This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

Privacy Leakage of Location Sharing in Mobile Social Networks: Attacks and Defense

Huaxin Li, Haojin Zhu, *Senior Member, IEEE,* Suguo Du, Xiaohui Liang, *Member, IEEE,* and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Along with the popularity of mobile social networks (MSNs) is the increasing danger of privacy breaches due to user location exposures. In this work, we take an initial step towards quantifying location privacy leakage from MSNs by matching the users' shared locations with their real mobility traces. We conduct a three-week real-world experiment with 30 participants and discover that both direct location sharing (e.g., Weibo or Renren) and indirect location sharing (e.g., Wechat or Skout) can reveal a small percentage of users' real points of interests (POIs). We further propose a novel attack to allow an external adversary to infer the demographics (e.g., age, gender, education) after observing users' exposed location profiles. We implement such attack in a large real-world dataset involving 22,843 mobile users. The experimental results show that the attacker can effectively predict demographic attributes about users with some shared locations. To resist such attacks, we propose SmartMask, a context-based system-level privacy protection solution, designed to automatically learn users' privacy preferences under different contexts and provide a transparent privacy control for MSN users. The effectiveness and efficiency of SmartMask have been well validated by extensive experiments.

Index Terms—location privacy, mobile social networks, privacy protection.

1 Introduction

Mobile social networks (MSNs) are increasingly popular for enabling users to continuously sense their locations and social context via mobile social apps, and receive accurate and high-quality location-based and personalized service. Popular MSNs include Facebook, Weibo, Renren, Foursquare, Wechat, Momo, and Skout, which have the registered accounts more than half of the globe population. Among these MSNs, Facebook (1.3 billion monthly users), Twitter (900 million users), Weibo (500 million users), and Renren (214 million active users) [1], support *Direct Location Sharing*, e.g., location check-in, geo-location tag, and geo-location semantic comments, to disclose the exact locations of the mobile users to other users in the networks. Existing research works show that the direct location sharing mechanisms are effective for attracting attention, boosting self-presentation, and promoting and sustaining social capital [2]. Recently, a new class of location-sharing application, called *location-based social discovery*, enables users in physical proximity to disclose their relative distances to other users in physical proximity. We define this kind of location sharing as *Indirect Location Sharing*. The typical example is Wechat, one of the most popular MSNs in China supporting location-based social discovery, has more than 600 million registered user accounts over 200 countries. Other examples are, Momo with 100 million registered user accounts and 40 million active users per month [3], and Skout, a very popular dating app in North America, with 1.5 million new users a month [4].

Although the MSNs greatly benefit our social life by integrating the cyber space with the physical world, the MSNs increase the danger of user privacy breaches due to the direct and indirect location sharing. There are quite a few studies addressing location privacy issues in MSNs [5], [6], [7], [8], [9]. However, little attention has been paid to quantify private information leaking issues arising from location sharing in MSN. In this paper, we will study the private information leaking in MSNs from three perspectives. i) We aim to understand how accurate the location information disclosed by the current location sharing mechanisms can reflect the users' real location patterns in terms of coverage and distribution. ii) Based on the shared locations [9], [10], [11], is it possible for the attacker to dig out more sensitive information such as the demographic information (e.g., the ages, genders)? iii) Last but not least, though it is well known that privacy enhancement cannot come for free, how to achieve the trade off between the privacy control and loss of the utility represents a great challenge. The challenge is that the concept of privacy may change significantly due to the contexts and locations, user's behavior, and many other factors [14], while the users may fail to change their privacy settings accordingly in the existing solutions [12], [13], [14]. Our motivation is to provide the users with a fine-grained and user-transparent privacy control mechanism, which is expected to provide different privacy controls for different contexts in a user friendly way. For example, the most visiting locations are most closely related to the user's Top Locations (e.g., home or work place) [7], thus deserve a higher privacy protection level. On the other hand, the public regions can be assigned with a lower privacy level to guarantee the MSN's service quality. What is the most important, the proposed framework should automatically sense the contexts and translate them into corresponding privacy levels, which does not require the excessive involvement of human operations.

In this work, we first take an empirical study on measurement of the location privacy leaking in MSNs, before we propose an effective privacy control approach. We focus on two real-world

- *Huaxin Li, Haojin Zhu and Suguo Du are with Shanghai Jiao Tong University, China. E-mail: lihuaxin003@sjtu.edu.cn, zhu-hj@cs.sjtu.edu.cn, sgdu@sjtu.edu.cn.*
- *Xiaohui Liang is with Computer Science Department at University of Massachusetts Boston. E-mail: Xiaohui.Liang@dartmouth.edu.*
- *Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo,Waterloo, Ontario, Canada. E-mail: sshen@uwaterloo.ca.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

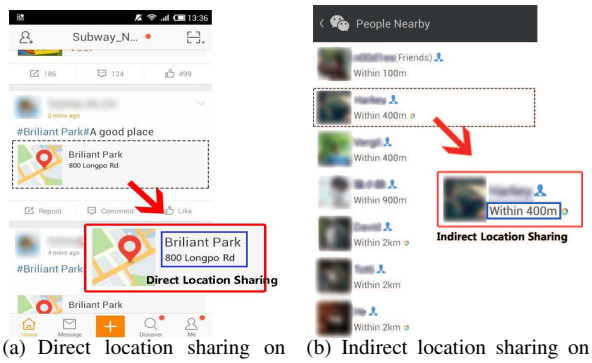
location datasets. The first dataset is a three-week experiment involving 30 volunteers. We collect GPS coordinates as the ground truth trajectories of volunteers with our customized sensing apps. We also obtain the disclosed location information from their MSN applications. The second dataset contains 5 months' location trajectories of 22,843 users, which are collected from a campus Wi-Fi network. We analyze the first dataset to understand how accurate the disclosed locations by the current location sharing mechanism can reflect the users' real location patterns in terms of coverage and distribution. We analyze both datasets to launch and propose two algorithms to simulate the demographic inference attack. The contributions of our work are summarized as follow:

- We measure the similarity between the disclosed locations in the MSN applications and the real mobility pattern, in terms of two novel defined metrics including coverage rate and the relative entropy. In particular, it is found that the direct location sharing and the indirect location sharing only reveal 16% and 33% of POIs and the relative entropy 1.68 and 0.92, respectively. The empirical study shows that there is a big gap between the shared location profile and the user's real mobility pattern.
- Though the shared locations only reveal a small portion of the real mobility patterns of human beings, we present a new attack, which is expected to infer users' demographics from the disclosed locations by checking their similar POIs. To infer a specific attribute of a user, We identify a group of users with similar location traces, and check the attribute of another user in the same group who publicly reveals her attributes. Our experiments, based on two datasets, show that our inference technique efficiently predicts demographic attributes (age, gender, occupation, education level and living place) that are very often hidden by users even with incomplete shared locations.
- We propose SmartMask, a system-level privacy control approach to provide a transparent privacy control for the users. SmartMask simply defines the privacy requirements based on location contexts, such as location, frequency of visits, and duration of visits. Users input some initial settings by a user-specified interface, and SmartMask automatically learns users' privacy preferences under different contexts by the Decision Tree model. The automatic privacy level configuration and fine-grained obfuscation module are achieved based on automatic location profile management, location context classification and user preference learning. SmartMask is also designed to be compatible with the existing obfuscation techniques [29].

The remainder of this paper is organized as follows. In Section 2, the background, attacker model and the datasets are introduced. In Section 3, we present two metrics to quantify the difference between the disclosed locations and the real trace. In Section 4, we propose novel inference techniques that can reveal the users' demographics from their shared locations. In Section 5, a system level privacy protection approach, named SmartMask, is presented. In Section 7, related works are introduced. In Section 8, we conclude this paper.

2 Location Sharing in MSNs

Current behavior disclosure on social network sites like Facebook reveals that users are generous to share their information [2]. In this work, we consider two kinds of location sharing in MSNs as is shown in Fig 1.



(a) Direct location sharing on Weibo (b) Indirect location sharing on Wechat

Fig. 1: Location Sharing on Mobile Social Networks

2.1 Direct Location Sharing in MSNs

Most of the popular social networks such as Facebook, Twitter, Foursquare, Weibo, and Renren provide the following location based sharing functionalities:

- *Geolocation Tags*: Mobile social networks (e.g., Facebook, Weibo, Renren) provide users with the option to reveal (or redact) location data on posts. A typical approach for implementing Geolocation tags is to add geographic information to an object, such as a photograph.
- *Check-in Services*: Check-in on mobile social networks (MSNs) like Facebook, Weibo, and Foursquare is another popular kind of location based service (LBS). It reveals users' mobility traces and can be potentially exploited by the adversary.
- *Location-dependent Comments*: Users' comments in MSNs might also involve their location information. A good example for location-dependent comments is Yelp or Dianping, which provide online reviews for local services (e.g., a restaurant or shop). An online review about the services of this restaurant/shop normally indicates the visiting of this restaurant/shop, which allows the attacker to correlate mobility traces with the physical location of this restaurant.

Based on Geolocation Tags and Check-in services, the attacker can easily obtain the location information shared by the mobile users by crawling down the interested information from web pages, extracting POIs from collected data, or even automatically transforming the name of this location to a GPS coordinate or vice versa. We denote such kind of location sharing as direct location sharing.

2.2 Indirect Location Sharing in MSNs

It is also witnessed that the location-based social discovery networks (e.g, Wechat, Momo and Skout), are quickly gaining popularity. These new MSNs explicitly enable on-the-spot connection establishments among users based on physical proximity. Instead of pinning users' exact locations on a map, it provides an implicit way for location sharing: it only displays coarse-grained proximity information, such as "Jack is within 3 miles". The latest works point out that, by only exploiting the public available information,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

the attacker can collect the users' almost exact locations, reveal their mobility traces as long as they are using the corresponding proximity-based social discovery functionality [5], [6], [7], [8]. We denote it as indirect location sharing as, in location-based social discovery networks, the users' locations can be obtained by the outside observers if the following two conditions are satisfied:

- *Locations are shared by the users:* The user's action of performing a proximity-based social discovery actually grants the permission of sharing his locations with others.
- *Users' location sharing activities are observed by outsiders:* Proximity-based social discovery apps only display the latest location of a specific user. The user's location of a specific time slot can be obtained by the outsider if and only if his sharing location activity is observed, otherwise his historical traces will be covered by the subsequent location sharing.

Different from the direct location sharing in which the users' historic traces can always be accessed by the outsider, the indirect location sharing requires the outsider to actively track the target to build the target's trace. For more details about the attacks towards Wechat, Momo, Skout or even facebook, please refer to [5], [6], [7].

2.3 Attacker Models

The goal of the adversary is to build the location profile of the target victim and infer the sensitive data or demographics of the target. In particular, we consider two kinds of adversaries corresponding to two location sharing approaches:

- *Casual Tracking Attacker:* For the direct location sharing, the attacker can always access the target's location history and build his location profile if the users' information is public [2], [15]. Therefore, the attacker can launch the attack anytime and this attack can be casual and last for a random duration. The typical examples of casual tracking attacker model include Facebook, Foursquare, Weibo and Renren.
- *Continuous Tracking Attacker:* For the indirect location sharing, the attacker should perform a long term tracking towards the victim [5], [6], [7]. The attacker can collect the users' almost exact locations, reveal their mobility traces without physical contact, as long as they are using the corresponding proximity-based social discovery functionality. The considered typical examples of this attack model include Wechat, Momo and Skout. According to our previous study [7], more than 80% of tracking results on Momo can geolocate the victims in 40m, more than 90% of tracking results on Skout geo-locate the victims to 0-20m and 80-100m, and over half of the tracking on Wechat users can be located to the accuracy of less than 60m.

In both cases, the capability of the attacker is limited to exploiting the public available information on the Internet without hacking the system of the service provider. However, as discussion above, the attackers can still infer the users' locations accurately (so we use the term "shared locations" and "inferred locations" equally under the attacker model in this paper). The attacker aims to obtain the location information of the target, build his location profile, and infer the demographics. The following part of this paper will investigate these problems based on the practical attack model.

2.4 Datasets

2.4.1 Dataset I: A Real-world MSN Dataset

The first dataset is a real-world MSN dataset, which is denoted as *Dataset I*. We recruited 30 volunteers from different departments and different grades on campus to collect their directly shared locations from Weibo and Renren (*Direct Sharing Trace*), indirectly shared locations from Wechat, Momo and Skout (*Indirect Sharing Trace*), and mobility traces at fixed interval (*Ground Truth Trace*). For *Direct Sharing Trace*, we crawled down the users' personal information, such as age, gender, interests, hometown, and user ID from MSN webpage. Note that, some of these profiles are normally kept private by the users such as age, occupation and living place. We discuss how to infer the demographics in Section 4. Then, with the user ID, we collected traces of the users' check-ins and location-tags using open API. For *Indirect Sharing Trace*, we collected traces when the volunteers use the geosocial functionalities such as *look around* and *shake hands* by the approaches presented in [5], [6], [7]. The detailed information of the traces is listed as follows:

- *Direct Sharing Trace* contains 252 Weibo and Renren Check-ins and location tags. Each check-in or location tag includes a timestamp, the name of a POI, and the location coordinate.
- *Indirect Sharing Trace* contains 2,404 coordinate records collected from volunteers using Wechat, Skout and Momo. It captures volunteers' inferred location when they use location based service. Each coordinate record includes a timestamp, a GPS coordinate, the app name, and the user's nickname.
- *Ground Truth Trace* contains totally 886,737 GPS coordinate records collected by the sensing apps developed by us. It captures each volunteer's GPS location every 30 minutes. Each coordinate record includes a timestamp, a GPS coordinate, and the user's nickname.

2.4.2 Dataset II: A Large Scale Real-world Dataset from Wi-Fi Traffic

In addition to the small scale real-world experiments above, we collected the second dataset. *Dataset II* is a set of large scale real-world Wi-Fi traffic records which involve data of 22,843 users within 5 months. This dataset has MSNs traffic logs from 98 Wi-Fi hotspots deployed on the campus and the MSNs traffic logs record location trajectories. Each log contains anonymized user id, MSN name, location, and access time. Meanwhile, the dataset also provides anonymized user attributes such as gender and education level, which provides the ground truth to evaluate the performance of the proposed inference results. Table 1 shows the distribution of the users' demographic attributes.

TABLE 1: User demographic attributes distribution

Gender		Education		
Male	Female	Bachelor	Master	Doctor
11509	11334	11509	7896	3438
50.4%	49.6%	50.4%	34.6%	15.0%

Justification of Utilizing Dataset II to Simulate The Data Sharing: To avoid the dataset bias of *Dataset I*, we introduce *Dataset II* as an important complement of *Dataset I* to implement the inference attack as well as countermeasure evaluation. Note that, location sharing in MSNs reveals a part of real POIs of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

users. Exposed locations from Wi-Fi access history in *Dataset II* also gave a partial review of users' real POIs. Therefore, it is reasonable to simulate location sharing in MSNs by exploiting *Dataset II* with different location sharing probabilities. In order to further justify the utilizing of *Dataset II*, we define a metric "data sharing rate" that defines the possibility that a user shares his location in MSNs. The results of *demographic inference* (which will be discussed in Section 4) under different data sharing rates are shown in Appendix.

In Section 3, we use *Dataset I* to quantify similarity between shared locations and ground truth mobility traces. In Section 4, we use both datasets to infer demographics with different methods.

3 Quantifying Leakage of Shared Locations

The first problem in this work is how well the shared location data correspond to each user's physical mobility patterns. In this section, we explore this problem by first giving a formal definition of location profile, and then proposing two metrics to measure the distance between shared location profiles and the users' physical location profiles.

3.1 The Location Profile of MSNs

Different from previous research works which build location profile according to phone call log, GPS navigation and intended GPS tracking applications [16], [17], [18], location profile from MSNs has special properties. On one hand, granularity of locations varies from place to place, because real locations will be clustered as POIs. On the other hand, the exact information in temporal, such as exact entering time and exiting time of a POI, is difficult to be obtained in MSNs. So an adversary can't perform real time tracking via MSNs.

Considering a set of users $\mathcal{U} = \{u_1, \ldots, u_n\}$ who enjoy a wide range of location based services provided by various MSNs. The considered area is partitioned into a finite set of POIs which represent the locations within minimum granularity, i.e., $\mathcal{R} = \{r_1, \ldots, r_N\}$. The user's trajectory refers to his movements along the spatial and temporal domain. We model the trajectory of a specific user u as a function mapping a time point in $\mathcal{T}$ to the user's location in $\mathcal{R}$ at that time, i.e., $\alpha_u : \mathcal{T} \rightarrow \mathcal{R}$. Thus, if the user traverses k POIs for k different time slots, the ground truth mobility traces of the user u can be denoted as $\mathcal{M} =<< u, t_1, \alpha_u(t_1) >, \ldots, < u, t_k, \alpha_u(t_k) >>$. In mobile social networks, the attackers can exploit various opportunities to collect the snapshots of mobility traces of mobile users. The mobility traces collected by the attacker can be denoted as $\widetilde{\mathcal{M}} \subseteq \mathcal{M}$.

From the trajectory $\mathcal{M}$, we can obtain an aggregate view on a user's mobility pattern by building his location profile, which includes the user's visited location set $\mathcal{L} = \{l_1, \ldots, l_n\}$ and its discrete probability distribution as $\theta_i = P(l_i)$. Here $P(l_i)$ denotes percentage of visits of l_i. Similarly, from the shared trajectory collected by the attackers $\widetilde{\mathcal{M}}$, the attacker can also obtain the inferred location profile, including the inferred location set $\mathcal{L}' = \{l'_1, \ldots, l'_n\}$ and its discrete probability distribution θ', using the methods mentioned in Section 2.3.

To capture the degree of predictability of the mobile user's mobility, we consider an entropy-based definition to model the users' mobility pattern [21], which is defined as follows

Entropy of Mobility Patterns *Let $\mathcal{L}$ be the user u's physical location profile, we define the uncertainty and thus the entropy of this user mobility pattern as*

$$\mathcal{E}(\mathcal{L}) = -\sum_{l \in \mathcal{L}} \theta(l)\ log_2 \theta(l) \tag{1}$$

Similarly, from the inferred location profile $\mathcal{L}'$, we can obtain the entropy of the inferred user mobility pattern as

$$\mathcal{E}(\mathcal{L}') = -\sum_{l \in \mathcal{L}'} \theta'(l)\ log_2 \theta'(l) \tag{2}$$

The entropy of user's mobility pattern is determined not only by number of POIs that he visited but also by the frequency of the visitation. Therefore, the entropy $\mathcal{E}(\mathcal{L})$ has the maximum value when the probability of visiting each location follows a uniform probability. On the other hand, $\mathcal{E}(\mathcal{L})$ has the minimum value when the probability of staying at one location is dominant (e.g., $\mathrm{p}_i = 1$).

Whereas, the properties of location profile in MSNs, as discussed above, make it difficult to reveal user's real location profile. To well quantify leakage of shared locations, we propose two novel metrics to quantify the similarity between shared locations and real mobility patterns, which indicate how much privacy leaks out from the location sharing and are introduced in details in the following section.

3.2 The Comparison of Real and Inferred Location Profiles

The shared locations are only a partial view of a user's real mobility traces, which means that $\mathcal{L} \subset \mathcal{L}'$ or $\theta \subset \theta'$. To measure how much $\mathcal{L}$ can match $\mathcal{L}'$, we introduce the first metric:

Metric I: N-Location Coverage Rate *Given $\mathcal{L}$ as the ground truth location set, $\mathcal{L}'$ as the inferred location set, we define $Sel()$ as the function that returns all of POIs of a specific location set and define $Sel_N(\mathcal{L})$ as the function that returns N POIs from $\mathcal{L}$ by following a certain rule. We define N Location Coverage Rate as*

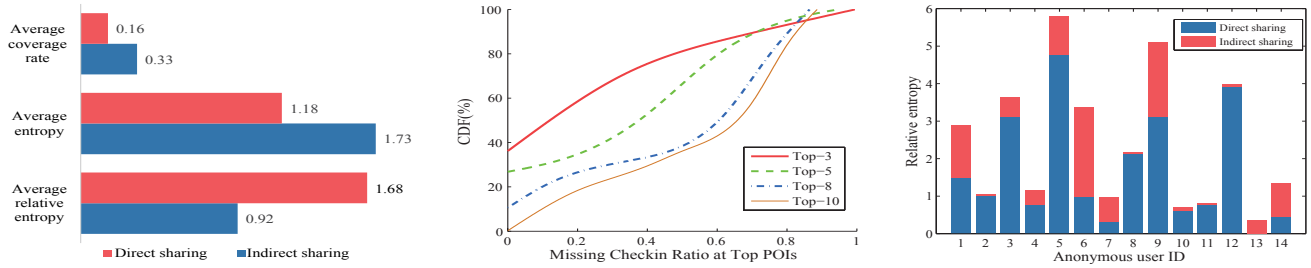
$$TNR = \frac{|Sel_N(\mathcal{L}) \cap Sel(\mathcal{L}')|}{N} \tag{3}$$

which refers to the percentage of locations that belong to both of selected N locations in both the real location set and the inferred one.

N-Coverage Rate can be used to evaluate how many POIs are exposed from shared locations in MSNs. For example, the existing research works point out that "Top N" locations refer to the locations that are most correlated to users' identities (e.g.,"top 2" locations correspond to home and work locations) [19]. Thus, in the context of top N location discovery, N-Coverage Rate can be re-defined to *Top N Location Coverage Rate*. In other cases, if the attackers care more about the most semantic sensitive POIs (e.g, hospitals, clubs and etc), N-Coverage Rate can be defined as N Sensitive Location Coverage Rate.

N-Coverage Rate alone cannot well describe how much location privacy is leaked from location sharing in MSNs. This is because, according to the entropy definition of users' mobility pattern, a user's mobility pattern should be based on not only how many POIs he visits but also its probability distribution. To measure the distance between the ground truth location profile and the inferred location profile from probability distribution point of view, we give the following relative-entropy based definition,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing



(a) Average of the metrics in Datasets (b) Missing check-in ratio of indirect sharing (c) Relative entropy of selected users

Fig. 2: Comparison of shared mobility and ground truth traces

which is regarded as a good measure of the distance between two distributions in Information Theory [20].

Metric II: Relative Entropy of The Real and Inferred Location Profile *Let the location set $\mathcal{S} = \mathcal{L} \cap \mathcal{L}'$ refer to the set of the locations that belong to a user's real location profile and inferred location. Let θ and θ' be discrete probability distribution on location set $\mathcal{S}$ for real location profile and inferred location profile, respectively. Then, we can define the relative entropy of these two distributions and thus the confidence on the inferred location profile as*

$$D_{kl}(\theta, \theta') = \sum_{s \in \mathcal{S}} \theta'(s) \cdot log \frac{\theta'(s)}{\theta(s)} \tag{4}$$

It is obvious that this metric is always non-negative and is zero if and only if $\theta = \theta'$. In the next subsection, we will show the matching results from the real-world experiments.

3.3 Quantifying Location Profile Similarity

We evaluate the location profile similarity using *Dataset I* described in Section 2.4 and compare the two metrics with the ground truth.

3.3.1 N-Location Coverage Rate

N-Location Coverage Rate represents how many POIs are exposed from the locations shared by users in MSNs. Firstly, we investigate an extreme case that sets $N = |\mathcal{L}|$, which measures the percentage of the real users' mobility traces that are included in the shared locations. From Fig 2(a), it is shown that, Direct Sharing Trace and Indirect Sharing Trace can achieve 16% and 33% coverage rate. We further investigate coverage rate of users' most sensitive locations by setting $N = 3, 5, 8, 10$. The Top N coverage rate of Direct and Indirect Sharing Trace are 38.1%, 34.3%, 26.8%, 25.7% and $65.1\%, 56.2\%, 45.2\%, 39.5\%$, respectively. We are also interested in which locations users are not checking in at. The intuition here is, due to the existence of usage pattern, the users may not share their locations at specific locations. To validate it, we identify the top-N most visited POIs of each user, and examine the portion of their missing check-ins. The missing check-in radio for top N locations means the percentage of users who have missing check-ins in their top N location. Fig 2(b) plots the CDF of the missing check-in radio for all of the users for their top 3, top 5, top 8 and top 10 locations. The results show that some locations account for a large portion of missing check-ins, which demonstrates the users do not share their most sensitive locations in many cases.

3.3.2 Relative Entropy

Relative entropy represents the difference between the location distribution of the users' shared location and their physical visiting. We firstly look into the entropy of users' shared locations from Direct Sharing Trace and Indirect Sharing Trace of real-world experiment, which is shown in Fig 2(a). For Direct Sharing Trace, which mainly includes the shared data from Weibo and Renren, it is observed that its average entropy is 1.18 while the average entropy of Indirect Sharing Trace is 1.73. The former is lower because the users may choose not to share locations at some places directly (such as private locations or Top 2 locations) while these locations can be exposed to an attackers indirectly [7]. The latter is more consistent with the existing research works [21], which points out that the human's mobility is highly predictable and that the real uncertainty in a typical user's whereabouts is about 1.74, fewer than two locations [21]. As is shown in Fig 2(c), this is also demonstrated by evaluating their corresponding relative entropy, in which Direct Sharing Trace has a much higher relative entropy. This shows that Indirect Sharing Trace (or the shared data from Wechat, Momo or Skout) is closer to the users' real mobility pattern, and can be used to learn more information about the victim, compared with the Direct Sharing Trace.

3.3.3 Discussion

There is a diversified range of factors which contribute to the difference between the real and inferred location profiles, including: users' usage pattern, wireless access, as well as the different social network platforms. Among the potential factors, users' usage patterns significantly affect the inferred location profiles. The existing research works show that users typically do not check in at places which they think are "boring" or "private", which leads to a generally low Top 5 location coverage rate [15]. In our experiments, an interesting observation is that the most sensitive locations (Top 2 location) of the users' real traces may not rank the highest in the inferred trace, which leads to the inference of users' attributes (especially the demographic attributes) from their traces not so straightforward. In the next section, we will introduce a novel inference technique based on similar-trace users selection.

It is also noted that there exists a clear difference between public accessible social networks (e.g., Weibo and Renren), which are mainly represented by Direct Sharing Trace, and location based social discovery networks (e.g, Wechat, Momo and Skout), which are mainly represented by Indirect Sharing Trace. The former shows a much lower coverage rate, which is consistent with the research on Foursquare check-in dataset [15]. This is because the shared locations in Weibo and Renren are permanently saved in the networks and will be accessible for everyone, which make the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

users hesitate to share their sensitive locations. Instead, Wechat, Momo and Skout are geo-social applications and sharing location is their major feature, which makes the users easier to share their locations.

From the above discussion, it can be concluded that the inferred location profiles only reveal a small percentage of the real location profiles of the users. In the next section, we will study if an adversary can exploit this incomplete information to infer the user's demographics.

4 From Mobility Traces to Demographics

4.1 Overview

Existing research works demonstrate approaches to making automatic semantic annotations of places [22], [23] that can be used to infer users' activities and public attributes. However, in the previous section, we have shown that the locations shared in MSNs are only a partial view of the users' mobility traces. For example, the users may choose not to check in at their private places (such as Top 2 locations). Further, some demographics such as genders and ages are not highly relevant to the users' mobility patterns. Hence, inferring hidden or private profile/attributes is a non-trivial task. In this section, we present different approaches to infer demographic attributes according to the different scale of datasets.

We start from a novel common-trace based approach to infer demographics of a target user in a small scale. Our observation is that users with similar traces have similar personal profiles, thus our algorithm is based on the famous algorithm Longest Common String [24]. Hence, to infer a specific user's demographics in his profile, we can identify a group users who share similar traces and reveal their personal profiles publicly. Then, we can infer the hidden values from the similar-trace users' public attribute values. Previous works predicted the user's movement among locations and recognized individual activities on each location based on location similarity [25]. Different from previous works, our work uses location similarity to infer user demographic attributes.

Further, we propose a more scalable machine learning approach and apply it to *Dataset II* for demographic inference in a large scale data set. And finally, we compare merits of these two approaches and discuss their applications.

4.2 A Maximum Common Trace based Demographic Inference Algorithm

4.2.1 Work Flow

As illustrated by Fig 3, the proposed inference algorithm consists of the following steps:

- Collecting Users' Mobility Traces: The attackers collect the mobility traces from the different users of a specific region.
- Finding out the Maximal Common Trace between The Target and Other Users: By setting a target, the attacker finds out the maximal common trace between this specific target and the other users.
- Calculating the Similarity Score: According to location semantics contained in each maximal common trace, the attacker can calculate a similarity score between the target and other users, and then rank these users in terms of their similarity score.
- Inferring Demographics: The attacker can infer the target's demographic attributes from top k users and do majority voting for the Demographics.

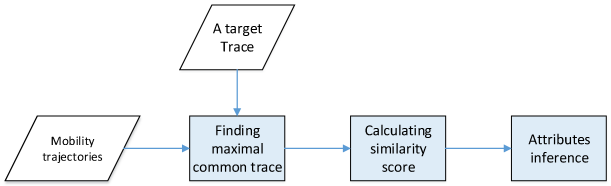


Fig. 3: The architecture of common-trace based inference approach

4.2.2 Finding the Maximal Common Trace

We propose a Maximal Common Trace (MCT) matching algorithm to find the maximal common trace of two traces. Given two location traces A and B, we define the Maximal Common Trace as a subsequence containing the maximum length of common locations sequence without changing the original time order in both A and B. The basic idea of MCT is to use a table $L[i, j]$ to record the length of maximal common trace between the sub-trace before the ith location of trace A and the sub-trace before jth location of trace B. Once we acquire the table of $L[i, j]$, it will be easy to output the trace by backtracking the table. The algorithm is summarized in Algorithm 1. The proposed algorithm needs a two dimensional array with the size of $m \times n$ to store the length of common subsequence and thus its complexity is $O(m \times n)$. This complexity can be further reduced to $O(m + n)$ if the users' traces are maintained as the suffice tree [24].

Algorithm 1 Maximal Common Trace $\mathbf{MCT}(\mathcal{M}_1, \mathcal{M}_2)$

1: **Input**: Two attack semantic traces $\mathcal{M}_1$ and $\mathcal{M}_2$, where $p = |\mathcal{M}_1|$ and $q = |\mathcal{M}_2|$.
2: **Output**: $\mathcal{C}$: maximal common location trace of $\mathcal{M}_1$ and $\mathcal{M}_2$.
3:
4: Construct a two dimensional array $L[p, q]$.
5: Record the length of maximal common location:
6: **if** $\mathcal{M}_1[i] = \mathcal{M}_2[j]$ **then** $L[i, j] \leftarrow L[i-1, j-1] + 1$
7: **else**
8: $\quad L[i, j] \leftarrow \max|L[i, j-1], L[i-1, j]|$
9: **end if**
10: Backtracking $L[p, q]$ to construct $\mathcal{C}$
11: Output $\mathcal{C}$

4.2.3 Calculating the Similarity Score

The calculation of trace similarity is based not only on the sensitiveness of different POIs but also on the localization precision of the shared locations. For the former, the sensitiveness of different POIs are not uniform. The POIs which are more related to the users' identities or demographics, such as Top Locations (e.g., home or work place) [7], are automatically assigned with a higher weight according to the visiting frequency and the location semantics while the public regions (e.g., public square or cafe) can be assigned with a lower weight. For the latter, a higher localization precision or a lower coverage of shared location provides more information to the attacker. Therefore, we assign different levels of *sensitiveness* to different POIs and different *granularity* to different collected shared locations. Both sensitiveness and granularity will be represented by the metric weight, assigned by the function r_w. Then, the scores of similarity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

between two users' trace $\mathcal{M}_1$ and $\mathcal{M}_2$ are calculated by Equation 5 and 6:

$$\mathcal{C} = \mathbf{MCT}(\mathcal{M}_1, \mathcal{M}_2) \tag{5}$$

$$\mathbf{Score}(\mathcal{M}_1, \mathcal{M}_2, \mathcal{C}) = \frac{\sum_{i=1}^{|\mathcal{C}|\times|\mathcal{C}|} r_w(\mathcal{C}[i])}{|\mathcal{M}_1| \times |\mathcal{M}_2|} \tag{6}$$

where $\mathcal{C}[i]$ represents the ith location of common trace $\mathcal{C}$, and $|\mathcal{M}_1|$, $|\mathcal{M}_2|$ and $|\mathcal{C}|$ represent the length of $\mathcal{M}_1, \mathcal{M}_2$ and $\mathcal{C}$ respectively.

Algorithm 2 Inference of a certain demographic ($\mathcal{M}$,$\mathcal{N}$)

1: **Input**: The target's trace $\mathcal{M}$ and trace dataset $\mathcal{N}$
2: **Output**: The inferred attribute $\mathcal{A}$
3: Users' scores $\mathcal{S} = \emptyset$
4: **for** $N_i \in \mathcal{N}$ **do**
5: $\quad \mathcal{C} = \mathbf{MCT}(\mathcal{M}, N_i)$
6: $\quad s = \mathbf{Score}(\mathcal{M}, N_i, \mathcal{C})$
7: $\quad$ insert s into $\mathcal{S}$
8: **end for**
9: $\mathcal{S}' = \mathbf{Sel}(k, \mathcal{S})$
10: $\mathcal{A} = \mathbf{MVoting}(\mathcal{S}')$
11: output $\mathcal{A}$

4.2.4 Inferring Demographics

We can infer a user's demographic attributes from that of the k most similar users, whose traces are the most similar to this user's trace. The detailed inference algorithm is presented as follows: First, we select the k most similar users out of all whose attribute is defined and public available. Then, based on the collected the k most similar users as well as their public attributes, Then we infer the target user's demographic attributes by performing majority voting. In other words, the attribute value which is shared by the most users out of the total k most similar users is chosen. If there is more than one attribute having the maximal number of votes, we will randomly pick one. The algorithm is summarized in Algorithm 2. Here we can use the function $\mathbf{Sel}()$ to represent the function of selecting the k most similar users based on the similarity scores and the function $\mathbf{MVoting}()$ to represent the function of making a majority voting for the value of demographic attributes based on the k most similar users' public profiles.

The proposed inference algorithm is based on the observation that people who share similar mobility patterns are likely to have more common attributes than people who don't. Therefore, it is required that the inferred demographics should be closely related to the users' mobility patterns. However, its effectiveness may be limited in the case that the demographics are not closely related or even loosely related to the users' mobility. In our experiments, we will show the correlation between the selected demographics (age, occupation, living place) and the users' mobility.

4.2.5 Evaluation

In order to validate our common-trace based inference, we firstly perform the experiments based on *Dataset I*. We select Occupation, Age, Living place from *Dataset I* as demographic attributes in our experiments. We set top 5 users of similarity ranking for majority voting and the baseline is set to the most likely value for all users (i.e. the demographic attribute x belongs to most users). We perform the inference towards the users by following the previously described techniques and then compare the inference results with ground truth data which are collected from a survey of the experiment participants.

TABLE 2: Inference Accuracy for Demographics

(a) Inference Accuracy of Dataset I

Demographic	Baseline	Random guess	Inference
Occupation	54.2%	33.3%	69.2%
Age	42.7%	20%	53.8%
Living place	37.5%	20%	54.5%

(b) Inference Accuracy of Dataset II

Demographic	Baseline	Random guess	Inference
Gender	50.3%	50%	73%
Education level	50.4%	33.3%	76%

Table 2(a) shows that the demographics of Occupation, Age and Living Place can be inferred at high accuracy rates, which are much higher than the accuracy rate of baseline guesses and random guesses. Taking the Occupation Inference as an example, our algorithm performs 15% better than the baseline guess and about 35.8% better than the random guess. Compared with previous work [26], our techniques also achieve a considerable successful rate.

To further validate the proposed algorithm, we also evaluate the degree of similarity between two trace similarity rankings and the similarity ranking of the demographics in ground truth data by using three kinds of correlation coefficient: Kendall correlation, Pearson correlation and Spearman correlation. The results are shown in Fig 4, which depicts the mean, maximum, minimum and median of these correlation coefficients. The mean of three kinds of correlation coefficients are around or above 0.5. It shows that there is a strong correlation between trace similarity and the attribute similarity which further demonstrates the motivation of the proposed algorithm.

4.3 Exploiting Machine Learning for Demographic Inference in A Large Scale Data Set

The previously proposed common trace based inference algorithm can achieve a higher accuracy at the cost of a higher complexity. In this sub-section, we introduce a machine learning based inference scheme which is suitable for the large dataset.

4.3.1 Problem Definition

As we have discussed above, mobile location profile $\mathcal{L}$ can be collected by an attacker. Based on the collected location profiles or mobility pattern, the goal of the attacker is to infer similar demographics based on mining the mobility pattern from large scale data. So it can be interpreted as a supervised classification problem in machine learning:

$$\Phi(\mathcal{L}_u) = d_u \tag{7}$$

where the input $\mathcal{L}_u$ is the location profile of a user u, and the output d_u is the predicted demographic label of the user u. The classification model Φ can be previously trained by a set of users with demographic information and their location profiles. Once the model has been trained, it can be used to infer other users' demographic information, according to their location profiles.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

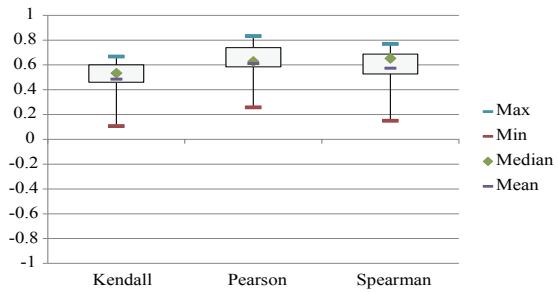


Fig. 4: Correlation coefficients of trace similarity ranking and demographics similarity ranking

4.3.2 Implementation

In practice, we implement a set of popular supervised classification algorithms (K-Nearest Neighbors, SVM, Decision Tree, Random Forest, AdaBoost, Logistic Regression, Naive Bayes) using *scikit-learn* [27] to find the best classifier for predicting demographics. We take all locations in our dataset as features in the learning process. If a user shows up at a location, the value of this location is set to 1, otherwise if a user never appears at a location, the value of this location is set to 0 (we also take frequencies of visiting locations as features, the result is similar). In this way, we translate the location profile $\mathcal{L}$ into the feature vector $\mathcal{F}$. In the training phase, a set of feature sets and their corresponding users' demographics are used to train the model Φ. In the predicting phase, we can infer a user's demographic information d_u based on his/her feature vector $\mathcal{F}_u$.

4.3.3 Evaluation

To validate this approach, we perform experiments on *Dataset II* and try to infer the gender and education level within the 22,834 users. For each group of experiments, we randomly selected 50% of users data as training set and 50% as testing set and repeated experiments for five times. We also take the most likely value for all users (i.e. the demographic attribute belonging to the most users) and the random guessing value as baseline and reference. Table 2(b) shows that for gender inference and education level inference, our algorithm outperforms the baseline for 22.7% and 44.7%, and outperforms the random guess for 23.0% and 42.7%, respectively. Our machine learning based approach shows that MSNs users' demographics can be inferred in a large scale.

4.4 Comparison of The Two Approaches

To further compare the Maximum Common Trace based inference approach described in Section 4.2 and Machine Learning based inference approach presented in Section 4.3, we randomly select 2000 users with equal number of males and females and 2000 users with equal number of bachelors, masters and doctors. The rate of successful inference of common-trace based approach is 78% for education levels attribute and 73% for gender attribute, and the rate of successful inference of machine learning approaches is 65% for education level attribute and 62% gender attribute. It shows that the first approach performs better than machine learning approach. But the running time of common-trace based approach is almost 496 times longer than machine learning based approach under same amount of data and same running environment. Thus we propose different methodologies under different circumstances.

5 Countermeasure Design and Implementation

In this section, we develop SmartMask, a novel privacy protection framework which aims to provide the fine-grained privacy management for MSN users. The main idea of SmartMask is to balance the tradeoff of privacy protection and utility by assigning the privacy levels according to different locations and user preferences. Our insight is that the mobile users tend to have more social demands in the public places (e.g., pub or coffee shops) while having a higher privacy-preserving need for Top 2 locations (e.g., working place or home). Different from the previous works which focus on how to obfuscate the data, SmartMask has the following desirable properties:

- *Context-driven Privacy Management*: Privacy is context dependent as is pointed out by [14]. Similarly, in MSNs, users also have different privacy preferences in different contexts. For examples, visiting a hospital is obviously more sensitive than visiting a garden because the former may raise the health concerns while the latter is positioned as a social spot. Top locations (or most visiting places, e.g., home or work place) are more closely related to users' identities, thus much more sensitive. Visiting a bar or hotel in the morning and in the evening may have different implications. Therefore, locations, visiting frequency, staying time are the factors of the contexts. SmartMask should take them into consideration and support context based privacy management.
- *Fine-grained Location Privacy Control*: SmartMask allows a fine-grained privacy level assignment to meet the diversified demand of the different users. For ease of presentation, we consider a system with three privacy levels, including high, medium and low. Under this privacy setting, the most visiting locations are most closely related to the user identity (Top Locations) and thus deserve stronger obfuscation when they are shared in MSNs. So the privacy level of Top Locations should be defined as the high level. On the other hand, the public regions (e.g., scenery spot) can be assigned with the low privacy level, which means less obfuscation, to guarantee its service quality.
- *Automatic Privacy Level Assignment*: Fine-grained location privacy control can be achieved only if the system can automatically assign privacy levels based on different contexts. The higher sensitive contexts lead to a higher privacy protection level and a bigger obfuscation range. In case that the automatic privacy level settings do not fully match users' preferences, SmartMask also allows the users to set the privacy levels for specific locations or apps.

5.1 Framework Implementation

We design and implement SmartMask as a location-privacy preserving module in Android system. Fig 6 illustrates the architecture of SmartMask, which comprises of four components.

Contexts Generator: Collecting user's location information and storing user's mobility history in the local database. We implement a LocationProfile class in Android framework. This class provides methods to execute clustering algorithm in the mobility dataset, and then extract user's location profile. Location profile records user's frequency, duration, time period of visiting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

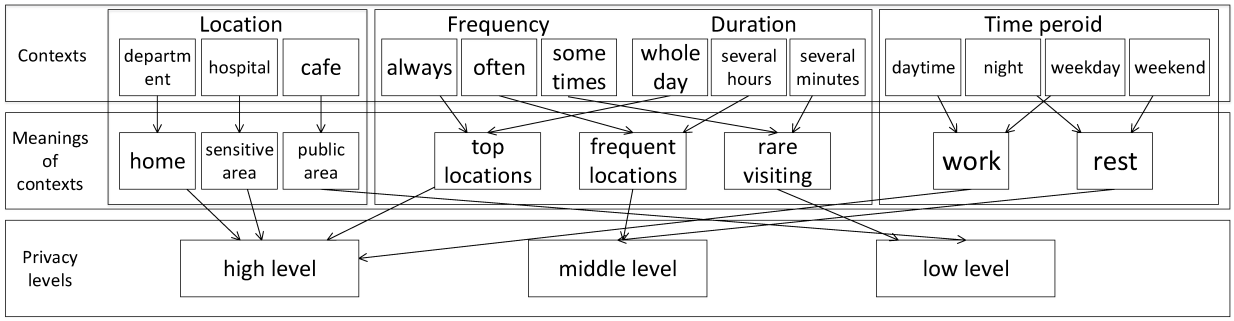


Fig. 5: Context acquisition for privacy level

a location, which will be assigned a privacy preserving level according to their sensitiveness.

Privacy-level Generator: Assigning the sensitive level for each geographic position and LBS app based on different contexts. There are three kinds of privacy level in SmartMask, low, medium and high. To set the privacy levels automatically, a decision tree model is introduced at server and assigns privacy levels based on the contexts. Privacy level information is stored as XML file in the SystemDir of Android file system or in secure cloud server and will be used as obfuscation parameters in our obfuscating process. We will discuss automatic privacy level assignment in Section 5.2.

User-specified Interface: Provides an interface for users to specify their location sharing preferences. User-specified interface was implemented as an system application in Android. This application can report the LBS app by scanning the manifest file to find *ACCESS_COARSE_LOCATION* or *ACCESS_FINE_LOCATION* permission. The application can also show the geographic position in a map. And then users can specify privacy preserving level for LBS apps and geographic positions to meet their requirements.

Obfuscation Engine: This module is designed to implement obfuscation techniques and execute obfuscation process. We introduce a function, ObfuscateLocation(), in LocationManager class. ObfuscateLocation() provides the interface for different obfuscation techniques. Parameters of this function are the original location, the package name of query app, and the privacy preserving level. We also modify Android location requesting APIs so that whenever they are invoked by a querying app, they will call obfuscateLocation() and return the obfuscated results to the querying app.

5.2 Automatic Privacy Level Assignment based on Contexts

5.2.1 Privacy Level

As we mentioned above, users have different privacy preferences in different location contexts. So different privacy levels should be assigned to different locations so that different privacy preserving strategies can be applied. For example, the most visiting locations are most closely related to the users identity (Top Locations) and thus deserve stronger obfuscation when they are shared in MSNs. So the privacy level of Top Locations should be defined as high level. On the other hand, the public regions (e.g., scenery spot) can be assigned with a lower privacy level, which means less obfuscation, to guarantee its service quality.

Without loss of generality, we consider four kinds of features in the context recognition, including *Attributes of locations*, which divides the locations to different classes according to attributes of

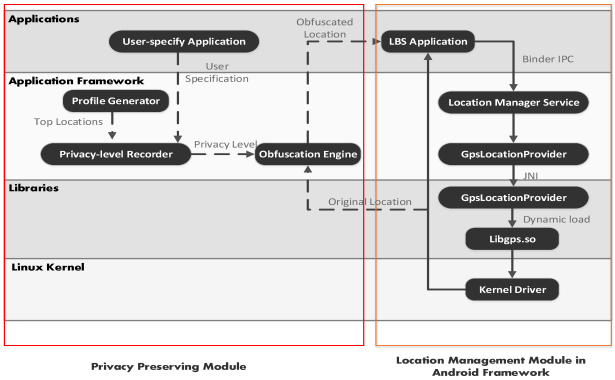


Fig. 6: SmartMask architecture

location semantics, *Frequency*, *Duration* and *Time period*. Fig. 5 illustrates an example, three classes privacy levels are classified according to the different features of location contexts. Whereas, assigning privacy levels based on the features of contexts is non-trivial because some of the features are dependent and non-linear. To address this issue, we propose a decision tree based automatic privacy level assignment approach. Decision tree models employ human readable if-then-else statements which perfectly fit the considered problem. And the cost of adopting decision tree is logarithmic in terms of the number of training data.

5.2.2 Decision Tree Model

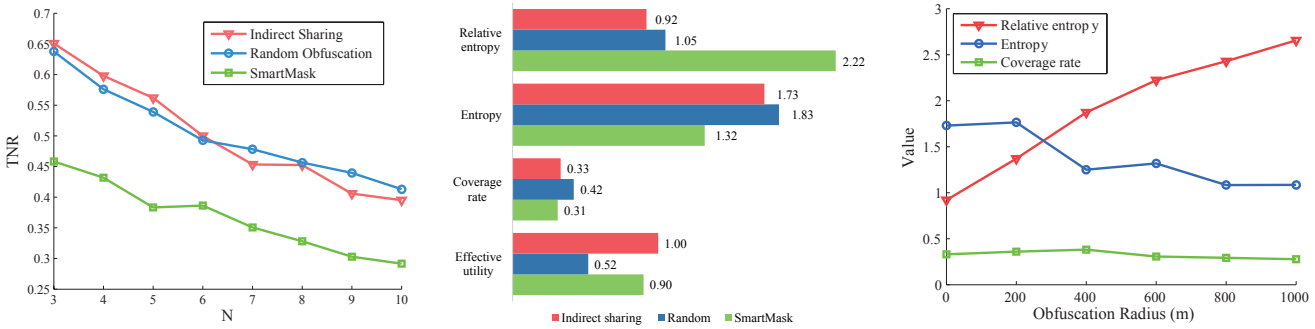
Our decision tree model takes a set of location features $\mathcal{F}$ as prior knowledge. Then the model aims to predict privacy level of other users' visiting locations $\mathcal{L}$. So it can be formulated as a classifier Ψ which predicts privacy level $j \in \mathcal{J} = \{1, ..., J\}$ at the input $\mathcal{L}$ over independent replicates of the learning set $\mathcal{F}$, which is denoted as:

$$\Psi(\mathcal{L}, \mathcal{F}) = j \tag{8}$$

Given a D dimensional feature vector $\mathcal{F} = \{x_1, x_2, ..., x_D\}$, a decision tree h is a collection of nodes n_i organized in a hierarchical tree structure. Node can be a split node or a terminal leaf node. Assuming a binary decision tree, for each split node n_i, the splitting function $f(\mathcal{F}, \pi_i, \phi_i)$ can be represented as:

$$f(\mathcal{F}, \pi_i, \phi_i) = \begin{cases} 1 & \text{if } \mathcal{F}_{\pi_i} > \phi_i \\ 0 & \text{if } \mathcal{F}_{\pi_i} < \phi_i \end{cases} \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing



(a) TNR with different obfuscation techniques (b) Comparison of obfuscation techniques (c) Impact from different obfuscation ratio

Fig. 7: Evaluations on SmartMask

where $\pi_i \in \{1, ..., D\}$ is feature index and ϕ_i is the threshold to divide two classes. For privacy level labels $\mathcal{J}$, prediction result $j \in \mathcal{J}$ of the decision tree h can be formulated as decision:

$$d(\mathbf{u}, j) = 1 \tag{10}$$

while

$$d(\mathbf{u}, \mathcal{J}/\{j\}) = 0 \tag{11}$$

In training phase, decision tree h grows with a number of features $\mathcal{F}_k$ ($\subset \mathcal{F}$) specified at each node n_i. The $\mathcal{F}_k$ features are selected at random out of the $\mathcal{F}$. The best splitting on these $\mathcal{F}_k$ is used to split the node and form splitting function $f(\mathcal{F}_k, \pi_i, \phi_i)$ (which is introduced above). The criterion of information gain $I(\cdot)$ is taken as reference when splitting the node. Feature index π_i^* and threshold ϕ_i^* can be chosen as:

$$\pi_i^*, \phi_i^* = \underset{\pi,\phi}{\operatorname{argmax}}\, I(\mathcal{F}_k, \pi, \phi) \tag{12}$$

The splitting ends when a predefined depth is reached or a leaf is reached.

In prediction phase, given a feature vector as input, the tree is traversed according to the Equation 9 until a leaf is found. To obtain the result of prediction, the majority voting rule is applied: label of the leaf is derived from the class with majority of training samples that finished in this leaf. If the count is same for all classes, the label is chosen randomly.

5.2.3 Implementation

In particular, we implement a Classification and Regression Trees(CART) [28]. CART constructs a binary tree using the features and thresholds that yield the largest reduction in entropy. Features with larger entropy reduction are likely to be more distinct among the classes. Hence they are chosen first while building the decision tree from root to leaves. Then, we apply reduced error pruning to reduce the tree size.

To validate the effectiveness, we manually set the privacy levels of contexts in our *Dataset II* and randomly select half of users as training set and use others as testing set. The precision, recall and F1-score achieve 89%, 88% and 89%, which demonstrates the effectiveness of our proposed privacy level learning. Fig 8 illustrates our decision tree model based on training data.

5.3 The Obfuscation Mechanisms in SmartMask

There is a large body of research works on location obfuscation algorithms [29], [30]. We believe the novel location obfuscation

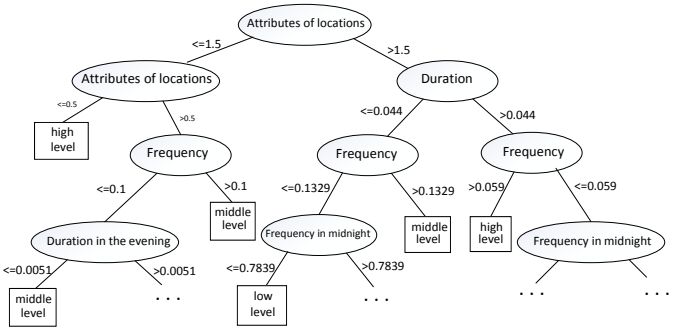


Fig. 8: Decision tree model for learning privacy levels

algorithm design is an important research topic and deserves separate research. We implement a hybrid obfuscation method based on two different obfuscation techniques. The first obfuscation technique was proposed in [29], which includes three basic obfuscation operators, *radius enlargement*, *radius reduction* and *center shifting*. Given the local measurement $A_m = (x_m, y_m, r_m)$ returned by sensing technology where x_m, y_m are the coordinates of the center of A_m and r_m is A_m's radius, we can achieve an obfuscated area by random combination of the following operations:

radius enlargement:

$$(x_f, y_f, r_f) = (x_m, y_m, r_m\sqrt{\frac{R_m}{R_f}}) \tag{13}$$

radius reduction:

$$(x_f, y_f, r_f) = (x_m, y_m, r_m\sqrt{\frac{R_f}{R_m}}) \tag{14}$$

center shifting:

$$(x_f, y_f, r_f) = (x_m + dsin\theta, y_m + dcos\theta, r_m) \tag{15}$$

where $\theta \in [0, 2\pi]$, d is a random distance generated in a range that decided by privacy levels, R_m and R_f are two implementations of the *Relevance*, which is defined in [29], of the local measurement area A_m and obfuscated area A_f, respectively.

When privacy level is low or medium (or the LBS has a certain requirement on the utility), SmartMask executes obfuscation by randomly combining these obfuscation operators (low privacy level means a slighter obfuscation than middle level). In the case of high privacy level, the users don't want to disclose their real

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

locations in these privacy levels. So SmartMask adopts a simple cloaking strategy that makes the obfuscation results deviated to the nearest public and less sensitive region (e.g, shopping malls or cinema). This technique is useful for defending malicious or unwanted location requests.

5.4 Evaluation Results

Our evaluation is two-fold. To evaluate properties and merits of SmartMask, we conduct another 3-week experiment, in which 15 volunteers on campus take part using the modified Android system that incorporates SmartMask. In the first 2 weeks, we gather each user's location data to generate location profile. In the last week, SmartMask begins to obfuscate each location request according to location profile and users' preferences. To evaluate effect of hiding demographics of users in a large scale, we perform SmartMask's strategy on *Dataset II*.

5.4.1 Properties of SmartMask

Comparison of Obfuscation Mechanisms: For the two different obfuscation mechanisms, we first consider the N-coverage rate which reflects a user's N most private places. As is shown in Fig 7(a), SmartMask's N-Coverage Rate decreases significantly, but Random Obfuscation's N-Coverage Rate doesn't decrease. This is reasonable because SmartMask shifts locations towards some certain spots (public regions), which makes top N locations invisible for adversary. In contrast, Random Obfuscation shifts locations and disperses the distribution randomly, which won't decrease the coverage. So there are many chances that shifted visits are still within top N locations and observed by an adversary. The comparison of entropy and coverage rate in Fig 7(b) also demonstrates the observation.

Next, we consider the relative entropy metric. It is obvious that relative entropy of SmartMask greatly increases by 1.3 while that of random obfuscation only increases 0.13. As is discussed in section 3.3, relative entropy represents the difference between distribution of shared locations and distribution of ground truth locations. This result also proves that SmartMask performs better in the aspect of hiding original location profiles.

Sensitivity Analysis: To further assess the effectiveness of SmartMask, we test SmartMask with different obfuscation radii ranging from 0m to 1000m. In Fig 7(c), the coverage rate does not vary significantly, while the relative entropy increases apparently. We can conclude that compared with original locations, probability distributions of obfuscated locations vary significantly, which leads to conspicuous change of location profile, so that the top N locations can be hidden.

5.4.2 Privacy and Utility Trade-offs

Obfuscation techniques will lead to the decrease of the utility. To evaluate decrease of the utility, we define the *Utility* metric as

$$Utility = 1 - \frac{Min(Dist(l_o, l_r), MaxDist)}{MaxDist} \qquad (16)$$

where l_o represents the original location, l_r represents the obfuscated result, *MaxDist* represents the maximum deviation error that the user could tolerate, which is set by users, and function *Dist()* returns the distance of two locations. It is obvious that, when the obfuscated result is the same as original location, the *Utility* achieves the maximum value 1. When $Dist(l_o, l_r)$ is no smaller than *MaxDist*, the *Utility* is 0.

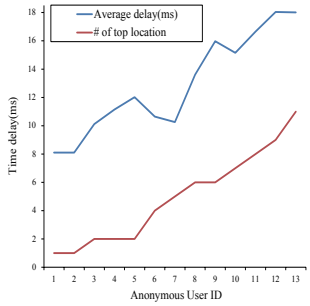


Fig. 9: Time delay

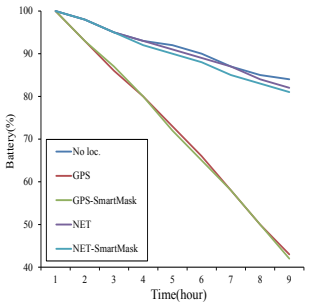


Fig. 10: Energy overhead

To evaluate MSN utility, we further define *EffectiveUtility* as the average utility of those locations which are not in top locations. Since users care more about the LBS utility than privacy in these locations, *EffectiveUtility* can accurately evaluate obfuscation's influence on MSN utility. Fig 7(b) shows that *EffectiveUtility* of SmartMask is far larger than random obfuscation.

5.4.3 Demographics Hiding

We use our *Dataset II* to evaluate effect of demographics preservation. We set obfuscation radii as 100, 300, 500, 1000 respectively and apply the same method as Section 4. The results are shown in Table 3.

TABLE 3: Inference accuracy of demographics preservation

obfs.radii(m)	100	300	500	1000	no obfs.
Gender	0.57	0.56	0.54	0.47	0.73
Edu	0.62	0.61	0.57	0.51	0.76

Along with increase of obfuscation radii, which indicates obfuscation strength, the metrics decrease accordingly. For a moderate obfuscation radii, i.e. 500 meters, accuracy of *Gender* decreases 19% and accuracy of *Education* obviously decrease 19%. These results indicate SmartMask is useful to reduce probability that an attacker successfully infers the targeted user's demographics. It must be admitted that SmartMask can't preserve demographics to the extent of random guess. This result is reasonable because our goal is not to totally balance the distribution of each class of people but to achieve trade-off of privacy and service quality. As discussed in [13], [14], in the age of information, privacy protection involves lots of aspects. Combining different techniques, such as data preservation, anonymity, and obfuscation is the best way to preserve privacy.

5.4.4 Performance and Energy consuming

We evaluate the performance of SmartMask by measuring time delay to obfuscate locations. We simulate location access through apps based on location profiles of real-world datesets. According to our design, time delay should be likely to increase with the increase of the scale of location profile. The evaluation results are shown in Fig 9. Even in the worst case, average delay of 18.04ms does not impact the application usability.

We evaluate the energy consuming by measuring the rate of battery depletion in the following scenarios: no location access, a load of location access of one request per 3 seconds based on network, and a load of location access of one request per 3 seconds based on GPS. Fig 10 shows the rate of battery depletion for each load with SmartMask running and SmartMask not running. For a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2016.2604383, IEEE Transactions on Dependable and Secure Computing

high location access rate(one request per 3 seconds), the battery depletion rate with SmartMask running is very close to the case without SmartMask running. So it is demonstrated that energy consuming of SmartMask won't affect usability.

6 Limitation

For the sake of our inability to access a large scale dataset in mobile social networks, we recruit volunteers to perform the real-world experiments to collect their shared locations, which contribute to *Dataset I*. To have a larger dataset, we leverage Wi-Fi access data on a university campus to simulate the location sharing events which involve 22,843 users and 98 locations. While this limitation may result in the bias of our dataset, we argue that it does not invalidate our approach, or privacy inference through shared locations. Our study is based on the observation that the users sharing similar demographics usually have similar mobility traces, which has been partially validated by previous works under different contexts such as recommendation system via mobility trajectories. Therefore, our proposed approach can be applied to other datasets although the considered features or location semantics may have some differences. Our study confirms that the threat of leaking users' sensitive demographic information through the shared locations is realistic. As one of our future works, we will consider a more resourceful adversary which can collect a large scale shared locations in mobile social networks to have a better understanding on the impact of the different users on privacy leakage arising from location sharing.

7 Related Work

A representative work to validate the geosocial mobility traces is comparing Foursquare check-in data with users' GPS data [15]. However, the previous work only considers the coverage rate. In this study, we consider diversified MSNs and introduce both of coverage rate and relative entropy to measure the distance of the inferred and real mobility pattern. There are quite a few studies addressing location privacy issues in MSNs [5], [6], [7], [8]. For example, our previous work [7] pointed out that it is possible for the adversaries to launch a long-term tracking towards a mobile user as long as she uses *Who's nearby?* function in Wechat, Momo and Skout. However, little attention has been paid to further privacy leaking issues arising from location sharing in MSNs.

There are many other works which study how to infer the victim's trajectory and further re-identify his other private information [9], [10], [11], [31], [32] . Different from the existing works, which are based on users' real mobility trace, this work aims to infer the sensitive information of the users from their shared locations, which is only the partial view of the users' real mobility pattern. In traditional online social networks, there are also some research works on inferring users' hidden attribute based on their interest [26]. To this best of our knowledge, our work is the first work to infer users' demographics based on the users' shared locations.

Location privacy protection in location-based services is a long-standing topic [16], [30], [33], [34], [35]. The most popular approach to achieve location privacy in LBS is utilizing obfuscation techniques to coarse the spatial or temporal granularity of real locations [36], [37], [38], [39]. The service utility and the privacy protection are always a trade-off. Different from previous works, we propose a system level solution, which can provide different privacy levels to different locations based on an automatic location management system. SmartMask remains compatible with the existing obfuscation techniques and compliments existing solutions. LP-Guardian [40] is framework for location privacy protection for Android smartphone users and it leverage the solution of [30] to anonymize user's locations. SmartMask is complementary to LP-Guardian; it learns users' privacy preferences and automatically assigns different privacy levels to achieve the balance of the privacy and the utility.

Privacy-aware and context-based systems become hot topics recently [41], [42]. L. Li et al [41] proposed system uses a classifier to learn the owner's finger movement patterns to achieve continuous and unobservable re-authentication for smartphones. DeepDroid [42] extract the context information to enforce a fine-grained policy on Android devices. Different from previous works, SmartMask is a system-level privacy protection framework leverage location contexts to achieve context-driven privacy management and fine-grained location privacy control.

8 Conclusion

The pervasiveness of location sharing in MSNs raises increasing privacy concerns. In this work, we have quantitatively evaluated the similarity of the shared locations and real locations based on the real-world collected datasets. Our quantitative evaluations indicate that although direct location sharing and indirect location sharing only reveal 16% and 33% of the users real points of interest (POIs), the attacker can exploit the similarity of the traces among different users to infer their age, occupation, living place, gender and education level at the successful rate of 69.2%, 53.8%, 54.5%, 73% and 76%. We then proposed SmartMask, a system-level solution to thwart location privacy breaches without significantly reducing the service quality. SmartMask can automatically learn and generate privacy levels of locations based on location contexts. As a general platform, SmartMask can incorporate with other advanced obfuscation techniques to resolve a wider range of location privacy issues.

Appendix
Simulation under Data Sharing Rates

In order to well justify the utilizing of dataset II, we define a metric "data sharing rate" as the possibility that a user shares his location in MSNs and tune this metric to show the simulation is reasonable. The precision, recall and F1-score of inferring demographics of users are shown in Table. 4.

TABLE 4: Justification of Utilizing Dataset II

Data sharing rate	Demographic	Precision	Recall	F1-score
20%	gender	0.62	0.61	0.61
	education	0.59	0.53	0.54
40%	gender	0.63	0.62	0.62
	education	0.63	0.59	0.59
60%	gender	0.65	0.62	0.63
	education	0.67	0.64	0.64
80%	gender	0.65	0.62	0.63
	education	0.70	0.68	0.69
100%	gender	0.67	0.64	0.65
	education	0.76	0.73	0.74

REFERENCES

[1] Craig Smith, "How Many People Use the Top Social Media, Apps & Services?" http://expandedramblings.com/index.php/resource-how-many-people-use-the-top-social-media/.

[2] K. P. Tang, J. Lin, J. I. Hong, D. P. Siewiorek, N. Sadeh. "Rethinking Location Sharing: Exploring the Implications of Social-Driven vs. Purpose-Driven Location Sharing," In *Proceedings of UbiComp*, ACM, 2010, pp. 85-94.

[3] Tracey Xiang, http://technode.com

[4] Apk Apps, "Skout 3.4.3 Apk Meet, Chat, Friend," http://www.apk4.net/applications/social-applications/skout-3-4-3-apk-meet-chat-friend

[5] H. Feng, Huan and K. G. Shin. "POSTER: Positioning Attack on Proximity-Based People Discovery," In *Proceedings of CCS*, ACM, 2014, pp. 1427-1429.

[6] Ding, Y., Peddinti, S. T., & Ross, K. W. "Stalking Beijing from Timbuktu: A Generic Measurement Approach for Exploiting Location-Based Social Discovery." In *Proceedings of Workshop on Security and Privacy in Smartphones & Mobile Devices*, ACM, 2015, pp. 75-80.

[7] M. Li, H. Zhu, Z. Gao, S. Chen, K. Ren, L. Yu, and S. Hu. "All Your Location are Belong to Us: Breaking Mobile Social Networks for Automated User Location Tracking," In *Proceedings of MobiHoc*, ACM, 2014, pp. 43-52.

[8] I. Polakis, G. Argyros, T. Petsios, S. Sivakorn and A. D. Keromytis, "Where's Wally?: Precise User Discovery Attacks in Location Proximity Services," In *Proceedings of CCS'15*, ACM, 2015, pp. 817-828.

[9] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing and X. Shen. "Location Privacy Preservation in Collaborative Spectrum Sensing," In *Proceedings of INFOCOM*, IEEE, 2012, pp. 729-737.

[10] M. Srivatsa and M. Hicks. "Deanonymizing mobility traces: Using social network as a side-channel," In *Proceedings of CCS*, ACM, 2012, pp. 628-637.

[11] J. Shao, R. Lu, and X. Lin, "FINE: A Fine-Grained Privacy-Preserving Location-based Service Framework for Mobile Devices," In *Proceedings of INFOCOM*, IEEE, 2014, pp. 244-252.

[12] Tiwari, Mohit, et al. "Context-centric security," In *Proceedings of USENIX conference on Hot Topics in Security*, USENIX, 2012, pp. 9-9.

[13] Landau, S. "Control use of data to protect privacy," In *Science*, vol. 347, no. 6221, pp. 504-406, 2015.

[14] Acquisti, A., Brandimarte, L., & Loewenstein, G. "Privacy and human behavior in the age of information," In *Science*, vol. 347, no. 6621, pp. 509-514, 2015.

[15] Z. Zhang, L. Zhou, X. Zhao, G. Wang, Y. Su, M. Metzger, H. Zheng, and B. Y. Zhao. "On the Validity of Geosocial Mobility Traces," In *Proceedings of HotNets*, ACM, 2013, pp. 11.

[16] R. Shokri, G. Theodorakopoulos, J. Le Boudec, and J. Hubaux. "Quantifying location privacy," In *Symposium on Security and Privacy*, IEEE, 2011, pp. 247-262.

[17] X. Chen, J. Pang and R. Xue. "Constructing and comparing user mobility profiles for location-based services," In *Proceedings of SAC*, ACM, 2013, pp. 261-266.

[18] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. "Identification via location-profiling in gsm networks," In *Proccedings of WPES*, ACM, 2008, pp. 23-32.

[19] H. Zang and J. Bolot. "Anonymization of location data does not work: A large-scale measurement study," In *Proccedings of MobiCom*, ACM, 2011, pp.145-156.

[20] Thomas M. Cover, Joy A. Thomas. "Elements of Information Theory," John Wiley & Sons, 1991.

[21] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. "Limits of predictability in human mobility," In *Science*, vol. 327, no. 5968, pp. 1018-1021, 2010.

[22] M. Ye, D. Shou, W.-C. Lee, P. Yin and K. Janowicz, "On the Semantic Annotation of Places in Location-Based Social Networks," In *Proceedings of KDD*, ACM, 2011, pp. 520-528.

[23] Y. Chon, Nicholas D. Lane, F. Li, H. Cha, F. Zhao, "Automatically Characterizing Places with Opportunistic CrowdSensing using Smartphones," In *Proceedings of UbiComp*, ACM, 2012, pp. 481-490.

[24] Gusfield, Dan, "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology," USA: Cambridge University Press, 1997.

[25] Q. Li, Y. Zheng, X. Xie, Y, Chen, W, Liu and W Ma. "Mining user similarity based on location history," In *Proceedings of GIS*, ACM, 2008, pp. 34.

[26] A. Chaabane, G. Acs, M. A. Kaafar, "You Are What You Like! Information Leakage Through Users' Interests," In *Proceedings of NDSS*, 2012.

[27] scikit-learn: Machine learning in python. http://scikit-learn.org/stable/.

[28] Loh, W. Y. "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2011.

[29] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," In *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 13-27, 2011.

[30] N. E. Bordenabe, K. Chatzikokolakis. C. Palamidessi. "Optimal geo-indistinguishable mechanisms for location privacy," In *Proceedings of CCS*, ACM, 2014, pp. 251-262.

[31] K. Zhang, X. Liang, R. Lu, K. Yang, and X. Shen, "Exploiting Mobile Social Behaviors for Sybil Detection," in *Proceedings of INFOCOM*, IEEE 2015, pp. 271-279.

[32] K. Zhang, X. Liang, X. Shen, and R. Lu, "Exploiting multimedia services in mobile social networks from security and privacy perspectives," *Communications Magazine, IEEE*, vol.52, no.3, pp.58-65, Mar. 2014.

[33] A. Narayanan, N. Thiagarajan, M. Hamburg, M. Lakhani, and D. Boneh. "Location privacy via private proximity testing," In *NDSS*, 2011.

[34] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux and J.-Y. Le Boudec. "Protecting Location Privacy: Optimal Strategy against Localization Attacks," In *Proccedings of CCS*, ACM, 2012, pp. 617-627.

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

[36] M. Gruteser and D. Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking," In *Proceedings of MobiSys*, ACM, 2003, pp. 31-42.

[37] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. "Preserving privacy in gps traces via uncertainty-aware path cloaking," In *Proceedings of CCS*. ACM, 2007, pp. 161-171.

[38] A. R. Beresford, A. Rice, N. Skehin and R. Sohan. "Mockdroid: Trading privacy for application functionality on smartphones," In *Proceedings of HotMobile*, ACM, 2011, pp. 49-54.

[39] K. Micinski, P.Phelps and J.S.Foster. "An Empirical Study of Location Truncation on Android," In *Weather*, vol. 2, pp. 21, 2013.

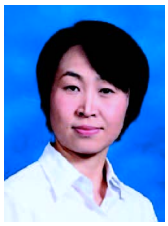
[40] K. Fawaz and K. G. Shin. "Location Privacy Protection for Smartphone Users," In *Proceedings of CCS*, ACM, 2014, pp. 239-250.

[41] L. Li, X. Zhao, G. Xue. "Unobservable Re-authentication for Smartphones," In *NDSS*, 2013.

[42] X. Wang, K. Sun, Y. Wang, J. Jing. "DeepDroid: Dynamically Enforcing Enterprise Policy on Android Devices," In *NDSS*, 2015.

Huaxin Li is a graduate student working towards his M.Sc. degree in Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received the B.Sc. degree in Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2011. His research interests include social networks privacy, smartphone security, network security and privacy, and machine learning.

Haojin Zhu is currently an Associate Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc.(2005) degree from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. His current research interests include network security and data privacy. He published 29 international journal papers, including IEEE Trans. On Parallel and Distributed Systems, IEEE Trans. on Wireless Communication, IEEE Trans. on Vehicular Technology, IEEE Wireless Communications, IEEE Communications, and 50 international conference papers, including ACM MOBICOM, ACM MOBIHOC, IEEE INFOCOM, IEEE ICDCS, IEEE GLOBECOM, IEEE ICC, IEEE WCNC. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award (2014) due to his contribution to wireless network security and privacy, Distinguished Member of the IEEE INFOCOM 2015 Technical Program Committee, Outstanding Youth Post Expert Award for Shanghai Jiao Tong University, SMCYoung Research Award of Shanghai Jiao Tong University. He was a corecipient of best paper awards of IEEE ICC 2007 and Chinacom 2008. He serves as the Associate/Guest Editor of IEEE Internet of Things Journal, IEEE Wireless Communications, IEEE Network, and Peer-to-Peer Networking and Applications.



Suguo Du received the BSc degree in Applied Mathematics from Ocean University of Qingdao, China, in 1993, the MSc degree in Mathematics from Nanyang Technological University, Singapore, in 1998, and the PhD degree in Control Theory and Applications Centre from Coventry University, U.K., in 2002. She is currently an Associate Professor of Management Science Department in Antai College of Economics & Management, Shanghai Jiao Tong University, China. Her current research interests include Risk and Reliability Assessment, Fault Tree Analysis using Binary Decision Diagrams, Fault Detection for nonlinear system and Wireless Network Security Management.

Xiaohui Liang received the B.Sc. degree in Computer Science and Engineering and the M.Sc. degree in Computer Software and Theory from Shanghai Jiao Tong University (SJTU), China, in 2006 and 2009, respectively. He is currently working toward a Ph.D. degree in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include applied cryptography, and security and privacy issues for e-healthcare system, cloud computing, mobile social networks, and smart grid.

Xuemin (Sherman) Shen (IEEE M'97-SM'02-F09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, Infocom'14, IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07 and QShine'06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.