Deanonymizing Mobility Traces With Co-Location Information

Youssef Khazbak  
Department of Computer Science and Engineering  
The Pennsylvania State University  
University Park, Pennsylvania 16802  
ymk111@cse.psu.edu

Guohong Cao  
Department of Computer Science and Engineering  
The Pennsylvania State University  
University Park, Pennsylvania 16802  
gcao@cse.psu.edu

Abstract—Mobility traces have been widely used in the design and evaluation of mobile networks. To mitigate the privacy threat of publishing mobility traces, the traces are often anonymized and obfuscated. However, even with anonymization and obfuscation techniques, traces can still be deanonymized by exploiting some side information such as users’ co-location. With online social networks, mobile users increasingly report their co-locations with other users. For example, a user may report being with friends at a restaurant for lunch or dinner, and hence his friends’ location information can be inferred. To find out whether co-location information can be exploited to identify a user and reveal his behavior from a set of mobility traces, we use a dataset from Twitter and Swarm to illustrate how an adversary can gather side information consisting of users’ location and co-location. Based on the collected information, the adversary can run a simple yet effective location inference attack. We generalize this attack, formulate the identity inference problem, and develop inference attacks, under different observed side information, that deem effective in identifying the users. We perform comprehensive experimental analysis based on real datasets for taxi cabs and buses. The evaluation results show that co-location information can be used to significantly improve the accuracy of the identity inference attack.

I. INTRODUCTION

Mobility traces have been widely used to address many practical problems in mobile networks. For instance, researchers exploit mobility traces to develop effective resource allocation methods [1], design efficient protocols for data dissemination [2] [3], improve network connectivity and urban planning [4], etc. However, releasing a user’s mobility trace poses great privacy threats, as it can provide information about the user behavior and interests. Therefore, published mobility traces are often anonymized and obfuscated. Traces can be anonymized by replacing the user’s true identity with a pseudonym that can not be correlated to the user. Location obfuscation can be achieved by replacing the actual location by an obfuscated location (i.e., adding noise), or by reducing the granularity of the location. After anonymizing the traces, the location information of each trace can be obfuscated in order to prevent the adversary from associating locations such as the user’s home location with his true identity [5].

Even with anonymization and obfuscation techniques, traces can still be deanonymized by exploiting some side information. For example, Ma et. al [6] found that some known user locations in public spaces can be exploited as side information, which can be inferred directly from physical meetings, or indirectly from conversations, web logs, or online social networks. Srivatsa et. al [7] found that location traces can be deanonymized given an easily obtained social relationship. A user may be identified by those she meets: meetings between anonymized users in a set of traces can be structurally correlated with their social relationships, thereby identifying anonymized users.

In this paper, we focus on another kind of side information, users’ co-locations. With online social networks such as Twitter, Foursquare, and Swarm, mobile users increasingly report their co-locations with other users, besides disclosing their own locations. Co-locations when combined with location information can be exploited to improve the inference of users’ locations. For example, a user may report being with friends at a restaurant for lunch or dinner, and hence his friends’ location information can be inferred. Then, a natural question will be: can we use location and co-location information to identify a user and reveal his behavior from a set of mobility traces? To answer this question, we first use a dataset from Twitter and Swarm to illustrate how an adversary can gather side information consisting of users’ location and co-location. Based on the collected information, the adversary can run a simple yet effective location inference attack. We generalize this attack, formulate the identity inference problem, and develop inference attacks, under different observed side information, that deem effective in identifying one or more users. Once users are identified, the adversary can reveal their movements and infer sensitive information such as where they live and work. Finally, we quantify the loss of user privacy under different system parameters, such as the side information obtained, the observed trace time, the number of traces considered, and the obfuscation function.

Our main contributions can be summarized as follows:

- We use a collected dataset from Twitter and Swarm to demonstrate that co-location information can be exploited to improve the inference of user’s locations.
- We identify the identity inference problem and propose inference attacks based on different observed side information. In the first attack, the adversary has access
to user’s location and co-location information over the observed trace time, and in the second attack, adding to the above knowledge, the adversary has access to past location traces to construct user’s mobility profile.

- We perform comprehensive experimental analysis based on real datasets for taxi cabs and buses. The evaluation results show that co-location information can be used to significantly improve the accuracy of the identity inference attack.

The rest of the paper is organized as follows. In Section II, we present the preliminaries. Section III presents the problem definition. The proposed inference attack is described in Section IV. Section V presents performance evaluations. Section VI describes the related work and Section VII concludes the paper.

II. PRELIMINARIES

In this section, we first present the motivation of the work, and then use a real example from Twitter and Swarm, to illustrate how an attacker can gather users’ co-location information and run a simple yet effective location inference attack.

A. Motivation

With some side information such as location and co-location information of a user, the attacker can link and identify users from a set of anonymous traces. As shown in Figure 1 (a), suppose we have two anonymous mobility traces: $L_a$ and $L_b$, each containing three events. Trace $L_a$ contains three events occurring at locations $L_1, L_2, L_3$ at time $t_1, t_2,$ and $t_3$ respectively. Assume the adversary learned that Alice visited locations $L_1$ at time $t_1$ and visited $L_2$ at time $t_2$ based on some side information. Then, Alice can be identified, since only trace $L_a$ contains locations $L_1$ and $L_2$ at time $t_1$ and $t_2$, and no other trace contains these locations.

In Figure 1 (b), a trace $L_c$ is added. As a result, both trace $L_a$ and trace $L_c$ contain locations $L_1$ and $L_3$ at the same time. The adversary knows that Alice appears in trace $L_a$ or $L_c$, but does not know which one. Suppose the adversary has some side information, e.g., he knows that Alice is co-located with Bob at time $t_3$. Then the adversary can infer that Alice’s trace is $L_a$, since only traces $L_a$ and $L_b$ contain the same location $L_3$ at time $t_3$. In the next subsections, we will illustrate how the adversary obtains such side information.

B. Dataset Collection

The dataset is collected from the public Twitter feed using Twitter REST API [8]. Twitter allows users to post geo-tagged tweets that include location information, however, from such tweets it is not straightforward to extract co-location information. That is because the current tweet’s format does not differentiate users reporting co-locations with friends from users mentioning their friends in their posts. To obtain co-location information, we consider another online social network, Swarm, that includes “I’m here with” feature which enables users to post check-ins along with the person they are with. Twitter offers its users the option of linking their Twitter accounts with their Swarm accounts, so that whenever
a user uses Swarm for check-ins, a short URL to the Swarm check-in, along with other information, are posted on the user’s Twitter timeline. We gathered these tweets which contain Swarm check-ins. Each check-in includes the user name, the location, the timestamp, and the co-location information. Co-location information is obtained by parsing the Swarm URL from the tweet. Then from the obtained Swarm check-in data, co-location information can be extracted. For example, Figure 2 shows how a user reports his location and co-location information on Twitter and Swarm. In Figure 2 (a), the user Lewis checked in at restaurant Diamond Jim’s with Blanca on January 1, 2016. As the user sets swarm check-ins to be automatically posted on Twitter, the check-ins will appear on Twitter as shown in Figure 2 (b).

The collected dataset contains 41,017 users and 1,231,023 check-ins from December 2015 to May 2016. We run our analysis on users around New York and California where we found the largest number of users using check-ins and co-location reporting features. The resulted dataset contains around 700 users when only considering those reported at least 5 co-locations. The Check-in locations are either actual locations or obfuscated locations. Location obfuscation is achieved through spatial cloaking, by generalizing a given location to a rectangular area at the next neighborhood town, city, state, or country level.

C. Knowledge of the Adversary

The adversary is capable of observing and collecting various user information such as location and co-location information, from online social networks. Although the location information may be obfuscated or noisy due to user privacy requirements or imprecise observations by the adversary, such information can be refined using simple location inferences, based on the fact that co-locations form dependencies between users’ locations. For instance, as shown in Figure 3, two users Carrancho and Franklin report their locations at a given time through Twitter. Carrancho reported being co-located with Franklin at George & Seans’s Place, a private place without giving the exact location (i.e., a cloaked region as shown in Figure 3 (c)). Similarly, as shown in Figure 3 (d), Franklin reported being at the Camelot, with the exact location. If the adversary has collected both information (i.e., location and co-location information), he will be able to infer Carrancho’s exact location. In general, co-located users may report obfuscated locations with different cloaking levels depends on their privacy requirement. If the adversary has collected users location and co-location information, he will be able to infer that the actual possible locations are at the intersection of their overlapped cloaking regions.

By using the aforementioned inference strategy, more user location information can be obtained. Even though some users do not provide their location information, based on the co-location information, the locations of these users can still be identified. For example, consider a user $u$, let $L_u = \{L_1, ..., L_n\}$ represent his visited locations obtained from his Twitter posts. Let $L_u^c$ denote $u$’s visited locations inferred by using co-location information, for instance, one of his friends may report being with him at a restaurant for lunch and hence his location can be inferred. As shown in Figure 4, although user $u$ did not report any location information ($L_u = \{\emptyset\}$), his visited locations are identified by using the co-location information as shown by $L_u \cup L_u^c$.

To quantify the effect of exploiting co-location information, we introduce a new metric called $P$, which is the percentage increase of the revealed visited locations by using co-location information. For instance, for a user $u$, it is defined as follows.

$$P_u = \frac{\left| \{L_i : L_i \in L_u^c \text{ and } L_i \notin L_u\} \right|}{|L_u \cup L_u^c|}.$$  \hspace{1cm} (1)

Figure 5 shows the distribution function of $P$ based on our collected trace. By using the co-location information, more visited locations can be identified. For around 50% of the users, more than 20% of the visited locations are revealed from the co-location information. Thus, co-location information can
be exploited to improve the inference of users’ locations and identify more location information.

III. PROBLEM DEFINITION

A. Problem Model

We consider a set of mobility traces, each is represented by a sequence of time-stamped locations. The traces include users or vehicles movements in a geographical area within a period of time. To preserve user privacy, the user’s identity is anonymized by replacing the user true identity with a pseudonym. Moreover, the locations in a trace are obfuscated by reducing the spatial granularity. The adversary, with access to the anonymized traces and some side information of several users, aims to identify the complete trace of some users, referred to as the target users.

With online social networks such as Twitter, Foursquare, and Swarm, the adversary can collect side information of the target users. The side information may provide: a set of locations over the trace time, a set of reported co-locations with other users, and/or past location traces which can be used to build mobility profiles. There are also other practical ways to obtain side information. For example, other data formats such as pictures can be obtained from online social networks, which may contain the location and time of the event in EXIF data format. Then, the co-location information can be obtained by applying face recognition techniques to identify the users in the picture. Another example is through direct encounters, in which the adversary may meet the target users, and use Bluetooth equipped device to report the presence of other co-located Bluetooth neighboring devices. In the rest of the paper, we will show how an adversary can exploit such side information to launch more powerful identity inference attacks.

B. Threat Model

The adversary is an observer who has access to published mobility traces. Given some side information about target users, the adversary’s objective is to identify one or more target users and reveal their movements and habits; i.e., carrying out identity inference attack. We consider two different attacks based on the obtained side information of the target users. In the first attack, the adversary has access to the users’ location and co-location information over the trace time. In the second attack, in addition to the above information, the adversary has access to the users’ past location traces which can be used to build users’ mobility profiles.

IV. PROPOSED IDENTITY INFERENCE ATTACKS

In this section, we first introduce some notations, and then present identity inference attacks.

A. Notation and Terminology

We consider n observed mobility traces, associated with a set \( \mathcal{U} \) of n users. Each trace includes a number of sampled locations and the sampling time. Then, user \( u \)'s trace is denoted by \( L_u = \{L_u(1), ..., L_u(T)\} \), where \( L_u(t) \) represents \( u \)’s actual location at time \( t \in \{1,..,T\} \). The trace will be anonymized by replacing the user identity with a pseudonym, which is denoted by \( L_{\sigma(u)} = \{L_{\sigma(u)}(1), ..., L_{\sigma(u)}(T)\} \), where \( \sigma(.) \) is a random function that generates a random identifier for each user. To further improve privacy, location obfuscation can be applied. Then the obfuscated trace is denoted as \( L'_{\sigma(u)} = \{L'_{\sigma(u)}(1), ..., L'_{\sigma(u)}(T)\} \). The location obfuscation replaces the actual location of a user with another location (i.e., adding noise to the actual location). Formally, let \( f_u(r,r') \) denote the probability that the obfuscation mechanism used by \( u \) obfuscates location \( r \) to \( r' \); i.e., \( p(L'_{\sigma(u)}(t) = r' | L_{\sigma(u)}(t) = r) \). The obfuscation function of user \( u \) is denoted by \( f_u \). Similarly, we can define the obfuscation function of the whole trace as \( f \).

The adversary has access to the observed location and co-location information of one or several target users. For a user \( u \), the location information, which may be noisy or obfuscated, is denoted by \( O_u = \{O_u(t_1), ..., O_u(t_k)\} \), where \( \{t_1, ..., t_k\} \subset \{1,..,T\} \). Let \( C_{u,v}(t) \) denote whether two users \( u \) and \( v \) are co-located at time \( t \), more specifically:

\[
C_{u,v}(t) = \begin{cases} 
1, & \text{if } L_u(t) = L_v(t) \\
0, & \text{Otherwise}
\end{cases}
\]


Let $c_{u,v}$ denote the co-location relationship between user $u$ and user $v$, which is a vector; i.e., $c_{u,v} = [C_{u,v}(1), \ldots, C_{u,v}(T)]$. Let $c_u(t)$ denote all co-locations of user $u$ at time instant $t$, which is a vector; i.e., $c_u(t) = [C_{u,v_1}(t), \ldots, C_{u,v_n}(t)]$, where $\{v_1, \ldots, v_n\} \subseteq \mathcal{U}$. Let $C_u$ denote all observed co-locations of user $u$. Then, we have $C_u = [c_u(1), \ldots, c_u(T)]$.

**TABLE I**

**SUMMARY OF NOTATION**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Number of observed mobility traces</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of geographical regions</td>
</tr>
<tr>
<td>$\mathcal{R}$</td>
<td>Set of regions that partition the geographical area</td>
</tr>
<tr>
<td>$\mathcal{U}$</td>
<td>Set of users</td>
</tr>
<tr>
<td>$T$</td>
<td>Observed time period</td>
</tr>
<tr>
<td>$\mathcal{L}_u$</td>
<td>Actual mobility trace of user $u$</td>
</tr>
<tr>
<td>$\mathcal{L}'_u$</td>
<td>Obsfuscated mobility trace of user $u$</td>
</tr>
<tr>
<td>$\mathcal{L}'(\sigma(u))$</td>
<td>Obsfuscated mobility trace of a user with pseudonym $\sigma(u)$</td>
</tr>
<tr>
<td>$O_u$</td>
<td>Set of observed locations of user $u$</td>
</tr>
<tr>
<td>$c_{u,v}$</td>
<td>Observed co-location relationship between user $u$ and user $v$</td>
</tr>
<tr>
<td>$C_u$</td>
<td>All observed co-locations of user $u$</td>
</tr>
<tr>
<td>$P_u$</td>
<td>Mobility profile of user $u$</td>
</tr>
<tr>
<td>$\Pi_u$</td>
<td>Stationary distribution of user $u$</td>
</tr>
<tr>
<td>$f$</td>
<td>Obfuscation function of the mobility trace</td>
</tr>
<tr>
<td>$f_u$</td>
<td>Obfuscation function of user $u$</td>
</tr>
</tbody>
</table>

**B. Inference attacks of the adversary**

1) **Inference attack given user’s locations and co-locations ($O_u, C_u$):** In this attack, the adversary has access to the user’s location and co-location information over the observed trace time $T$. The adversary’s objective is to find the trace among all observed traces that best matches the given information about the user. To formulate the objective, we use maximum likelihood estimator (MLE) to find the $\mathcal{L}'_x$ that maximizes the likelihood function. The distributions in this section are represented by probability densities. Accordingly, using the Bayes’ Rule, the formulation is as follows:

$$p(\mathcal{L}'_x|O_u, C_u, \mathcal{L}') = \frac{p(O_u, C_u, \mathcal{L}'(\mathcal{L}'_x)) \cdot p(\mathcal{L}'_x)}{p(O_u, C_u, \mathcal{L}')},$$

where $\mathcal{L}' = \{L'_{y_t}|y_t=1, \ldots, y_T \neq x\}$. With MLE, we consider the term $\frac{p(\mathcal{L}'_x)}{p(O_u, C_u, \mathcal{L}')} = 1$ as a constant, and our goal is to find the value of $L'_{x_t}$ that maximizes the likelihood, i.e., $p(O_u, C_u, \mathcal{L}'(\mathcal{L}'_x))$. Formally, the objective is to find $x$ which maximizes the following likelihood:

$$\arg \max_x p(O_u, C_u, \mathcal{L}'(\mathcal{L}'_x)).$$

Since $O_u$ and $C_u$ are independent, the likelihood would be:

$$\arg \max_x p(O_u|\mathcal{L}'_x)p(C_u|\mathcal{L}'(\mathcal{L}'_x)).$$

The observed locations, at different time instants, are obfuscated independently of each other. Hence, the objective becomes as follows:

$$\arg \max_x \left( \prod_{i=1}^{T} p(O_u(t_i)|L'_{x_t}(t_i)) \cdot \prod_{v \in \mathcal{U}} p(c_{u,v}|\mathcal{L}'(\mathcal{L}'_x)) \right),$$

where $x = 1, \ldots, n$.

Equation (5) can be simplified based on the obfuscation functions (i.e., $f$ and $f_u$). For example, similar to the literature [6], we consider the obfuscation functions to be Gaussian and uniform distributions.

1) Gaussian distribution with $N(0, \sigma^2)$: in this case, the problem can be solved using the method of least squares, which minimizes the sum of the squares of the distance between two observed locations. To use least squares, we need to find the $\mathcal{L}'_x$ that minimizes (i) the distance between each observed location $O_u(t_i)$ and the equivalent location $L'_{x_t}(t_i)$ in the trace $\mathcal{L}'_x$, and (ii) the distance between the locations, $L'_{x_t}(t)$ in the trace $\mathcal{L}'_x$ and $L'_{y_t}(t)$ in the trace $\mathcal{L}'_y$, when co-locations are observed; i.e. $C_{u,v}(t) = 1$. Hence, the optimization function would be:

$$\arg \min_x \left( \sum_{i=1}^{T} (O_u(t_i) - L'_{x_t}(t_i))^2 + \sum_{v \in \mathcal{U}} \min_{y \neq x} \sum_{t=1}^{T} (L'_{x_t}(t) - L'_{y_t}(t))^2 \right).$$

2) Uniform distribution in $[-d, d]$: in this distribution, the obfuscated location is at a distance of at most $d$ from the actual location. Then we have

$$p(O_u(t_i)|L'_{x_t}(t_i)) = \begin{cases} 0, & \text{if } |O_u(t_i) - L'_{x_t}(t_i)| > d \\ \frac{1}{2d}, & \text{if } |O_u(t_i) - L'_{x_t}(t_i)| \leq d \end{cases}$$

Let $c_{u,v}^t$ denote the fact that two users $u$ and $v$ were co-located at time $t$ has been observed. We have the following:

$$p(c_{u,v}|L'_{x_t}(t)) = \prod_{y \neq x \in \mathcal{U}} p(c_{u,v}^t|L'_{y_t}(t), c_{u,v}^t|L'_{x_t}(t)),$$

where

$$p(c_{u,v}^t|L'_{y_t}(t), L'_{x_t}(t)) = \begin{cases} 0, & \text{if } |L'_{y_t}(t) - L'_{x_t}(t)| > d \\ \frac{1}{2d}, & \text{if } |L'_{y_t}(t) - L'_{x_t}(t)| \leq d \end{cases}$$

Then by using the calculated values for $p(O_u(t_i)|L'_{x_t}(t_i))$ and $p(c_{u,v}|L'_{x_t})$ in Equation 5, we can obtain the highest likelihood trace.

2) **Inference attack given user’s mobility profile, locations and co-locations ($P_u, O_u, C_u$):** Besides the side information obtained in the first attack, the adversary has access to the past location traces to construct user’s mobility profile. To build the mobility profile, we assume the geographical area is partitioned into $m$ regions represented by a set $\mathcal{R} = \{R_1, \ldots, R_m\}$. Then for a target user $u$, the mobility profile is modeled as a
A homogeneous first order Markov Chain on \( R \), represented by a Markov transition matrix \((P_u)\). Each matrix entry, denoted by \( P_u(R_i, R_j) \), represents the probability that user \( u \) moves from region \( R_i \) to \( R_j \) during one time instant. Let \( \Pi_u \) denote the stationary distribution of \( P_u \). Each matrix entry, denoted by \( \Pi_u(R_i) \), represents the probability that user \( u \) is in region \( R_i \).

The inference problem translates to solving a first order Hidden Markov model (HMM). In the HMM, we have known states and hidden states. The known states include: the obfuscated trace, observed locations, and co-locations, while the hidden states are the actual locations. Hence, the problem reduces to a HMM evaluation problem, where we have a complete HMM, with transition and obfuscation probabilities. The goal is to find the trace that best matches the HMM for the target user. This is achievable by calculating the likelihood of each trace given the HMM. Then, we identify the target user's trace as the trace that gives the highest likelihood.

To solve the HMM problem, we use a well known inference algorithm called the iterative forward algorithm [9]. This algorithm computes recursively the probability that the model produces a sequence of observed locations and co-locations by defining a forward variable \( \alpha_t(r) \), which represents the joint probability of the following observed information: obfuscated trace, locations, co-locations from the beginning of the trace up to time \( t \), and that the actual location at time \( t \) is \( r \). Formally,

\[
\alpha_t(r) = \begin{cases} 
P(L_x^1(1), \ldots, L_x^t(t), \{O_u(t_i)\}_{i=1}^t, c_u(1), \ldots, c_u(t), 
\{L'_y(1), \ldots, L'_y(t)\}_{y=1..n,y \neq x}, L_x(t) = r | P_u), 
\end{cases}
\]

(7)

where \( t_i \) denotes the time sample when location is observed.

The inference objective is to find \( x \) which maximizes the following likelihood:

\[
\arg \max_x \ p(L_x^1, O_u, C | P_u),
\]

(8)

where \( x = 1, \ldots, n, C = \{C_u, L'\} \), and \( L' = \{L'_y(1), \ldots, L'_y(t)\}_{y=1..n,y \neq x} \).

The forward variables \( \alpha_t(r) \) are initialized using the stationary distribution \( \Pi_u(r) \). Let \( \beta_t(r) = \sum_{r' \in R} \alpha_t(r') P_u(r, r') \), then the forward variables can be defined recursively over \( t \) for all \( r = R_1, R_2, \ldots, R_m \) as follows:

\[
\alpha_t(r) = \begin{cases} 
\Pi_u(r), & \text{if } t = 0 \\
\sum_{r' \in R} \alpha_t(r') f(r, O_u(t), r'), & \text{if } t = t_i \\
\sum_{r' \in R} \alpha_t(r') f(r, O_u(t), r'), & \text{if } t \neq t_i \\
\sum_{r' \in R} \alpha_t(r') f(r, L_x^t(t), r'), & \text{if } t = t_i \\
\sum_{r' \in R} \alpha_t(r') f(r, L_x^t(t), r'), & \text{if } t \neq t_i \\
\end{cases}
\]

where

\[
p(L'_y, L'_y | L_x^t(t)) = \max_{y=1..n,y \neq x} \ f(r, L'_y(t)). f(r, L'_y(t)).
\]

(9)

With the forward variables, we can calculate the likelihood of each trace \( L'_x \) as:

\[
p(L'_x, O_u, C | P_u) = \sum_{r=R_1}^{R_m} \alpha_T(r).
\]

(10)

Combining Equation 10 and Equation 8, we can obtain the highest likelihood trace.

We now calculate the computation complexity of the proposed attack. The complexity of computing the likelihood of all traces takes \( O(n^2 + m^2 + T) \) time. We have \( n \) traces, and for each trace, the number of time steps is \( T \). For each time step, we need to consider all \( m \) regions. For each region, the computation of the forward variable is obtained by summing over all \( m \) forward variables from the previous stage. Hence, for every time step, we have \( O(m^2) \) computations. When a co-location is observed at a time step, all \( n \) locations of the traces need to be checked to compute the probability that two users are co-located, with a computation complexity \( O(n) \).

V. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of the inference attacks. We first introduce our evaluation setup and then show the evaluation results.

A. Evaluation Setup

The evaluations are based on two datasets as shown in Table II. The Rome cabs dataset [10] compromises mobility traces of taxi cabs observed over 30 days in Rome, Italy. The Shanghai buses dataset [11] consists of mobility traces of buses collected over 6 days in Shanghai, China. Both datasets include time-stamped location traces collected using GPS device.

<table>
<thead>
<tr>
<th>TABLE II DATASETS STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Region</td>
</tr>
<tr>
<td>Number of Users</td>
</tr>
<tr>
<td>Minimum latitude</td>
</tr>
<tr>
<td>Minimum longitude</td>
</tr>
<tr>
<td>Maximum latitude</td>
</tr>
<tr>
<td>Maximum longitude</td>
</tr>
</tbody>
</table>

In the traces, the area is divided into square cells of size 0.01° in latitude and longitude. The GPS coordinate of each location sample can be mapped into the region (i.e., the cell) it falls into. The distance between any two cells given their latitude and longitude is calculated using the haversine formula as in [12]. In the trace, the time is divided into \( T \) time intervals. For each time interval, one location is sampled. Two users are co-located if they are located inside the same cell for at least 5 minutes. Figure 6 shows the regions where the traces are captured. The visited cells are represented by black cells, and other colors (yellow, orange, and red) represent cells where co-locations are observed.

We compare the proposed attacks with the attack in [6] which exploits location information as the side information, but does not consider co-locations. In our proposed attacks,
co-locations are used as another source of side information where \( c \) represents the number of observed co-locations for a target user. By changing the values of \( c \) (from 0 to 15), different amount of co-location informations are exploited for attacks. Note that \( c = 0 \) represents the attack in [6], which does not consider co-location and it is the baseline for comparison.

In each simulation, we first randomly choose a target user trace, and then try to identify his trace among all other traces, based on the collected side information. The results are averaged over 200 simulation runs. The performance of the inference attacks is evaluated based on a metric called Identification Accuracy, which measures the percentage of target users that are correctly identified by the adversary.

**B. Evaluation Results: Inference attack given \((O_u, C_u)\)**

We consider an adversary with access to the location and co-location information \((O_u, C_u)\) of the target user. In each simulation, the amount of side information represented by \( c \) is randomly picked from the selected target user’s trace. For instance, with \( c = 5 \), five co-locations are randomly picked among all the observed co-locations in the target user’s trace. The number of sampled locations, denoted by \( k \), is chosen similarly. The locations in the traces and the user’s observed locations are by default obfuscated using Gaussian distribution with \( \mu = 0 \) and \( \sigma = 5 \) if not mentioned otherwise.

1) **Effects of the number of observed locations \( k \) and co-locations \( c \):** Figure 7 shows the identification accuracy as a function of \( k \) and \( c \). As can be seen, when the observed locations are combined with co-locations, the identification accuracy improves. That is, with more available location (co-location) information, the attacker is more likely to identify the target user’s trace. Moreover, the identification accuracy in the bus trace is lower than that in the cab trace. This is because bus movements have higher correlation than the cab movements, and then it is harder to differentiate these bus traces.

In Figure 8, we evaluate the effects of using an obfuscation function of uniform distribution with \( d = 10 \) on the performance of the inference attack. Since this obfuscation distribution adds more noises on average to the locations compared to the Gaussian distribution with \( \mu = 0 \) and \( \sigma = 5 \), the users’ location privacy is increased. As shown in Figure 8 (a), even with larger noises through obfuscation, the adversary can still identify 40% of the users using 20 location samples, and up to 80% if 15 co-locations are observed.

C. **Evaluation Results: Inference attack given \((P_u, O_u, C_u)\)**

We consider an adversary who has access to the target user’s mobility profile. In order to build the mobility profile in our analysis, we divide the user’s trace into two parts: the first part
represents the past location trace, and the other part represents the observed trace. With the past location trace, we construct the mobility transition matrix $P_u$ by counting the transitions from one cell to another within one time instant. For the cab traces, we build the transition matrix based on traces over 20 days, and the remaining 10 days are used as the observed traces. The observed traces include one location every hour (i.e., $T = 1 \times 24 \times 10 = 240$). While for the bus traces, 1 day is used to build the transition matrix, and 4 days are used as the observed traces. The observed traces include one location every half hour (i.e., $T = 2 \times 24 \times 4 = 192$). We select a region of $15 \times 15$ square cells where the largest number of observed locations and co-locations are found. The total number of cells within the considered region is $m = 225$. Finally, the locations are obfuscated using a uniform distribution function with $d = 1$, in which the actual cell is replaced by a cell chosen uniformly at random among cells at a distance of at most 1.

1) Effects of the number of observed locations $k$ and co-locations $c$: In Figure 9, we study the effect of $k$ and $c$ on the identification accuracy. Even without the knowledge of any locations (i.e., $k = 0$), the attack is effective when using co-locations (i.e., $c > 0$), with mobility profiles. As shown in Figure 9 (a), without knowing any locations or co-locations (i.e., $k = 0, c = 0$), the attacker can identify up to 17% of the users in the cab traces using only mobility profiles. Hence, the more accurate the constructed mobility profiles on presenting the users’ movements, the more effective the attack would be.

2) Effects of the number of traces $n$: As shown in Figure 10, when the number of traces is small, the adversary can identify the target user with high probability. For example, the adversary can identify 90% of the target users among a small set of traces $n = 10$ with only $k = 2$ locations. When the number of traces is large (i.e., $n = 200$), with the same amount of side information, the identification accuracy drops to 38%. This is because with more traces, it is harder to differentiate and identify the target user’s trace.

3) Effects of the observed time interval $T$: Figure 11 shows identification accuracy as a function of $T$. As the observation time increase, the attacker can obtain more side information such as observed locations and co-locations, and then launch better attacks. To see the effect of $T$, we create several strategies based on two parameters: the observed location rate ($r_s$) which represents the number of observed locations within a time interval, and the observed co-location rate ($r_c$) which represents the number of observed co-locations within a time interval. For example, $r_s = 2, r_c = 2$ means that the attacker observes two locations and two co-locations within each time window.

In Figure 11 (a), three approaches (“$r_s = 0, r_c = 0$”, “$r_s = 1, r_c = 1$”, “$r_s = 2, r_c = 2$”) are compared. As can be seen from the figure, as the time increases, the attacker can learn more information and then has better identification accuracy. Among the three approaches, since the attacker can learn more information in the “$r_s = 2, r_c = 2$” approach, its identification accuracy is the highest.

Comparing Figure 11 (a) to Figure 11 (b), we can see that the attacker can achieve higher identification accuracy for the bus trace, because more accurate mobility profiles can be generated from the bus traces due to its movement regularity.

VI. RELATED WORK

Many location protection mechanisms have been proposed in the literature [13]–[18]. To preserve privacy of the mobility traces, Chow et. al [13] presented techniques such as k-anonymity, mix zones, and dummy traces generation. With k-anonymity, user’s location can be hided inside a cloaked spatial region that contains other $k − 1$ users. A mix zone is another way to hide the user identity. Once a user enters a mix-zone with other users, pseudonyms are used so that users cannot be distinguished from each other when they exist the mix zone. Recently, some research such as [15], [17], [19]–[21], relies on differential privacy to ensure user privacy. Compared to other conventional approaches such as k-anonymity, differential privacy has a stronger theoretical framework and can provide stronger privacy guarantee.

Several recent work highlights the privacy vulnerability of anonymized traces by exploiting different forms of side information. Ma et. al [6] found that some known user locations in public spaces can be exploited as side information, which can be inferred directly from physical meetings, or indirectly from conversations, web blogs, or online social networks. Srivatsa et. al [7] found that location traces can be deanonymized given an easily obtained social relationship. A user may be identified by those she meets: meetings between anonymized users in a
set of traces can be structurally correlated with their social relationships, thereby identifying anonymized users. Jurgens [22] exploited social relationship to infer users’ locations. Sadilek et al. [23] presented probabilistic models based on social ties to infer users locations. However, none of them considers co-location information, which is the focus of this paper.

Besides these various side information that can threaten users’ privacy, a powerful adversary may leverage the fact that the privacy of individual users is bound to be affected by the decision of others. In [24], Biczok et al. define interdependent privacy in the context of online social networks where individual’s privacy may be affected by the action of others. In [25], Benjamin et al. showed that a user’s friends can share images contain location information and hence degrade the user location privacy. In [26], Olteanu et al. proposed attacks that use co-location information to find user’s location at a specific time, namely carry out localization attack. In their work, the user identity is known prior to the attack. Different from the existing work, we exploit co-location information to identify users in anonymized mobility traces and reveal the user’s full trajectory, namely carry out deanonymization attack.

VII. CONCLUSIONS

In this paper, we studied the identity inference problem, where the attackers can deanonymize mobility traces by exploiting the co-location information. With online social networks, mobile users increasingly report their co-locations with other users, and such information can be exploited to identify the user and reveal his behavior from a set of mobility traces. We used a dataset from Twitter and Swarm to illustrate how an adversary can gather side information consisting of users’ location and co-location. Based on the collected information, the adversary can run a simple yet effective location inference attack. We generalized this attack, formulated the identity inference problem, and developed inference attacks, under different observed side information, that deem effective in identifying the users. We performed comprehensive experimental analysis based on real datasets for taxi cabs and buses. The evaluation results show that co-location information can be used to significantly improve the identification accuracy of the proposed attacks.

REFERENCES