original attacks without protection.

3.2 Geo-Indistinguishability

Geo-indistinguishability [10] is a variant of differential privacy, which provides provable guarantees of location privacy. Its main idea is to bound the difference between distributions of observations that are produced by two close locations by probabilistic perturbation. Formally, a mechanism M is geo-indistinguishable if and only if for any l, l' which satisfy $\mathsf{dist}(l, l') \le R$:

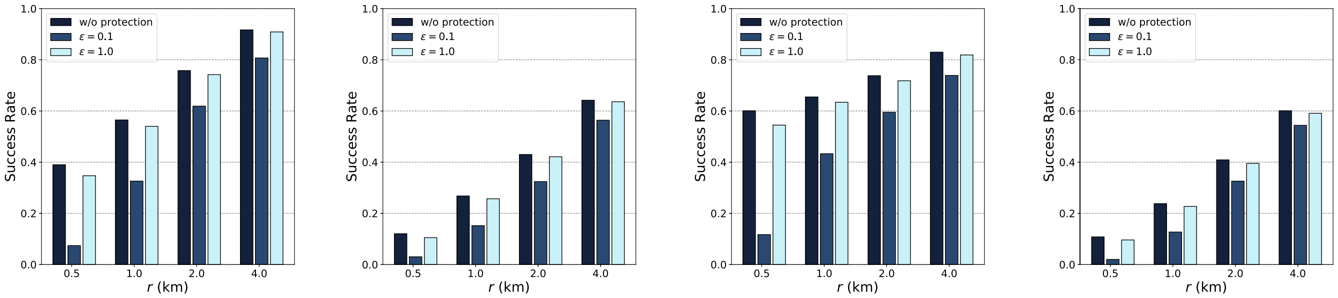
$$|\ln \frac{M(l)}{M(l')}| \le \epsilon R. \tag{12}$$

Planar Laplacian [10] is a canonical way to achieve geo-indistinguishability, which runs in the following way: given user's location l and the privacy parameter ϵ, for any other location l' in the considered area, the mechanism chooses l' as the reported location by the following probability:

$$M_\epsilon(l)(l') = \frac{\epsilon^2}{2\pi} \exp^{\epsilon \times \mathsf{dist}(l,l')}. \tag{13}$$

Results. In this set of experiments, we evaluate the defense strategy with four datasets:(a) T-drive [3] user locations in Beijing; (b) randomly generated user locations in Beijing; (c) Foursquare check-ins [4] in NYC; (d) randomly generated user locations in NYC. In our experiments, we also note that the unit of distance is set to 100 meters, which will affect the privacy level and the utility of the perturbation given the specific privacy parameter. For each dataset, 1000 locations are randomly selected for the experiments. We set the unit distance as 100 meters for geo-indistinguishability. We note that privacy parameter ϵ and the unit

[1] We note that although the locations are randomly chosen, the POI frequency distributions are generated based on the real geo-information instead of being artificially created in a random way.



(a) Bejing: T-drive (b) Bejing: Random (c) NYC: Foursquare (d) NYC: Random

Fig. 4. Performance of Planar Laplacian.

distance together determine the degree of privacy protection. For instance, when the unit distance is changed from 100 meters to 10 meters, if we want to achieve the same scale of perturbation, the privacy parameter should be $\epsilon/10$.

Fig. 4 shows the performance of *Planar Laplacian* for defending against location re-identification. We can find that when the privacy budget is larger (e.g., $\epsilon = 1.0$), the mechanism can barely mitigate the inference attack. When we set $\epsilon = 0.1$ and $r = 0.5$, 1.0, 2.0, and 4.0, respectively, the *Planar Laplacian* can mitigate about 81.01%, 42.30%, 18.34%, and 12.00% of attacks for T-drive dataset in Beijing, about 75.00%, 43.28%, 24.65%, and 12.15% for random locations in Beijing, about 80.53%, 33.89%, 19.38%, and 10.96% for Foursquare dataset in NYC, and about 81.48%, 46.64%, 20.29%, and 9.48% for random locations in NYC. The defense can mitigate most of the attacks when the query range is small, but the performance is limited when the query range is large.

Our results are closely relevant to the findings in [11] that geo-indistinguishability may not be an ideal notion for location privacy in terms of privacy-utility trade-off. We borrow two utility metrics from [11]: average loss: the euclidean distance between the actual location and reported location; r_{95} the radius of the area centered with the actual location where a reported location is in it with the probability of 95%. For $\epsilon = 0.1$ or $\epsilon = 1.0$ with the unit of distance as 100 m, the average loss is 2000 m or 200 m, respectively; and r_{95} is 4744 m or 474 m, respectively. When ϵ is large, the obfuscation of the actual location is limited. Thus if the query range is comparable large, the POI frequency distribution generated on two locations may be very similar, which makes the notion of geo-indistinguishability not necessarily a good choice for protecting the location privacy in this case.

3.3 Spatial k-Cloaking

Spatial k-cloaking is a type of location privacy protection mechanism, which aims to hide a location in a larger area containing the requester and at least k-1 other users. In our evaluation, we have adopted the adaptive-interval cloaking algorithm [12] as the protection scheme. For the sake of completeness, we briefly review the adaptive-interval cloaking algorithm below:

1) The algorithm first sets the whole city area as the initial current area for cloaking.
2) It partitions the current area into four non-overlapping sub-regions with equal size and tests whether the sub-region which contains the targeted location satisfies the k-anonymous property, i.e., there are at least k users in this sub-region.
3) If the sub-region satisfies the k-anonymous property, it repeats 2) and 3); otherwise, it chooses the generated region in the last iteration as the cloaking area.

Results. In this set of experiments, we evaluate the defense strategy with four datasets that are the same as we have adopted in the § 3.2. We assume that there are 10,000 users who are uniformly distributed all over the city for each city. Fig. 5 shows the performance of spatial k-cloaking for defending against location re-identification. We can find that the success rate decreases with k increasing, but its performance is still not satisfactory when k is sufficiently large (e.g., $k = 50$).

3.4 Takeaways

We have the following three major observations from the above experiments study:

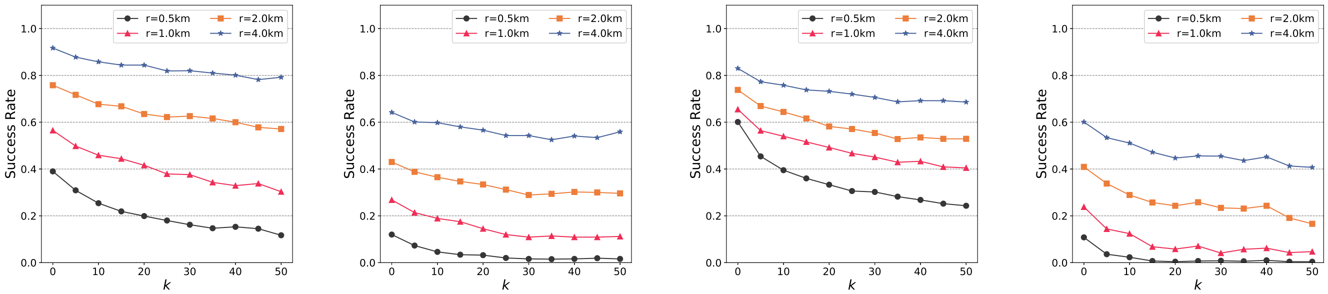
1. Location-level protection (e.g., Geo-indistinguishability, Spatial k-cloaking) achieves better performance when the query range is small. From Figs. 4 and 5, we can find that when the query range is small, both the Geo-indistinguishability and the Spatial k-cloaking can reduce the success rate of the attacks more significantly compared with the cases where the query ranges are larger. Intuitively, the area of retrieval increases squarely when the query range increases, and then the POI frequency distribution may not change much after the actual location being obfuscated. Formally, for the *Planar Laplacian* with a given ϵ and the spatial k-cloaking with a given k, we assume that the perturbation makes the actual location l deviate d km, and the POIs are uniformly distributed in the city. Without loss of generality, we assume $d < r$. Then the change of the area of retrieval is

$$2\pi r^2 - 4\arccos\left(\frac{d}{2r}\right)r^2 + 2d\sqrt{r^2 - \frac{d^2}{4}}.$$

The ratio of the change to the area of retrieval is

$$\begin{aligned} f(r) &= \frac{2\pi r^2 - 4\arccos\left(\frac{d}{2r}\right)r^2 + 2d\sqrt{r^2 - \frac{d^2}{4}}}{\pi r^2} \\ &= 2 - \frac{4\arccos\left(\frac{d}{2r}\right)}{\pi} + \frac{2d\sqrt{r^2 - \frac{d^2}{4}}}{\pi r^2} \end{aligned}$$

which monotonically decreases when $r > d$. For instance, let $d = 1$, when $r = 2$ and $r = 4$, the ratio is about 0.63 and 0.32, respectively. For $r \leq d < 2r$, we have a similar result.



(a) Bejing: T-drive (b) Bejing: Random (c) NYC: Foursquare (d) NYC: Random

Fig. 5. Performance of spatial k-cloaking.

2. We can observe that the frequency-level protection (e.g., sanitization without recovery) performs better compared with geo-indistinguishability or spatial k-cloaking. One of the reasons is that the sanitization makes two strong assumptions: the frequencies of POI types are mutually independent, and the adversary cannot distinguish whether a POI frequency distribution is generated based on a real location in the city. Moreover, for the frequency-level protection, the modification is directly applied to the POI frequency distribution, while for the location-level protection, the obfuscation is applied to the user's actual location and then affects its corresponding POI frequency distribution indirectly, which produces a weaker association between the location-level protection and the perturbed POI frequency distribution. We also observe that sanitization can provide better protection when the query range is large. From Fig. 3, we can find that the sanitization can significantly reduce the success rate when the query range is large. However, the sanitization also cannot provide sufficient protection when the query range is low or more powerful attacks exist. One of the reasons could be that more POIs will be in the area of retrieval when the query range is larger. Then the POI frequency distribution has a higher probability of containing the types that will be sanitized.

3. The attack could be more powerful with the user traces in real-world applications. From these three sets of experimental studies, we can observe that the re-identification attack can achieve higher success rates for the real-world data traces.

4 Understanding the Location Uniqueness Via Practical Attacks

An important direction for location privacy research has been pointed out by Cao *et al.*'s work [1] by introducing the concept of *location uniqueness*. They also provide a feasible method for re-identifying regions that may contain the target user based on POI type distribution. However, as we have mentioned above, their approach is mainly used for exploring the existence of location uniqueness, and an adversary who is interested in users' location privacy may need more powerful tools for launching the attacks.

For the practical attacks, we identify two major goals: 1) the re-identified location should be more precise, which means the adversary can determine the user's location in a sufficiently small area; 2) the success rate should be further improved, which means the adversary has a high probability to determine the user's location in only one area successfully. In this section, we introduce two practical variants of the region re-identification based on POI type distribution to pursue the above two goals, respectively.

4.1 Fine-Grained Attack

After applying the Cao *et al.*'s attack, an adversary can re-identify those locations with the property of location uniqueness and narrow each successfully re-identified location in a circle with radius r, but cannot determine where the locations exactly are. We find that the basic re-identification method uses the relationship between $F_{l,r}$ (i.e., POI type distribution around actual location l with radius r) and $F_{p_{t_l},2r}$ (i.e., POI type distribution around found POI p_{t_l} with radius $2r$) to prune the candidate set of re-identified POIs. Their method only uses the POI type distribution information of POIs with the most infrequent type. Nevertheless, we find that other POIs with other types can also be exploited to locate the user.

The basic idea of the proposed fine-grained inference method is to shrinkage the area the user is in by iteratively applying the candidate pruning strategy for other types of POIs, and find POIs in $P_{p^*_{t_l},2r}$ that are also in $P_{l,r}$. After finding these POIs, the adversary can further locate l because l is definitely within r of the selected POIs. Specifically, we present the following scheme to find a significantly smaller area l should be in, which consists of three steps:

- The first step is to re-identify the location l by Cao *et al.*'s region re-identification method, which can infer that location l is in the range of $p^*_{t_l}$ with radius r. We refer to the found POI $p^*_{t_l}$ as the major anchor for the location inference.
- Once the anchor POI $p^*_{t_l}$ is found, we can further improve the accuracy of the re-identification of location l by leveraging other types of POIs in the surrounding area. Though the queried POIs $P_{l,r}$ based on location l are unknown to the attacker, it can obtain the set of POIs $P_{p^*_{t_l},2r}$, which is a superset of $P_{l,r}$. Based on this knowledge, the attacker can carefully filter the points in $P_{p^*_{t_l},2r}$ and find some auxiliary anchors to position the location l. An algorithm to find these auxiliary anchors is presented in Algorithm 1.
- After generating the set of auxiliary anchors, which are all in the range of r of the location l, and thus the location l can be positioned into a very fine-grained area by computing the feasible area that satisfies the requirements of these anchors.

Algorithm 1. Iteratively Shrink the Region.

```
Input: $F_{l,r}$: the frequency vector of POI types;
       $t_l$: most infrequent POI type;
       $p^*_{t_l}$: corresponding POI for re-identifying $l$;
       $\max_{aux}$: maximum size of set of anchors.
Output: $Aux$: set of POI for positioning $l$
1  $Aux \leftarrow \emptyset$;
2  $P_{p^*_{t_l},2r} \leftarrow$ Query($\mathbb{G}; p^*_{t_l}, 2r$);
3  $F_{p^*_{t_l},2r} \leftarrow$ Freq($\mathbb{G}; p^*_{t_l}, 2r$);
4  $F_{\text{diff}} \leftarrow F_{p^*_{t_l},2r} - F_{l,r}$;
5  Sort $F_{\text{diff}}$ based on the frequencies of POI types.
6  foreach $t_i \in F_{\text{diff}}$ do
7    if $F_{\text{diff}}[t_i] = 0$ then
8      $Aux \leftarrow Aux \cup \{p \in P_{p^*_{t_l},2r} | p.\text{type} = t_i\}$
9    else
10     foreach $p \in P_{p^*_{t_l},2r}$ and $p.\text{type} = t_i$ do
11       $flag \leftarrow True$;
12       $F_{p,2r} \leftarrow \text{Freq}(p, 2r)$;
13       foreach $t_p, v_p \in F_{p,2r}$ do
14         if $v_p < F_{l,r}[t_p]$ then
15           $flag \leftarrow False$;
16       if $flag$ then
17         $Aux \leftarrow Aux \cup \{p\}$
18   if $|Aux| >= \max_{aux}$ then
19     break;
```

In Algorithm 1, we first compute the difference between POI type distributions of the actual location and major anchor and get a differential vector F_{diff}. The algorithm traverses all types of POI based on the sorted type in F_{diff}. This operation is adopted to speed up the iterative shrinkage process because the algorithm can first consider the types that need fewer efforts to prune the POIs. For example, if for a type t_i such that $F_{\text{diff}}[i] = 0$, then all POIs with type t_i in $P_{p^*_{t_l},2r}$ are in $P_{l,r}$. Therefore, we can directly use these POIs with type t_i to shrink the target area without additional effort.

Example. Fig. 6 shows an example of the fine-grained attack.

a) For a location l, its nearby aggregate POI distribution with range r is $F_{l,r} = \langle \text{school}: 1, \text{gym}: 2, \text{restaurant}: 3 \rangle$. The POI found by Cao *et al.*'s method is a school with location $p^*_{t_l}$, and $F_{p^*_{t_l},2r} = \langle \text{school}: 1, \text{hospital}: 1, \text{gym}: 2, \text{restaurant}: 5 \rangle$. For now, the attacker knows that the victim user is in a circular region within radius r around $p^*_{t_l}$.

b) Our iterative region shrinking algorithm then computes $F_{\text{diff}} = \langle \text{school}: 0, \text{hospital}: 1, \text{gym}: 0, \text{restaurant}: 2 \rangle$. For types school and gym, because all the POIs with these two types are identical in both $P_{l,t}$ and $P_{p^*_{t_l},2r}$, these POIs can be used as anchors.

b) Because the type of hospital does not appear in $F_{l,r}$, the hospitals will not be used as anchors. Due to $F_{\text{diff}}[\text{restaurant}] > 0$, the algorithm verifies whether a POI with this type satisfies the requirement of an anchor for positioning l, and a POI is chosen as an anchor in our example.

d) We can observe that the search area (the area bounded by the green lines) of the victim user (location l) is significantly reduced after applying the iterative region shrinking algorithm.

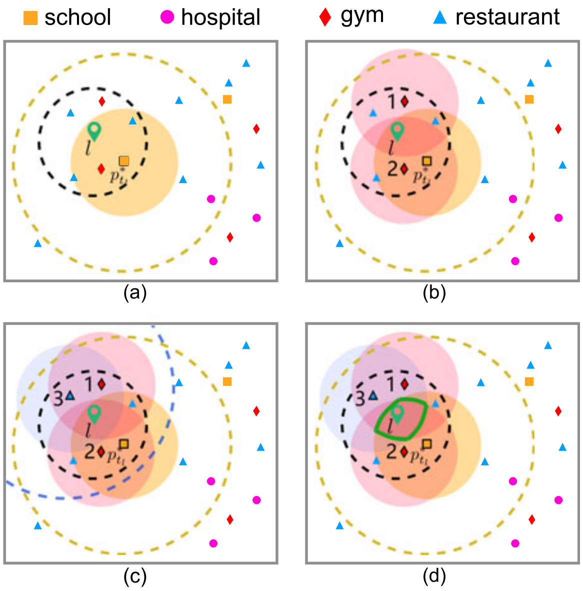


Fig. 6. Example of the fine-grained attack. The overall POI frequencies in the entire city $F = \langle \text{school}: 2, \text{hospital}: 3, \text{gym}: 4, \text{restaurant}: 9 \rangle$.

Computational Complexity. Recall that the number of POI types in the city is M, and we assume for any POI type, the number of POIs with this type is no more than N. In the worst case, the computational complexity of Cao *et al.*'s attack is $O(N \times M)$. Then we analyze the extra computation required for the fine-grained attack. The computation of $P_{p^*_{t_l},2r}$ and $F_{p^*_{t_l},2r}$ has been done by the Cao *et al.*'s attack. The complexity of computing F_{diff} is bounded by the number of POI types M. In the worst case, the innermost loop of the Algorithm 1 (Line 11 - 17) needs to execute $\min(|P_{p^*_{t_l},2r}|, \max_{aux})$ times. In most cases, we assume $\max_{aux} < |P_{p^*_{t_l},2r}|$, and we note that the frequency computation in Line 12 can be implemented by querying pre-computed indexes because any $p \in P_{p^*_{t_l},2r}$ is a POI, whose aggregate POI distribution can be computed before the attacks. Finally, we have the computational complexity of the Algorithm 1 is $O(\max_{aux} \times M)$, and overall, it is $O(M \times (\max_{aux} + N))$ for the fine-grained attack.

4.2 Attack With Trajectory Uniqueness

When users are using location-based services, they often query the service multiple times. Several successive queries may further reveal users' location based on the inference on the aggregated POI frequencies. We call this property trajectory uniqueness and demonstrate that it can be leveraged for location re-identification with a better success rate.

For the cases that the adversary can leverage multiple releases of POI type frequencies, the location re-identification problem is extended to the following form:

$$\{\Phi_1, \Phi_2, \ldots\} \leftarrow \text{Infer}(\{F_{l_1,r}, F_{l_2,r}, \ldots\}, \mathcal{P}). \tag{14}$$

By repeatedly applying the single location version of the re-identification attack, the adversary can get a series of inference candidates: $\{\hat{\Phi}_1, \hat{\Phi}_2, \ldots\}$. Our goal is to figure out which subset contains the areas that the user is possible in for a given candidate set $\hat{\Phi}_t$. An ideal case is that the adversary has the knowledge about the distance between two locations, i.e., $\text{dist}(\phi^*_t, \phi^*_{t+1})$. Thus, the adversary can filter the

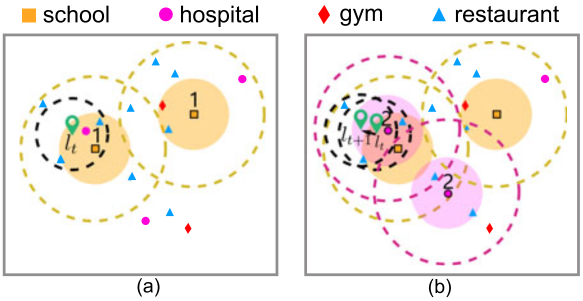


Fig. 7. Example of the trajectory uniqueness attack. The overall POI frequencies $F = \langle \text{school} : 2, \text{gym} : 2, \text{hospital} : 3, \text{restaurant} : 9 \rangle$.

pair of candidate areas in the candidate sets and find the possible pair of locations based on their distance. However, this assumption seems unrealistic in most cases, and we need a practical way to estimate the distance between two successive locations.

We consider the distance estimation as a regression problem. We find that the crucial part of this extended re-identification problem is that the prior knowledge $\mathcal{P}$ is also extended. The adversary also captures the duration between two successive releases. Therefore, we try to build a regression model mainly based on the duration and other auxiliary information to predict the distance between two locations of corresponding releases. Specifically, we construct the feature vector with the following information:

- The duration between two successive releases: time (l_t, l_{t+1});
- The $L1$-distance between $F_{l_t,r}$ and $F_{l_{t+1},r}$;
- In which hour of a day the first POI type frequency is released, and which day of a week for this release. These two types of information are encoded by one hot encoding in the feature vector.

Based on the constructed feature vectors, we adopt support vector regression [13] that is provided Scikit-learn machine learning package [9] to train the regressor.

Example. An example of the trajectory uniqueness attack is presented in Fig. 7.

a) For a location l_t, its nearby POI frequency distribution with range r is $F_{l_t,r} = \langle \text{school} : 1, \text{hospital} : 1, \text{restaurant} : 2 \rangle$. The POIs found by Cao *et al.*'s method are two schools (marked with "1" in Fig. 7a). For now, the attacker cannot decide which circular regions (centered with a school with the radius r) the victim user is in.

b) After a short period, the user moves from l_t to l_{t+1}. The corresponding POI frequency distribution is $F_{l_{t+1},r} = \langle \text{hospital} : 1, \text{restaurant} : 3 \rangle$. We apply the re-identification method again and find that the victim user may be within radius r around two POIs with type *hospital* (marked with "2" in Fig. 7b). By taking $F_{l_t,r}$ and $F_{l_{t+1},r}$ into consideration, we can infer that l_t is closer to the school on the left in the figure because we know the user cannot move very far in a short period of time. For a more complicated scenario, the attacker can use the aforementioned regression to infer the distance a user has moved.

Computational Complexity. As we have analyzed in Section 4.1, the computational complexity of Cao *et al.*'s is $O(NM)$. We assume that the user has T queries, and the set of inference candidates produced by the single location re-identification attack is $\{\hat{\Phi}_1, \hat{\Phi}_2, \ldots, \hat{\Phi}_T\}$. The time complexity of generating the candidates is $O(TNM)$. The time complexity of constructing the feature vector is $O(M)$ for any two successive queries. The inference time of the support vector regression is negligible compared to the above two parts because it is linear to the input dimension, which is small in our method. Thus, the computational complexity of the trajectory uniqueness attack is $O(TNM + (T-1)M)$.

5 Evaluating the Attacks

We have implemented the proposed attacks and evaluated them based on real-world user data traces. Our results show that the proposed attack needs less than 25% of the search area compared with the existing attack in most cases. The attack leveraging trajectory uniqueness can increase the attack's success rate up to about 20% when $r = 0.5$.

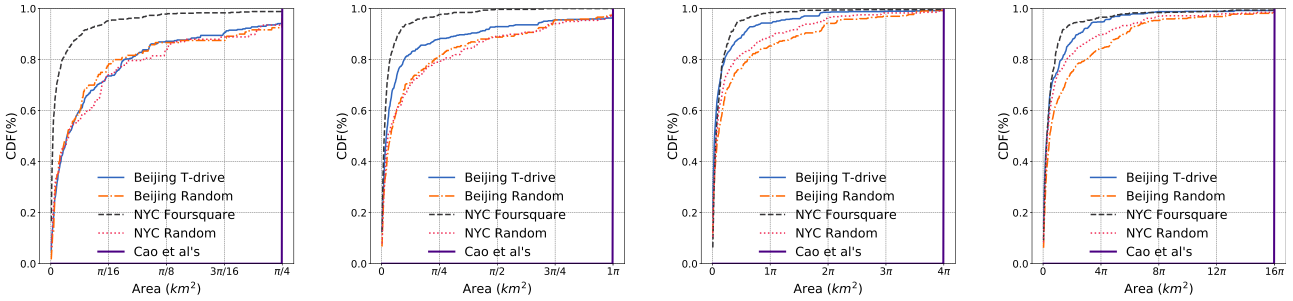
5.1 Settings

The evaluation of fine-grained attack is conducted on four datasets:(a) T-drive [3] user locations in Beijing, which contains trajectory data of 10,357 taxis in Beijing. We extract the trajectories which are within the given area of the city. (b) randomly generated user locations in Beijing; (c) Foursquare check-ins [4] in New York City, which contains 227,428 check-ins from 824 users; (d) randomly generated user locations in New York City. The evaluation of trajectory uniqueness is carried out on trajectories that are extracted from T-drive dataset. For each set of experiments, we randomly choose 1,000 locations or segments from the datasets for evaluation.

5.2 Search Area Reduction

Fig. 8 shows the performance of fine-grained attack that we have proposed in Section 4.1 when $\max_{aux} = 20$. We can find that this attack dramatically reduces the area size that needs to search for the user's actual location. In about 80% cases, the proposed attack can reduce the search area to no more than a quarter of the search area required by Cao *et al.* 's attack. Furthermore, we can find that with the query range increasing, the fine-grained attack performs better on the search area reduction.

In Fig. 9, we can see that, with the number of auxiliary anchors increasing, the attack achieves better performance for all the four datasets. On average, for these four datasets, the fine-grained attack can reduce the size of the search area from 1.70km^2 to 0.60km^2, 2.38km^2 to 1.35km^2, 1.92km^2 to 0.26km^2, and 2.63km^2 to 1.07km^2, respectively, when the number of auxiliary anchors increases from 5 to 40. We can also find that the reduction brought by more auxiliary anchors decreases with the number of auxiliary anchors increasing. Therefore, it may not be the best choice to use all the auxiliary anchors because the computation time will increase when more auxiliary anchors involve. In our experiments, let $\max_{aux} = 20$ could be a reasonable choice. We note that the search area of Cao *et al.*'s attack is always about 16.56km^2 when $r = 2\text{km}$.



(a) $r = 0.5$km (b) $r = 1.0$km (c) $r = 2.0$km (d) $r = 4.0$km

Fig. 8. Performance of the fine-grained attack: the CDF of search area. The indigo lines in the figures show the performance of Cao *et al.*'s attack in terms of search area. The search area of Cao *et al.*'s attack is always $\pi r^2 \text{km}^2$ ($\pi/4$, π, 4π, and 16π, respectively) for 100% of locations in any datasets.

5.3 Performance of the Trajectory Uniqueness Attack

Fig. 10 shows the performance of the re-identification attack when two successive releases are leveraged in Beijing with T-drive datasets. In our experiments, we extract the points in the trajectories satisfying the requirements: 1) the released POI type frequencies are changed because the adversary can be aware that if the POI type frequency is the same as the previous release, this release is useless; 2) the duration of two successive releases is less than 10 minutes, because when the duration is large, the user may start another new session of using the location-based services. We can observe from the results that the success rate is improved by using the knowledge of two successive queries. For $r = 0.5$km, 1.0km, 2.0km, and 4.0km, the enhanced attack has 0.203, 0.146, 0.09, 0.001 gains on success rate, respectively. We can find that the gain is minimal when $r = 4.0$km because the performance of location re-identification is good enough with a large query range.

5.4 Attack Performance Against Existing Defenses

In this set of experiments, we evaluate the proposed attacks against the existing defenses. We set $r = 2.0$km, and evaluate both the success rate and the size of search area of the fine-grained attack.

Fig. 11 shows the performance of the proposed fine-grained attack against the *Planar Laplacian* defense mechanism. We can find that when the privacy budget is larger (i.e., $\epsilon = 1.0$), the Planar Laplacian can hardly affect the performance of the attack on both the success rate and the size of the search area. In Fig. 11a, we can find that when $\epsilon = 0.1$, the Planar Laplacian can reduce about 18.02%, 23.72%, 20.71%, and 23.26% for Foursquare dataset in NYC, random locations in NYC, T-drive dataset in Beijing, and random locations in Beijing, respectively, on the success rate. However, we note that the search area reduction is the major feature of the fine-grained attack, and we can observe in Fig. 11b that the change of the size of the search area is not significant, even though a strong defense is applied (i.e., $\epsilon = 0.1$).

Fig. 12 shows the performance of the proposed attack when the spatial k-cloaking defense mechanism is applied. We change the cloaking parameter k from 10 to 50. In Fig. 12a, we can observe that with k increasing, the success rate decreases. Compared with Cao's location re-identification attack, the proposed fine-grained attack does not try to provide better performance on the success rate. Yet, in Fig. 12b, we can find that the spatial k-cloaking can hardly defend against the fine-grained attack on the size of the search area under various settings.

6 Optimization-Based Defense With Differential Privacy

In this section, we will describe a general differentially private framework to protect users' locations against the re-identification attacks in the sharing of POI frequencies. The proposed defense can also defend against the location re-identification

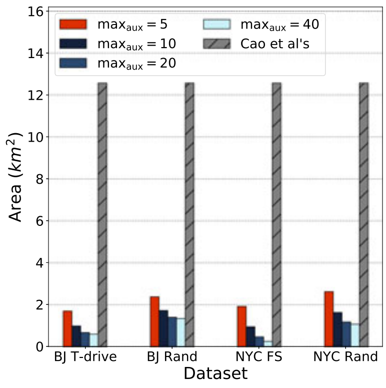


Fig. 9. Search area: changing the number of auxiliary anchors (the search area of Cao *et al.*'s attack is always $4\pi \text{km}^2$ in this setting).

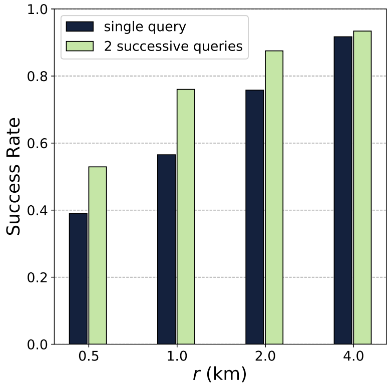
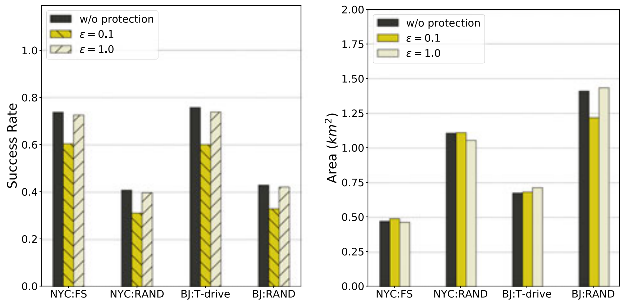
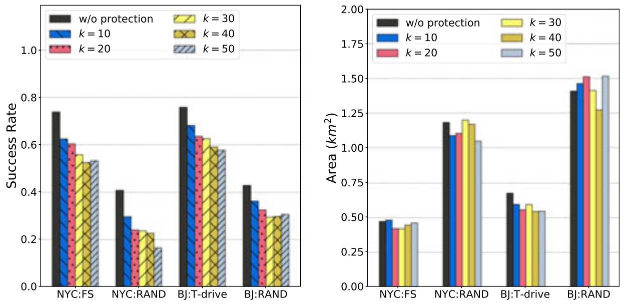


Fig. 10. Performance of the trajectory uniqueness: exploiting the power of two successive queries.



(a) Success rate of attacks (b) Search area of the attacks

Fig. 11. Fine-grained attack against Planar Laplacian.



(a) Success rate of attacks (b) Search area of the attacks

Fig. 12. Fine-grained attack against spatial k-cloaking.

attack and mitigate the privacy leakage in the cases where the user continuously releases the POI frequencies.

By revisiting Cao *et al.*'s attack [1], we can find that the region re-identification attack succeeds so long as the adversaries can locate the targeted users in areas with a radius r. Instead of finding the target user's exact location, the attacker tries to figure out to which POI the target is close. Under this setting, it is hard to achieve good defense performance with the location-level methods, i.e., only perturbing an actual location to a noise location seems not the right choice, and the evaluation of geo-indistinguishability also supports this argument. On the other hand, the aggregate-level methods can provide effective protection to some extent, but it is vulnerable to advanced attacks when the adversary obtains other background information, as has been shown in the evaluation of the sanitization. Besides, the aggregate-level protection may yield the POI type frequencies with poor utility because it will remove some essential information from the reported aggregate if we take an aggressive defense strategy. We resort to aggregate-level protection, which could perturb the POI type frequencies of users and make two improvements over the naïve sanitization method in Section 3.1:

- The naïve sanitization method does not take the utility into account, but it is crucial for services that need the POI type frequencies. We show that the proposed defense can provide the comparable utility of the perturbed frequencies.
- It has been shown that the naïve sanitization is vulnerable to advanced attacks with auxiliary information. The proposed defense provides a plausible guarantee of the perturbed frequencies against the auxiliary information-based attacks by introducing the notion of differential privacy.

There are two main challenges when supporting the multiple POI frequencies release: (1) the defense should be able to defend against the trajectory uniqueness; (2) the accumulated privacy loss should be audited carefully. A straightforward way of releasing the frequencies with respect to a trajectory of a user is to apply the mechanism we have proposed in Section 6.2 repeatedly. However, such a solution may introduce linearly increasingly privacy loss with the number of LBS queries based on the sequential composition rule(Theorem 4). Moreover, it is also not clear whether it can defend against the attack with trajectory uniqueness. The main idea is to characterize the correlation of POI frequencies of two successive locations on a trajectory. First, we incorporate the segment clustering algorithm and Gaussian mechanism to generate a dummy frequency vector for protecting the privacy of the original POI frequencies. Then, we incorporate the state-of-the-art Rényi differential privacy framework [14] to analyze the accumulated privacy loss for the multiple POI frequencies release in a fine-grained way. Moreover, we adopt a differentially private test to determine how to respond to the LBS requests such that the privacy budget can be further reduced.

6.1 Non-Private Formulation

We first formulate the perturbation objective in a non-private way. Assuming that a user requests the LBS with a sequence of locations $l_1, l_2, \ldots, l_L$, and the corresponding POI type frequency vectors are $F_{l_1,r}, F_{l_2,r}, \ldots, F_{l_L,r}$, we adopt the following optimization to find a proper release $\tilde{F}_t$ at time t:

$$\max_{\tilde{F}_t} \sum_{i=1}^{M} \frac{1}{R(i)} |\tilde{F}_t[i] - F_{l_t,r}[i]|, \tag{15}$$

$$s.t. \frac{1}{M} \sum_{i=1}^{M} \frac{1}{F_{l_t,r}[i]+1} |\tilde{F}_t[i] - F_{l_t,r}[i]| \leq \beta, \tag{16}$$

$$\frac{1}{M} \sum_{i=1}^{M} \frac{1}{\tilde{F}_{t-1}[i]+1} |\tilde{F}_t[i] - \tilde{F}_{t-1}[i]| \leq \eta, \tag{17}$$

$$\tilde{F}_t[i] \in \mathbb{N}^+,\ i = 1, 2, \ldots, M. \tag{18}$$

We want to maximize the weighted perturbation to the released frequencies while constraining the total distortion to the frequencies under a certain level, β. For the multiple POI frequencies release, we use Eq. (17) to limit the difference between two successive releases, such that the privacy leakage by trajectory uniqueness can be mitigated. In the above formulation, $R(i)$ is the infrequent rank of each POI type (the most infrequent POI type ranks 1, and so forth). For the case of single-shot aggregate POI distribution release, the parameter t and Eq. (17) can be omitted in the above formulation.

6.2 Differentially Private Release

The indistinguishability provided by the definition of DP guarantees that the released POI type frequency vector is insensitive to each POI type's frequency in the original frequency vector. To further illustrate the guarantee provided by DP, we specify the neighboring datasets in the release of POI frequency distribution. We refer to a pair of datasets $\mathbb{D}_1, \mathbb{D}_2 \in \mathcal{X}^d$ as neighbors if they are two POI frequency vectors and $\mathbb{D}_2$ can be obtained from $\mathbb{D}_1$ by only modifying one dimension of POI type frequency.

The main idea is to generate a privacy-preserving alternative to $F_{l_t,r}$ in Eq. (15) to Eq. (17), such that for time t, we can find a proper release $\tilde{F}^*_t$ which can defend against the re-identification attack and achieve differential privacy at the same time. Specifically, the defense mechanism consists of the following steps:

1) For a location l_t, if $\mathsf{dist}(l_{t-1}, l_t) < \text{Move}_{\min}$ (i.e., the user does not move significantly when making two successive LBS requests), then we use the POI type frequency vector of the previous location l_{t-1}, instead of generating a new POI type frequency vector $F_{l_t,r}$ based l_t. However, it may leak the information that the user's location does not change if we directly report the same frequencies. We adopt noisy version of the condition check as

$$\mathsf{dist}(l_{t-1}, l_t) + Lap\left(\frac{2\Delta_d}{\epsilon_c}\right) < \text{Move}_{\min} + Lap\left(\frac{2}{\epsilon_c}\right), \tag{19}$$

where Δ_d is the sensitivity of the dist function, and ϵ_c is the privacy budget for checking the condition. It is clear that the above check satisfies ϵ_c-differential privacy.

2) If $t > 1$, for a location l_t in the trajectory, the defense mechanism first prunes all the locations that have been submitted for a LBS request at time t and gets a location set $\mathcal{L}^t$. For $t > 1$, any location $d^t_j \in \mathcal{L}^t$ satisfies that

$$|\mathsf{dist}(d^{t-1}_j, d^t_j) - \mathsf{dist}(l_{t-1}, l_t)| \leq \gamma.$$

For each location $d^t_j \in \mathcal{L}^t$, a line segment can be constructed as (d^{t-1}_j, d^t_j), and the set of all line segments generated based on $\mathcal{L}^t$ is denoted by $\mathcal{S}^t$ (containing (l_{t-1}, l_t)). Then, we adopt the line segment clustering algorithm proposed by Lee *et al.* [15] to find a subset of segments, denoted by $\{(d^{t-1}_1, d^t_1), (d^{t-1}_2, d^t_2), \ldots, (d^{t-1}_k, d^t_k)\}$, in $\mathcal{S}^t$ as the dummies to protect the privacy of segment (l_{t-1}, l_t).

3) If $t = 1$ (i.e., the case of single-shot aggregate POI distribution release), the defense first adopts the spatial k-cloaking mechanism [12] to generate the a group of dummy locations as we have reviewed in Section 3.3, and the generated k locations (including l_t) are denoted by $d^t_1, d^t_2, \ldots, d^t_k$; if $t > 1$, we let $d^t_1, d^t_2, \ldots, d^t_k$ as the ends of segments $(d^{t-1}_1, d^t_1), (d^{t-1}_2, d^t_2), \ldots, (d^{t-1}_k, d^t_k)$, respectively. The POI frequency vectors are denoted by $F_{d^t_1,r}, F_{d^t_2,r}, \ldots, F_{d^t_k,r}$, respectively.

4) Then we compute the average of them with apply Gaussian mechanism for $i = 1, 2, \ldots, M$:

$$F^*_{\mathcal{D}_t,r}[i] = \left(\sum_{j=1}^{k} F_{d^t_j,r}[i] + \mathcal{N}(0, \sigma^2)\right)/k, \tag{20}$$

where we set $\sigma = \Delta\sqrt{2\ln(1.25/\delta)}/\epsilon$ according to Definition 2. ϵ and δ are privacy parameters.

5) Finally, we formulate the following optimization problem and get $\tilde{F}^*_t$:

$$\max_{\tilde{F}^*_t} \sum_{i=1}^{M} \frac{1}{R(i)} |\tilde{F}^*_t[i] - F^*_{\mathcal{D}_t,r}[i]|, \tag{21}$$

$$s.t. \frac{1}{M}\sum_{i=1}^{M} \frac{1}{F^*_{\mathcal{D}_t,r}[i] + 1} |\tilde{F}^*_t[i] - F^*_{\mathcal{D}_t,r}[i]| \leq \beta, \tag{22}$$

$$\frac{1}{M}\sum_{i=1}^{M} \frac{1}{\tilde{F}^*_{t-1}[i] + 1} |\tilde{F}^*_t[i] - \tilde{F}^*_{t-1}[i]| \leq \eta, \tag{23}$$

$$\tilde{F}^*_t[i] \in \mathbb{N}^+, \ i = 1, 2, \ldots, M. \tag{24}$$

For the cases where we only release aggregate POI distribution for a single location, the first two steps are not necessary, and the parameter t and the constraint Eq. (23) can also be omitted.

Deployment of the Defense. As we have shown in the system model, the POI aggregates can be generated by either the user itself or the geo-information service provider (GSP). For the existing protection methods that have been evaluated in Section 3, both the sanitization and the geo-indistinguishability can be deployed at either the user side or the GSP side, and the spatial k-cloaking requires the GSP to choose a region containing at least k users. The proposed defense in our paper also needs to be deployed on the GSP (or any trusted third party that can access the actual locations of users). In addition, the defense can be deployed in a hybrid setting where the step 1) condition check, step 2) location pruning, and 5) optimization can be implemented by the user itself, and only step 3) line segment clustering and step 4) noisy adding require the GSP to be involved.

6.3 Privacy Analysis

Theorem 5. *For each t, the above defense mechanism achieves (ϵ, δ)-differential privacy.*

Proof. First we analyze the sensitivity of sum of the POI type frequencies. Consider a pair of neighboring databases

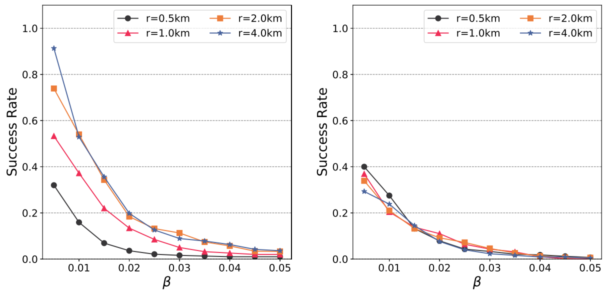
$$F_{d^t_1,r}, F_{d^t_2,r}, \ldots, F_{d^t_j,r}, \ldots, F_{d^t_k,r}$$

and

$$F_{d^t_1,r}, F_{d^t_2,r}, \ldots, F'_{d^t_j,r}, \ldots, F_{d^t_k,r},$$

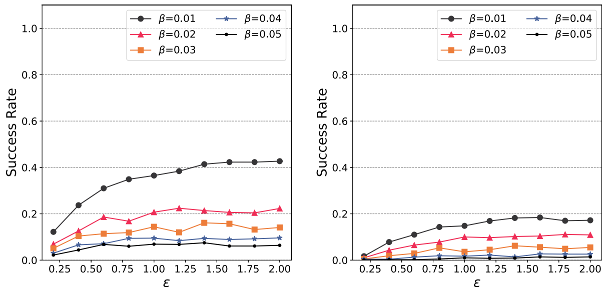
which differ in one POI frequency vector at one dimension. For any dimension i, $\sum_{j=1}^{k} F_{d^t_j,r}[i]$ will change at most $\max_d F_{d,r}[i]$, such that we can set the sensitivity at this dimension as $\max_d F_{d,r}[i]$.

Then we show that the publish of $F^*_{\mathcal{D}_t,r}[i]$ is differentially private. We set the variance $\sigma = \Delta\sqrt{2\ln(1.25/\delta)}/\epsilon$. By Definition 2, we have that Eq. (20) achieves (ϵ, δ)-differential privacy. In the optimization (Eq. (21)), we do not need to access the original POI frequency vector. The proposed defense mechanism is a sequentially apply of Eq. (20) and Eq. (21). By Lemma 3, it is (ϵ, δ)-differentially private. □



(a) BJ: T-drive (b) NYC: Foursquare

Fig. 13. The performance of the non-private defense mechanism (a lower success rate means better defense performance).



(a) BJ: T-drive (b) NYC: Foursquare

Fig. 14. The performance of the differentially private defense mechanism ($r = 2.0\text{km}$).

For the differentially private check in Step 1), we have that for a LBS request sequence with length R, the above defense mechanism achieves $(R_c\epsilon + R\epsilon_c, R_c\delta)$-differential privacy, where R_c is the number of locations that fail the differentially private check in Eq. (19) by applying the sequential composition rule Theorem 4. Then we use the notion of Rényi differential privacy (RDP) [14] to achieve a tighter bound of the accumulated privacy loss of releasing multiple POI frequencies. RDP is a variant of the standard DP. Instead of analyzing the privacy loss via KL divergence, RDP provides a way to bound the moment of privacy loss over any order by using Rényi divergence as a metric.

Definition 6 (Rényi differential privacy [14]). *A randomized mechanism $\mathcal{M} : \mathcal{X}^d \to \mathcal{Y}$ is (α, ϵ)-RDP if for any two neighboring databases $\mathbb{D}_1, \mathbb{D}_2 \in \mathcal{X}^d$, we have*

$$D_\alpha(\mathcal{M}(\mathbb{D}_1) \| \mathcal{M}(\mathbb{D}_2)) \leq \epsilon,$$

where D_α is the Rényi divergence between two probability distributions of order $\alpha > 1$.

We review some important results in the Rényi differential privacy, and we will use them to analyze the accumulated privacy loss of the defense mechanism.

Lemma 7 (RDP composition [14]). *Let $\mathcal{M} : \mathcal{X}^d \to \mathcal{Y}$ be an application of k randomized mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, and each mechanism $\mathcal{M}_i$ satisfies (α, ϵ_i)-RDP. Then $\mathcal{M}$ achieves $(\alpha, \sum_{i=1}^k \epsilon_i)$-RDP.*

Lemma 8 (RDP and DP [14]). *Let $\mathcal{M} : \mathcal{X}^d \to \mathcal{Y}$ be a (α, ϵ)-RDP mechanism, it also satisfies $(\epsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta)$-DP, for any $\alpha > 1$ and any $0 < \delta < 1$.*

When applying Gaussian mechanism or Laplace mechanism under the definition of RDP, there are the following results [14]. If a function $f : \mathcal{X}^d \to \mathcal{Y}$ has the sensitivity of Δ, then we have: a) the Gaussian mechanism applied to f: $f(x) + \mathcal{N}(0, \sigma^2)$ satisfies $(\alpha, \Delta^2\alpha/(2\sigma^2))$-RDP; b) the Laplace mechanism applied to f: $f(x) + Lap(\lambda)$ satisfies $(\alpha, \frac{1}{\alpha-1}\log(\frac{\alpha}{2\alpha-1}\exp(\frac{\alpha-1}{\lambda}) + \frac{\alpha-1}{2\alpha-1}\exp(-\frac{\alpha}{\lambda})))$-RDP, for any $\alpha > 1$ and any $0 < \delta < 1$. By the RDP composition rule, we have that the defense mechanism satisfies

$$\left(\alpha, \frac{R}{\alpha - 1}(\mu + \nu) + \frac{\alpha R_c}{4\ln(1.25/\delta)}\right)\text{-RDP},$$

where $\mu = \log(\frac{\alpha}{2\alpha-1}\exp(\frac{\epsilon_c(\alpha-1)}{2\Delta_d}) + \frac{\alpha-1}{2\alpha-1}\exp(-\frac{\epsilon_c\alpha}{2\Delta_d}))$ and $\nu = \log(\frac{\alpha}{2\alpha-1}\exp(\frac{\epsilon_c(\alpha-1)}{2}) + \frac{\alpha-1}{2\alpha-1}\exp(-\frac{\epsilon_c\alpha}{2}))$. Thus, we also have that the mechanism satisfies

$$\left(\frac{R}{\alpha - 1}(\mu + \nu)\right) + \frac{\alpha R_c}{4\ln(1.25/\delta)} + \frac{\log(1/\delta')}{\alpha - 1}, \delta')\text{-DP}.$$

7 Evaluating the Defenses

We have implemented the proposed defenses and evaluated them based on real-world user data traces. Our results show that the proposed defense can mitigate the location re-identification attacks to less than 20% success rate in most settings while well preserving the utility of the POI aggregates. The evaluation of the proposed defense mechanisms is performed on T-drive dataset and Foursquare NYC dataset.

7.1 Defense Performance

Fig. 13 shows the defense performance achieved by the non-private defense that is formulated in Eq. (15). We change the parameter β from 0.005 to 0.05, which is used to balance the utility and defense performance in the formulation. We can find that with the larger β, the mechanism performs better on the defense. When $\beta \geq 0.02$, the defense can mitigate the success rate of the attacks to less than 0.2 for various query ranges.

Fig. 14 shows the defense performance achieved by the differentially private defense mechanism that we have proposed in Section 6.2. We set the spatial cloaking parameter $k = 20$, privacy parameter $\delta = 0.2$, and change ϵ from 0.2 to 2.0. We can observe that for various choices of β, the defense performance gets worse when the privacy budget increases. In addition, the defense performance of the DP defense mechanism is also affected by the parameter β, and when β is small (say $\beta = 0.01$), which means the tolerated perturbation of the POI aggregate is small, the defense mechanism may need a strict privacy guarantee to achieve a satisfactory defense performance.

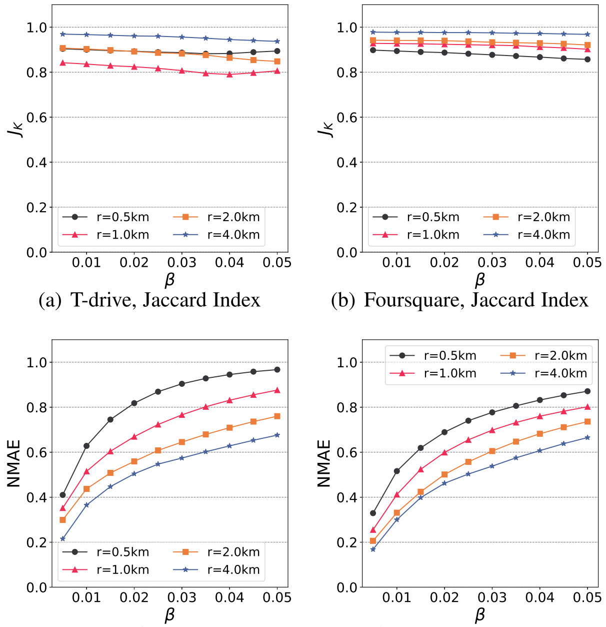
7.2 Service Quality

To measure the service quality, we use the following metrics.

1. Normalized Mean Absolute Error (NMAE). For the POI frequency distribution $F_{l,r} = (n_1, n_2, \ldots, n_M)$ and the protected POI frequency distribution $F^* = (n_1^*, n_2^*, \ldots, n_M^*)$, the Normalized Mean Absolute Error is calculated as

$$\texttt{NMAE} = \frac{\sum_{t=1}^M |n_t - n_t^*|}{\sum_{t=1}^M n_t}.$$

2. Jaccard Index on Top-K POI Types (J_K). We measure the service quality in a frequent POI type mining application



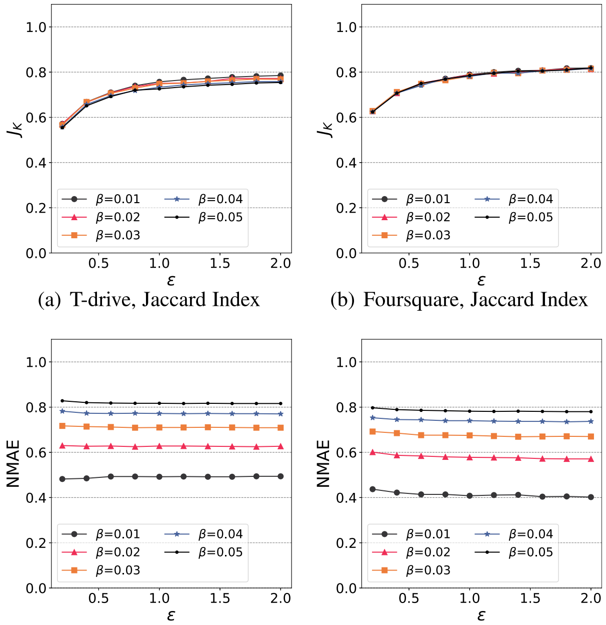
(a) T-drive, Jaccard Index (b) Foursquare, Jaccard Index

(c) T-drive, NMAE (d) Foursquare, NMAE

Fig. 15. Utility achieved by the non-private defense mechanism ($K = 10$).

which uses the top K most popular POI types in the POI frequency distribution. Specifically, for the POI frequency distribution $F_{l,r}$ and the protected POI frequency distribution $\tilde{F}^*$, we find the sets of K POI types with highest frequencies in the aggregates and denote them by $\mathrm{Top}(F_{l,r}, K)$ and $\mathrm{Top}(\tilde{F}^*, K)$. Jaccard Index [16] is adopted to measure the similarity between original POI type frequencies and protected POI type frequencies as

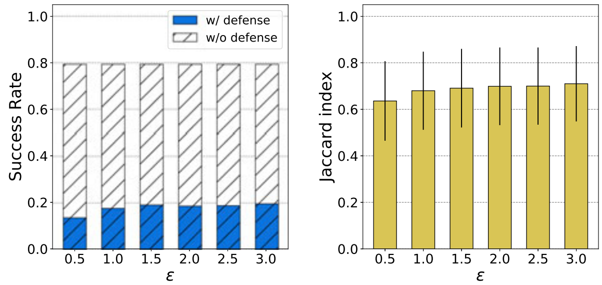
$$J_K(F_{l,r}, \tilde{F}^*) = \frac{|\mathrm{Top}(F_{l,r}, K) \cap \mathrm{Top}(\tilde{F}^*, K)|}{|\mathrm{Top}(F_{l,r}, K) \cup \mathrm{Top}(\tilde{F}^*, K)|}.$$



(a) T-drive, Jaccard Index (b) Foursquare, Jaccard Index

(c) T-drive, NMAE (d) Foursquare, NMAE

Fig. 16. Utility achieved by the differentially private defense mechanism ($r = 2.0\text{km}, K = 10$).



(a) Defense performance (b) Utility

Fig. 17. Performance of the multiple POI frequencies release defense.

Fig. 15 shows the results of the utility achieved by the non-private defense that is formulated in Eq. (15). For four settings of query range, the NMAE of the perturbed POI frequency distribution increases with β increasing. For the Jaccard index on Top-K POI types, we choose the top 10 most popular POI types in the POI frequency distribution. We observe J_10 decreases slightly with β increasing and is robust to the change of β.

Fig. 16 shows the results of the utility achieved by the differentially private defense. An interesting finding is that the change of NMAE is very slight with the change of ϵ. We speculate that the reason for this phenomenon is that for the whole POI frequency distribution, whose dimension is about 100, the effect of the optimization in Eq. (21) dominates the effect of the noise added in Eq. (20). On the contrary, we observe the metric J_{10} is sensitive to the noise parameter ϵ, but hardly changes with various values of β.

7.3 Multiple POI Frequencies Release

Fig. 17 shows the defense performance and the utility achieved by the multiple POI frequencies release mechanism. We change the privacy parameter from 0.5 to 3.0, and we can find that the success rate is significantly reduced from 79.36% to less than 20% for all the cases. We adopt top-10 as the target application, and the Jaccard index varies from $0.64(\pm 0.17)$ to $0.71(\pm 0.16)$ with the change of ϵ from 0.5 to 3.0.

8 Related Work

The related works of this paper fall into three categories: POI-based data analysis, location privacy, and uniqueness, and privacy of the aggregate location data.

8.1 POI-Based Analysis and Applications

POI data have been widely used in the applications of spatial-based analysis and recommendations. Some works, e.g., [17], [18] have been done for identifying the place with special meanings by leveraging POI data. Nishida *et al.* [18] propose a probabilistic identification method for personalized check-in in LBSs by analyzing users' past visited POIs. In [17], a clustering-based algorithm is proposed for analyzing the attractive areas by using crowdsourced data. POI-based recommendation also has been extensively studied, e.g., [19], [20], [21], [22], [23].

In [21], [22], the authors study the problem of time-aware POI recommendation to recommend POIs for a user to visit at a given time. Bin *et al.* [19] propose a personalized recommendation framework by leveraging users' multi-aspect behavior and preferences. POI data have also been used for human behavior analysis, e.g., [23], [24], [25], [26], [27], [28] Liu *et al.* [23] have developed a systematic framework to model POI demands by exploiting the daily needs of people identified from their large-scale mobility data. Park *et al.* [28] reveal the collective intelligence of the spatial choices expressed in the mobility patterns of the people that live in a city.

There are some previous works studying how to mine specific patterns based on POIs or the related information, e.g., [29], [30], [31], [32]. In [29], through the analysis of POI, the user habit pattern is obtained, and it is proposed that a vector representing the user's habit can be obtained through the processing of POI data. This paper proposes habit2vec, a representation learning method, which can generate a vector including users' living habit information through processing POI data. The work [30] explores the POI revisitation patterns and analyses the similarities and differences between website, smartphone app revisitation patterns and the POI revisitation. It uses global-scale check-in data and localization data in Beijing as datasets to complete its research. And it is the first work about the large-scale analysis of POI revisitation patterns in cities. Chen *et al.* [31] study the urban revisitation and re-check-in and explore the similarities and differences between them. It analyses the relevant factors that impact revisitation and re-check-in and examines the predictability. It is the first work about this. In [32], a pipeline system to detect popular temporal modes in population-scale unlabeled trajectory data is proposed. To test the system, it uses three large-scale real-world datasets to complete the experiment. And according to the evaluation, the system performs well in detecting popular temporal modes. This work contributes to uncovering the mechanisms behind urban mobility.

Lin *et al.* [33] propose a model called healthwalks, which takes mobility data as input to sense individual health conditions. It produces lots of mobility metrics. And then, through the process of regularization technique, the classification can be completed. To study the human mobility, Shi *et al.* [34] focus on the semantics of trajectory rather than simply focusing on the spatial and temporal patterns. And based on this, this paper proposes a semantics-aware hidden Markov model. Besides, to solve the data sparsity problem, this paper proposes a vMF mixture model. CROSSMAP [35] is a cross-modal representation learning model, which can make use of data to model people's activities and bridge the block of applying the geo-tagged social media.

8.2 Location Privacy With Uniqueness

A lot of research has been carried out on protecting location privacy, e.g., [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46]. Previous works on location privacy protection mainly focus on protecting users' geographical locations.

The uniqueness of location information has been studied by previous works, e.g., [1], [47], [48]. Montjoye *et al.* [47] show that human mobility traces have high uniqueness, and the uniqueness can be determined by users' mobility traces. In [48], an algorithm for anonymizing location data is proposed, which tries to decrease the privacy risk by this. The algorithm shortens the length of the trajectories. Through this, it lowers the uniqueness of the trajectories. Tu *et al.* [49] propose a protection method against semantic and re-identification attacks based on trajectory data. These works have considered the location privacy in presence of the uniqueness of locations or trajectories in terms of the users' geographical locations. In this paper, we try to understand the location privacy of the POI aggregate data instead of the geographical locations.

The most related work to our work is Cao *et al.* [1], which finds that even if the actual locations are not revealed, the adversary can still re-identify users' location by the aggregated POI type distribution. They observe the phenomenon of location uniqueness, which is ubiquitous in many metropolises. A computationally efficient location re-identification method is also proposed by [1]. However, as we have mentioned above, their method may not apply to launching practical attacks.

Some preliminary results in this paper have been published in [50], where the defense mechanism can only apply to the single POI frequency release. In this paper, we extend the study of location re-identification defense for a more general case.

8.3 Privacy of the Aggregate Location Data

POI type frequency can be viewed as a type of location aggregate data. Therefore, studies on aggregate data privacy are also related to our work. The aggregate data are often considered a way to hinder the exposure of individuals' data [51]. However, previous works show that various types of inference attacks on the aggregate data may still jeopardize users' privacy, e.g., [52], [53], [54], [55]. A pioneering work [52] by Pyrgelis *et al.* shows that an adversary with some prior knowledge can exploit aggregate information to improve her/his knowledge and even locate specific individuals that are part of the aggregates. Specifically, they consider the number of users that appear in a location at a certain time. The released aggregate data is generated based on the combinations of a set of regions of interest and a series of time slots. The adversary's goal is to profile the mobility pattern or infer the actual locations of the target users. Xu *et al.* [54] study how to recover users' trajectories, without any prior knowledge, from aggregate mobility data, which is a collection of the numbers of users at all the locations in a time slot.

Although it has been taken as an implicit assumption in [52], [54] that the membership information is known to the adversary, the membership privacy itself can also be critical when releasing aggregate location data. Pyrgelis *et al.* [53] try to figure out how an adversary can infer whether or not a target user appears in the group. In [53], a generic methodology is proposed for studying membership privacy on aggregate location data. Similar to [52], they consider the release of the aggregate location time-series, which represents the number of users in a group characterized in a spatial-temporal way. Based on [53], [55] conducts a systematical analysis of the membership inference attack on aggregate location time-series. By identifying the number of times a user appears in a location as the dominant feature of the membership inference attack, they have proposed a more advanced attack by leveraging the principal component analysis (PCA).

Unlike the previous attacks performed on the aggregate location time-series or the aggregate mobility data, our work tries to perform the inference attacks on the aggregate POI distribution, which a single user generates. Our goal is to infer users' locations or trajectories based on the aggregate POI distribution, which is similar to [52], [54], but our attacks are performed only on the aggregates of a single user and do not require the aggregate data of the whole system. In addition, since the aggregate POI distribution is generated by one user, the user-level membership inference attacks may not apply in our scenario.

9 Conclusion

In this paper, we conducted an in-depth study of the location privacy problem in the presence of location uniqueness. We have also conducted a study to evaluate whether the existing protection methods can adequately defend against the location re-identification attack. The results show that methods like sanitization, geo-indistinguishability, and spatial k-cloaking can hardly provide adequate location privacy protection in the presence of location uniqueness. Based on the existing location re-identification method, we present two practical variants that achieve higher precision in locating a user and better re-identification performance. Furthermore, we propose a differentially private POI type frequency release mechanism, which can be applied to both the single and the multiple POI aggregate data release. The evaluation shows that the proposed defenses provide adequate location privacy protection with acceptable utility loss.

Acknowledgments

The authors would like to thank all the reviewers for their helpful comments.

References

[1] H. Cao, J. Feng, Y. Li, and V. Kostakos, "Uniqueness in the city: Urban morphology and location privacy," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 62:1–62:20, 2018.

[2] O. Contributors, "OpenStreetMap," 2012. [Online]. Available: www.openstreetmap.org

[3] J. Yuan *et al.*, "T-Drive: Driving directions based on taxi trajectories," in *Proc. 18th ACM SIGSPATIAL Int. Symp. Adv. Geographic Inf. Syst.*, 2010, pp. 99–108.

[4] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[5] D. Yu, Y. Li, F. Xu, P. Zhang, and V. Kostakos, "Smartphone app usage prediction using points of interest," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 174:1–174:21, 2017.

[6] Y. Fan *et al.*, "Personalized context-aware collaborative online activity prediction," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 132:1–132:28, 2019.

[7] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[8] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.*, vol. 11, pp. 1471–1490, 2010.

[9] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 901–904.

[11] S. Oya, C. Troncoso, and F. Pérez-González, "Is geo-indistinguishability what you are looking for?," in *Proc. Workshop Privacy Electron. Soc.*, 2017, pp. 137–140.

[12] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst. Appl. Serv.*, 2003, pp. 31–42.

[13] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[14] I. Mironov, "Rényi differential privacy," in *Proc. 30th IEEE Comput. Secur. Found. Symp.*, 2017, pp. 263–275.

[15] J. Lee, J. Han, and K. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 593–604.

[16] Wikipedia contributors, "Jaccard index – wikipedia, the free encyclopedia," 2020. Accessed: Aug. 16, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Jaccard_index&oldid=965083523

[17] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proc. 1st Int. Conf. Exhib. Comput. Geospatial Res. Appl.*, 2010, pp. 1–4.

[18] K. Nishida, H. Toda, T. Kurashima, and Y. Suhara, "Probabilistic identification of visited point-of-interest for personalized automatic check-in," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 631–642.

[19] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1043–1051.

[20] B. Liu and H. Xiong, "Point-of-interest recommendation in location based social networks with topic and location awareness," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 396–404.

[21] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 363–372.

[22] Q. Yuan, G. Cong, and A. Sun, "Graph-based point-of-interest recommendation with geographical and temporal influences," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 659–668.

[23] Y. Liu, C. Liu, X. Lu, M. Teng, H. Zhu, and H. Xiong, "Point-of-interest demand modeling with human mobility patterns," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 947–955.

[24] D. Yu, Y. Li, F. Xu, P. Zhang, and V. Kostakos, "Smartphone app usage prediction using points of interest," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, 2018, Art. no. 174.

[25] G. Yu, J. Yuan, and Z. Liu, "Predicting human activities using spatio-temporal structure of interest points," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 1049–1052.

[26] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowd-sourced user footprints," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.

[27] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 712–725, Mar. 2015.

[28] S. Park, M. Bourqui, and E. Frias-Martinez, "Mobinsight: Understanding urban mobility with crowd-powered neighborhood characterizations," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops*, 2016, pp. 1312–1315.

[29] H. Cao, F. Xu, J. Sankaranarayanan, Y. Li, and H. Samet, "Habit2vec: Trajectory semantic embedding for living pattern recognition in population," *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1096–1108, May 2020.

[30] H. Cao, Z. Chen, F. Xu, Y. Li, and V. Kostakos, "Revisitation in urban space versus online: A comparison across POIs, websites, and smartphone apps," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 156:1–156:24, 2018.

[31] Z. Chen, H. Cao, H. Wang, F. Xu, V. Kostakos, and Y. Li, "Will you come back/check-in again?: Understanding characteristics leading to urban revisitation and re-check-in," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 3, pp. 76:1–76:27, 2020.

[32] F. Xu, T. Xia, H. Cao, Y. Li, F. Sun, and F. Meng, "Detecting popular temporal modes in population-scale unlabelled trajectory data," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 46:1–46:25, 2018.

[33] Z. Lin *et al.*, "Healthwalks: Sensing fine-grained individual health condition via mobility data," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 4, pp. 138:1–138:26, 2020.

[34] H. Shi, Y. Li, H. Cao, X. Zhou, C. Zhang, and V. Kostakos, "Semantics-aware hidden Markov model for human mobility," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 3, pp. 1183–1194, Mar. 2021.

[35] C. Zhang *et al.*, "Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 361–370.

[36] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Proc. 7th Int. Conf. Pervasive Comput. Pervasive*, 2009, pp. 390–397.

[37] W. Tong, J. Hua, and S. Zhong, "A jointly differentially private scheduling protocol for ridesharing services," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2444–2456, Oct. 2017.

[38] Y. Guo, H. Xie, C. Wang, and X. Jia, "Enabling privacy-preserving geographic range query in fog-enhanced IoT services," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2021.3095933.

[39] L. Wang, D. Yang, X. Han, D. Zhang, and X. Ma, "Mobile crowdsourcing task allocation with differential-and-distortion geo-obfuscation," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 967–981, Mar./Apr. 2021.

[40] J. Hua, W. Tong, F. Xu, and S. Zhong, "A geo-indistinguishable location perturbation mechanism for location-based services supporting frequent queries," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 5, pp. 1155–1168, May 2018.

[41] H. Shen, M. Zhang, H. Wang, F. Guo, and W. Susilo, "A lightweight privacy-preserving fair meeting location determination scheme," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3083–3093, Apr. 2020.

[42] Z. Fang, B. Fu, Z. Qin, F. Zhang, and D. Zhang, "Privatebus: Privacy identification and protection in large-scale bus WiFi systems," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–23, 2020.

[43] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2735–2749, 2020.

[44] W. Jin, M. Xiao, M. Li, and L. Guo, "If you do not care about it, sell it: Trading location privacy in mobile crowd sensing," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1045–1053.

[45] S. Narain, A. Ranganathan, and G. Noubir, "Security of GPS/INS based on-road location tracking systems," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 587–601.

[46] T. Zhou, Z. Cai, and F. Liu, "The crowd wisdom for location privacy of crowdsensing photos: Spear or shield?," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–23, Sep. 2021.

[47] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Sci. Rep.*, vol. 3, 2013, Art. no. 1376.

[48] Y. Song, D. Dahlmeier, and S. Bressan, "Not so unique in the crowd: A simple and effective algorithm for anonymizing location data," in *Proc. PIR, SIGIR CEUR Workshop Proc.*, 2014, pp. 19–24.

[49] 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 Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 264–278, Jan. 2019.

[50] W. Tong, C. Xia, J. Hua, Q. Li, and S. Zhong, "Practical location privacy attacks and defense on point-of-interest aggregates, ICDCS 2021, Washington DC, USA," in *Proc. 41st IEEE Int. Conf. Distrib. Comput. Syst.*, 2021, pp. 808–818.

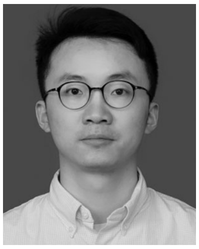
[51] R. A. Popa, A. J. Blumberg, H. Balakrishnan, and F. H. Li, "Privacy and accountability for location-based aggregate statistics," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, 2011, pp. 653–666.

[52] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "What does the crowd say about you? evaluating aggregation-based location privacy," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 4, pp. 156–176, 2017.

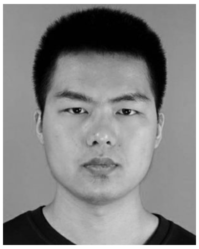
[53] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock knock, who's there? membership inference on aggregate location data," in *Proc. 25th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.

[54] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is NOT preserved in aggregated mobility data," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1241–1250.

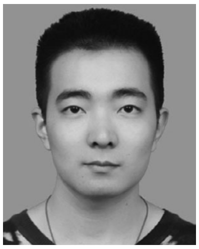
[55] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Measuring membership privacy on aggregate location time-series," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 4, no. 2, pp. 36:1–36:28, 2020.



Wei Tong (Member, IEEE) received the BS, MS, and the PhD degrees from Nanjing University, in 2013, 2016, and 2019, respectively. He is currently an assistant researcher with the Department of Computer Science and Technology, Nanjing University. He is interested in data privacy and information security.



Yinggang Tong (Graduate Student Member, IEEE) received the BS degree in computer science from Nankai University, in 2020. He is currently working toward the MS degree with the Department of Computer Science and Technology, Nanjing University. His current research interests include security and privacy.



Chang Xia received the PhD degree in the Department of Computer Science and Technology from Nanjing University, in 2021. His current research focuses on security and privacy in machine learning.

Jingyu Hua (Member, IEEE) received the BE and ME degrees in software engineering from the Dalian University of Technology, China, in 2007 and 2009, respectively, and the PhD degree in informatics from Kyushu University, Japan, in 2012. His current research interests include security and privacy in mobile computing, and system security.

Qun Li (Fellow, IEEE) received the PhD degree from Dartmouth College. His recent research focuses on wireless, mobile, and embedded systems, including pervasive computing, smart phones, energy efficiency, smart grid, smart health, cognitive radio, wireless LANs, mobile ad-hoc networks, sensor networks, and RFID systems.

Sheng Zhong (Senior Member, IEEE) received the BS and MS degrees in computer science from Nanjing University, in 1996 and 1999, respectively, and the PhD degree in computer science from Yale University, in 2004. He is interested in security, privacy, and economic incentives.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.