LA-UR-10-05378

Approved for public release;
distribution is unlimited.

Title: Privacy analysis of user association logs in a large-scale wireless LAN

Author(s): Keren Tan, Guanhua Yan, Jihwang Yeo, David Kotz

Intended for: IEEE Infocom 2011

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (7/06)

This page intentionally left blank.

Privacy analysis of user association logs in a large-scale wireless LAN

Keren Tan, Guanhua Yan†, Jihwang Yeo, David Kotz
Department of Computer Science, ISTS, Dartmouth College
†Information Sciences (CCS-3), Los Alamos National Laboratory

***Abstract*—User association logs collected from a large-scale wireless LAN record where and when a user has used the network. Such information plays an important role in wireless network research. One concern of sharing these data with other researchers, however, is that the logs pose potential privacy risks for the network users. Today, the common practice in sanitizing these data before releasing them to the public is to anonymize users' sensitive information, such as their devices' MAC addresses and their exact association locations. In this work, we aim to study whether such sanitization measures are sufficient to protect user privacy. By simulating an adversary's role, we propose a novel type of correlation attack in which the adversary uses the anonymized association log to build signatures against each user, and when combined with auxiliary information, such signatures can help to identify users within the anonymized log. Using a user association log that contains more than four thousand users and millions of association records, we demonstrate that this attack technique, under certain circumstances, is able to pinpoint the victim's identity exactly with a probability as high as 70%, or narrow it down to a set of 20 candidates with a probability close to 100%. We further evaluate the effectiveness of standard anonymization techniques, including generalization and perturbation, in mitigating correlation attacks; our experimental results reveal only limited success of these methods, suggesting that more thorough treatment is needed when anonymizing wireless user association logs before public release.**

I. Introduction

In many large-scale wireless local area networks (WLANs), user association logs keep a record of each association and disassociation event between users' wireless devices and the network's access points (APs). Such traces collected from production networks, when made available for research, play a critical role in understanding user activity patterns, analyzing network protocol dynamics, and evaluating the performance, reliability, and security of new network designs [42], [11], [40]. We, at Dartmouth College, have monitored a campus-wide WLAN for almost one decade and some of our collected traces have been made public through our CRAWDAD website [10]. These network traces have been extensively studied by the wireless research community and have been used in more than 100 research publications.

To preserve users' privacy, a trace publisher must *sanitize* the network traces before sharing them with the public. Although many network sanitization techniques have been proposed and developed, recent research has shown that these techniques provide limited protection against user (or host) re-identification attacks. Existing sanitization techniques usually deal with explicit sensitive fields in the dataset, such as IP/MAC addresses, port number, and TCP/UDP payloads, but ignore implicit information that can be potentially extracted and used to identify an anonymized user (or host). For an enterprise-wide network with thousands of users, privacy analysis on *wired* network traces has been widely studied to understand the severity of some potential trace-sharing risks [5], [33], [9]. However, similar research is scarce for enterprise-wide, large-scale *wireless* networks [45], [22]. As the edge of the Internet is increasingly becoming wireless, and because wireless networks have some unique characteristics, such as user mobility, it is important to evaluate privacy threats posed due to shared wireless network traces.

In this paper, we conduct privacy analysis on one of the simplest wireless network traces, a user association log, collected from a large-scale WLAN. Compared to other semantically rich wireless-network traces, we would hope the simplicity of the user association log could make it more resistant to potential privacy risks. We consider the following two questions: 1) Using only the "insensitive" information in an anonymized user association log, is it possible to build a unique signature for each user? These signatures, when combined with some auxiliary information, such as a short-term un-anonymized log, can be used to distinguish users and infer some sensitive information from the anonymized log. 2) If privacy breach is possible, how effective is a proposed mitigation approach in preventing an adversary from building such signatures?

In a nutshell, we make three major contributions in this work. First, we simulate the role of an adversary and propose a "correlation attack" – a method based on Conditional Random Field (CRF) – that can be used to breach user privacy from a released WLAN user association log. Second, we use extensive experiments to demonstrate the effectiveness of the CRF-based correlation attack. Using an anonymized campus-wide WLAN user association log with more than four thousand users and millions of user association records, and a short-term observation of the victim's association activities, we show that the CRF-based correlation attack, under certain circumstances, can reveal the victim's identity in the released dataset with a probability as high as 70%, or narrow down the victim's identity among 20 candidates with a probability close to 100%. Third, we evaluate the effectiveness of standard sanitization techniques, including generalization and perturbation, in mitigating the proposed correlation attack; the results reveal only limited success of these methods, suggesting that more thorough treatment is needed when anonymizing wireless user

association logs before the public release.

The remainder of this paper is organized as follows. We first present related work in Section II. In Section III, we introduce how user association logs are collected in WLANs and describe the common practice in sanitizing these data before sharing them with the public. In Section IV, we discuss the adversarial model and formulate the correlation attack problem. We present in Section V how an adversary can use a CRF-based technique to launch correlation attacks against released user association datasets, and evaluate the attack effectiveness of this method in Section VI. In Section VII, we consider two widely used anonymization techniques, generalization and perturbation, and evaluate their effectiveness in mitigating CRF-based correlation attacks. Finally, we draw concluding remarks in Section VIII.

II. Related Work

To share network traces while preserving privacy, data publishers usually define sanitization policies according to their specific privacy concerns. These policies determine which sanitization methods to apply and how. Many network-trace sanitization techniques and software tools have been proposed and implemented, such as FLAIM [41] and tcpmkpub [36], to fulfill the trace publishers' sanitization goals.

Due to the intrinsic complexity of network trace sanitization, however, recent research has revealed that there are few, if any, available network-trace sanitization schemes that can provide a water-tight guarantee under the worst-case analysis. These works often mimic the role of an adversary that tries to launch a de-sanitization attack against the sanitized trace. According to the employed attack strategies, these de-sanitization research can be classified into two categories. *Direct attacks* exploit the limitations of a sanitization algorithm [5], [33], [9]. *Indirect attacks* use implicit information contained in the trace [9], [14], [1], [35], auxiliary information obtained from other sources [23], [9], [35], or new techniques from other research fields, such as machine learning [8], [4], [35], [1], to uncover sensitive information from anonymized network traces.

In the domain of wireless networks, many physical-device-fingerprinting techniques could potentially be used to launch de-sanitization attacks [14], [1], [4]. Because most of these techniques work by monitoring unique variations in protocol behaviors, such as those seen across different vendors or device-driver implementations, they often require very-high-resolution data or even special measurement equipment. Such requirements greatly limit their applicability for de-sanitization on most types of released traces. Some other researchers have focused on how to fingerprint users. For instance, Pang et al. demonstrated that by combining information from multiple sources together, such as destination address, broadcast packet size and IEEE 802.11 MAC protocol fields, an adversary could uniquely identify users under certain circumstances [35]. Their techniques, however, rely on much more abundant trace semantics than our work and have only been evaluated with much smaller wireless network traces than the one we used. Previously, we speculated a potential threat against user mobility privacy in a general sense [45]. Most close to this work, Kumar and Helmy have recently shown that it is possible to breach privacy from WLAN user association logs [22]. Their attack model assumes that the adversary can inject data into the wireless network during the trace collection or has some out-of-band information such as the victim's academic major and gender. In practice, these conditions may be difficult to satisfy. The type of attacks we discuss in this paper, however, do not require these assumptions.

Location privacy has been investigated in diverse communication networks in the past few years. Krumm presented a comprehensive survey of computational location privacy, in which users' location data are treated as geometric information [20]. Hoh et al. analyzed a set of week-long GPS traces from 233 vehicles and showed that applying previous privacy algorithms either led to inaccurate results or failed to provide privacy guarantees; they further proposed an uncertainty-aware algorithm that is able to preserve privacy for all vehicles [17]. In comparison, the trace used in our work covered thousands of users and more than two months. Location privacy in sensor networks has also been studied under different adversarial models [32], [28], [34]. Our work differs from this line of research because it considers location privacy in a different network environment, which leads to a different threat model. For example, we do not assume that the adversary is capable of monitoring the entire network traffic in our work.

Narayanan et al. proposed a method to robustly de-anonymize a large dataset [29]. Their work is based on the assumption that the studied dataset is highly sparse; for example, in their studied Netflix dataset, the number of attributes (movies) is twenty times more than the number of potential targets (Netflix subscribers). In our study, we make no assumptions about the sparsity of user association logs.

Privacy has been intensively studied for a long time in the database field. Concepts such as k-anonymity [39], l-diversity [26], t-closeness [25], and differential privacy [13] have been analyzed in a theoretical manner under many settings [15], [6], [2]. Most related techniques aim to anonymize microdata, which is represented as a tuple with multiple attributes in a database table. The user association log, however, contains information with sequential semantics. An interesting research direction would be to develop methods to cast existing privacy-preservation concepts into a framework able to deal with sequential data, as required in sanitizing user association logs collected from WLANs, and we plan to explore further along this line in our future work. Recently, some efforts on analyzing privacy of graph data have shown that sensitive information in social network data can be de-anonymized [49], [3], [30]. Our work differs from this line of research because we deal with sequential activity data corresponding to individual users instead of the social graphs formed by the users.

III. WLAN User Association Logs

At Dartmouth College, we have been monitoring the campus-wide WLAN network usage since 2001. As of January

2010, this WLAN network consists of over 1300 Aruba APs that provide 54Mbps coverage to the entire campus. These Aruba APs are connected with and controlled by a small set of Aruba Mobility Controllers. We poll every controller every 5 minutes using the SNMP protocol and receive replies. In addition to traffic statistics, these replies contain a list of devices associated with every AP. After processing these replies, each row of the resulting user association log collected, which we call a *user association record*, has 4 comma-separated fields: the MAC address of the wireless card, the name of the AP that the wireless card has connected with, and the start and the end POSIX timestamp of this connection. The following is a snippet of the user association log that we extract from the SNMP information (it shows anonymized MAC addresses to protect user privacy):

```
001d4f3bc496,14.5.1, 1251690285,1251691544
002608e4cdf7,80.3.2, 1251690458,1251691544
0021e9082bfd,142.6.1,1251689384,1251691544
0016cf29eb6d,76.5.3, 1251691151,1251691544
001cb3b51b58,188.4.6,1251689569,1251691544
0016cf295f33,206.5.7,1251688817,1251691544
```

There are a few things worth noting. First, although it is possible that a wireless card may have been used in multiple devices or a device has been used by multiple people, we assume that such cases are rare in our dataset. Hence in this paper we use a "wireless card" and a "network user" (or a "user") interchangeably. Second, because the Aruba Mobility Controller only generates the start timestamp for each connection and we poll the controller every 5 minutes, the connection's end timestamp is only an estimated value, whose error is therefore bounded by 5 minutes. Third, we use a hierarchical naming scheme for APs in the dataset. For an AP named $x.y.z$, x is its building number, y is its floor number, and z is its serial number within the floor.

Sanitization. When sanitizing the user association logs, we use a one-to-one mapping function to rename the MAC addresses in the original dataset. Hence, the anonymized MAC addresses in the sanitized dataset do not have any physical meaning and are thus only symbolic names; a similar sanitization scheme has been used in other work [22]. By taking advantage of its hierarchical naming scheme, we truncate an AP's name according to different sanitization levels. For example, if we want to only keep building and floor information, we truncate the AP's name from $x.y.z$ to $x.y$.

IV. Threat Model and Problem Formulation

Complying with Narayanan's definition of privacy breach [29], the threat we study here is whether the limited insensitive information left in a sanitized association log could still form implicit signatures for individual users. These implicit signatures, when combined with auxiliary information, may provide the adversary the knowledge that the sanitization process has aimed to protect, such as whether an anonymized ID in the released dataset corresponds to a specific user. We make the following three assumptions in our threat model:

Assumption 1: The adversary has access to a sanitized WLAN user association log $\mathcal{L}_s$, which is shared to the public by a trace publisher. There are N_s users in this association log. All users' real MAC addresses are anonymized in $\mathcal{L}_s$ as follows: during the trace publisher's sanitization process, each real MAC address has been replaced with a new identifier ID_i $(1 \leq i \leq N_s)$ according to some one-to-one one-way mapping function. Hence, given an anonymized MAC address ID_i, the adversary cannot find the real MAC address that is mapped to ID_i. The AP's name can be either preserved or truncated. The rest of the fields, such as the start and end timestamp of each connection, are preserved during the sanitization process.

Assumption 2: The adversary knows a sequence of association records about a victim user's device. This sequence of records, $\mathcal{Q}$, need not be collected during the same time period as $\mathcal{L}_s$ (otherwise the problem will be trivial). It is important to note that the information provided in $\mathcal{Q}$ can be rather coarse. For example, the adversary may only need to know which buildings the victim has visited rather than which exact APs the victim has associated with.

There are a few ways for the adversary to obtain such information: (1) The adversary has some general knowledge about the victim. For example, the adversary knows the victim often stays in the library in the morning for 2 hours and then goes to the classroom around 3pm in the afternoon. (2) The adversary can manage to install some trojan software on the victim's device through some social engineering techniques (e.g., email attachments) or exploiting software vulnerabilities on the victim's device. The trojan secretly monitors the network association/disassociation activities of the device and reports them to the adversary through covert channels. (3) The adversary follows the victim physically and monitors the victim's network association/disassociation activities. For instance, when the victim opens a laptop, usually the laptop will automatically find the closest AP and connect to it, which leads to an association record. (4) The adversary can obtain the user association records of the victim user from a different dataset $\mathcal{L}'$, which may or may not be published by the same publisher as $\mathcal{L}_s$. $\mathcal{L}'$ may be produced using a weak anonymization scheme (or even no sanitization at all) so that it is easier for the adversary to identify the victim's AP association records in it than in $\mathcal{L}_s$.

Assumption 3: The adversary knows that the sanitized dataset $\mathcal{L}_s$ must contain the victim's AP association records. In many cases, $\mathcal{L}_s$ is published at an organization level (e.g., by a university) and thus contains complete AP association logs of the organization's wireless users. Hence, if the adversary knows that the victim was a member of the organization when $\mathcal{L}_s$ was collected, it is easy for him to know that $\mathcal{L}_s$ should contain the victim's AP association records.

Given the three assumptions in the adversarial model, the *(exact) correlation attack problem* is then formulated as follows: *given $\mathcal{L}_s$ and $\mathcal{Q}$, which anonymized identity ID_i $(1 \leq$*

$i \leq N_s$) *in* $\mathcal{L}_s$ *has also generated* $\mathcal{Q}$*?* In practice, however, due to incomplete data for training or inference, or some intra- and inter-user association activity variations, finding an algorithm to solve the exact correlation attack problem is difficult or even impossible. In this work, we consider a relaxed and more practical version of this problem. The *(relaxed) correlation attack problem* is formulated as follows: *given* $\mathcal{L}_s$ *and* $\mathcal{Q}$*, which subset of anonymized identities would contain the one that generated* $\mathcal{Q}$ *with high probability?*

It is important to emphasize the difference between the correlation attack problem and the mobility anomaly detection problem [43], [47]. The latter one is stated as follows: given the mobility history of a mobile user $\mathcal{H}$, is a test mobility record $\mathcal{R}$ generated from the same user? Although both problems are related to human mobility, the distinction in their conditions (i.e., prior knowledge) suggests the difference: The mobility anomaly detection problem is essentially a *statistical hypothesis test*, whose solution does not require the knowledge of other users' mobility history. In contrast, the correlation attack problem is about *classification*: considering that there are N_s classes and we know each class's association records, we want to find the correct class for the observed association sequence $\mathcal{Q}$. Their difference can further be explained with an example. It is possible that the observed association sequence $\mathcal{Q}$ does not exhibit the user's regular mobility pattern and can thus be treated as an anomaly in the mobility anomaly detection problem. But as long as no other users have association records closer to $\mathcal{Q}$, we may still be able to find the correct class for $\mathcal{Q}$ in the correlation attack problem.

V. Algorithm Description

The intuition behind the proposed algorithm is that human activities often follow certain regularities. These regularities are inherent in the temporal and spatial information of the association log, whether or not the log is sanitized. Different users may have different association patterns, and we can use such differences to fingerprint and distinguish users. To this end, we build a model that not only characterizes such inter-user differences but also is robust to intra-user variations. In the previous section, we formulate correlation attack as a classification problem, in which the two key components are feature representation and the learning algorithm. We use association activity templates to represent user association logs and employ CRF as the learning algorithm.

A. Data Representation

Previously, there are two general approaches to represent user association activities: *direct representation* and *abstract representation*. The method proposed by Song et al. [42], for instance, is a typical direct representation that puts all visited APs in an AP transition vector and the corresponding duration at each AP in a duration vector. Suppose that a user has traveled from AP1, AP2 to AP3 sequentially and connected with each AP for 30, 45, 25 minutes respectively. Then, the corresponding AP transition vector is [AP1, AP2, AP3] and the duration vector is [30 min, 45 min, 25 min].

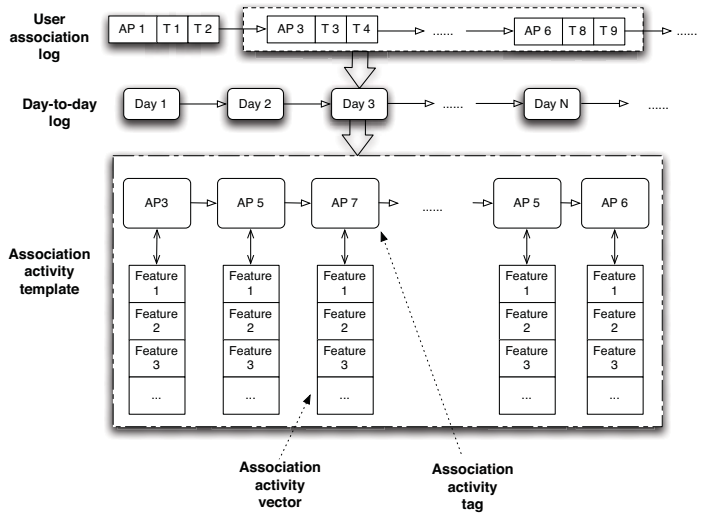

Fig. 1. Represent an user's association log using association activity template

While this method captures every AP association transition, it ignores other potentially valuable information, such as when the connection took place. On the other hand, the abstract representation method, such as Hsu et al.'s *normalized association vector* [18], aims to capture the overall trend of AP association changes at the expense of losing many details during the abstraction process.

As the previous data representation methods ignore details that are important to classification, we propose a new approach that uses *association activity templates* to represent user association logs. In this method, we first split the user's association log into day-to-day pieces and then for each day build an individual association activity template, because human activities often exhibit regularities associated with days of the week. An association activity template is a collection of association activity tags and their corresponding association activity vectors. As shown in Figure 1, the association activity tag is the name of the visited AP. Each element in an association activity vector is called a *feature*. In the current implementation, we let an activity vector have six features: *duration*, *day of week*, *starting time*, *previous AP*, *next to previous AP*, and *next AP*. Table I explains these features.

Several things are worth noting here. First, an association activity vector does not correspond one-to-one with an AP association record (i.e., a row in the user association log). This is because an association activity vector resides in an association activity template that only holds association information for a specific 24-hour calendar day. Thus, if an AP association record spans multiple days, it is divided into several association activity templates and represented by multiple association activity vectors inside these templates. The *duration* feature in an association activity vector follows this manner. If a connection is entirely contained in a 24-hour calendar day, the value of *duration* is the end timestamp less the start timestamp. If a connection spans several days, the value of *duration* is equal to this connection's cumulative amount of time in the corresponding 24-hour calendar day. Given the *duration* feature's maximum value is 24 hours, it

TABLE I
FEATURES OF AN ASSOCIATION ACTIVITY VECTOR

Feature name	Meaning	Value	Comments
duration	Adjusted connection duration	Integer	Normalized, inspired by Hsu's work [18]
day of week	Day of the week of this record	Enum. type, from Monday to Sunday	To represent periodic patterns, inspired by Kim's work [19]
starting time	Time slot of a day of this record	Enum. type, from Midnight to Night	
previous AP	The AP in the previous record	String, AP's name	To represent context information, inspired by Song's work [42] and YamCha [21]
next-to-previous AP	The AP in the next-to-previous record		
next AP	The AP in the next record		

is intrinsically normalized on a 24-hour base. Second, instead of assigning an exact time (hour, minute and second) to the *starting time* feature, we divide a day into six 4-hour slots (midnight, dawn, morning, afternoon, evening, and night) and use the name of these slots as the coarse start time. As shown in Kim's work [19], although user association behaviors have periodic patterns, they also have some variations. Compared to the exact representation, representing the time at a coarse level adds some tolerance for these variations. Third, at the beginning of each day, we assign a special string "NA" to the *previous AP* and *next-to-previous AP* features; similarly, we let the *next AP* feature be "NA" at the end of a day. Fourth, if a user is offline all day, no association activity template is generated for her that day.

B. Algorithm Procedure

To give a big picture about how the correlation attack works, we describe the attack algorithm in this section and defer the introduction to CRF to Section V-C.

Step 1. For each user in $\mathcal{L}_s$, split his/her association log into day-to-day pieces and represent each day's log using an association activity template as described in Section V-A.

Step 2. Feed each user's association activity templates into a linear-chain CRF to model this user's association behavior. As there are N_s users in $\mathcal{L}_s$, we build N_s CRF models. The input fed to a CRF model is a sequence of association activity vectors (Figure 1) and the output is a sequence of association activity tags, which are actually AP names. Let $CRF_i(\mathcal{V})$ denote the output from the i-th user's CRF model, where $1 \leq i \leq N_s$ and $\mathcal{V}$ denotes the sequence of association activity vectors fed to the CRF model.

Step 3. For the observed user association record $\mathcal{Q}$, we preprocess it as described in Section V-A to obtain an association activity template $\mathcal{T}$. Let $\mathcal{V}_\mathcal{T}$ and $\mathcal{G}_\mathcal{T}$ denote the sequence of association activity vectors and the sequence of association activity tags in template $\mathcal{T}$, respectively.

Step 4. We feed $\mathcal{V}_\mathcal{T}$ to all CRF models trained in Step 2 and count the number of tags that overlap between $\mathcal{G}_\mathcal{T}$ and $CRF_i(\mathcal{V}_\mathcal{T})$ $(1 \leq i \leq N_s)$, a score we denote w_i. The intuition applied here is that the victim's CRF model is more likely to produce correct activity association tags from her observed activity association vectors in $\mathcal{Q}$, and therefore score w_i is higher than the others if ID_i is the victim's identifier in the released user association log.

Step 5. We sort all users based on score w_i in non-increasing order and the algorithm outputs this sorted list.

Ideally the top identifier on the sorted list should be treated as the sole candidate that generated the observed user association sequence $\mathcal{Q}$. In practice, however, due to incomplete data for training or inference, or some intra- and inter-user association activity variations, the top identifier may not correspond to the victim who produced $\mathcal{Q}$. As mentioned earlier, we tackle the relaxed correlation attack problem instead and thus use a small number of top identifiers on the sorted list. Clearly, from the attacker's perspective, the smaller the number of top identifiers needed to include the victim's, the more successful his attack.

C. Conditional Random Field

One may wonder why we chose CRF models to characterize users' AP association behaviors. We explain this choice by analyzing the nature of the correlation attack problem and also provide a brief introduction to CRF.

Let $X = (X_1, X_2, ..., X_n)$ denote a random variable of an observed sequence, each element of which has k features. In our problem, a realization of X is a sequence of association activity vectors with the six features described in Table I. Let Y denote a random variable of a label sequence. A label here is actually an association activity tag that indicates an AP name. According to Figure 1, each association activity vector corresponds to an association activity tag. Hence, given an observed sequence of X (i.e., sequence $\mathcal{V}_\mathcal{T}$ in Step 3 of the algorithm shown in Section V-B), we need to produce a label sequence for it. It is thus a task of assigning label sequences to observation sequences, which is common to many applications in bioinformatics, computational linguistics and speed recognition [12], [27], [37].

We now explain why here we do not use Hidden Markov Model (HMM), a popular probabilistic sequence model that characterizes the joint distribution $p(X, Y)$ directly [38]. HMM is known to be a *generative* model in the field of graphical models. The challenge facing HMM is that it has to model the entire set of observation sequences $p(X)$ explicitly, which is intractable in our case (and many other domains) for two reasons. First, the limited data collected from real-life network measurement makes it difficult to obtain a full-fledged $p(X)$. Second, the features in X (the features in the association activity vector) can be highly correlated. For example, Song's work shows that there is a strong correlation between the lastest three APs visited by an user [42]. Kim's work demonstrates that the time and the location that a user will visit may follow a periodical pattern [19]. Such dependences among features are difficult to model within HMM. To circumvent the

problem, generative models like HMM and Naive Bayes make independence assumptions that may not be realistic in practice.

Note, however, that modeling the joint distribution for X and Y (i.e., $p(X, Y)$) is not important for the sequence labeling problem at all, because the observation sequences have already been available to us. What is needed is actually finding the conditional probability $p(Y|X)$ from the training dataset. Although it is possible to derive $p(Y|X)$ as $\frac{p(Y)p(X|Y)}{p(X)}$ based on Bayes' rule, the need to model the marginal distribution $p(X)$ makes it a difficult approach. The CRF method, in contrast, eliminates the necessity of knowing $p(X)$ by building models to predict label sequences Y conditional on observation sequences X. Hence, CRF is indifferent to the dependence among features in X because X is now treated as given (i.e., a condition). Because CRF models the conditional probability $p(Y|X)$ instead of the joint distribution $p(X, Y)$, it is a *discriminative* approach rather than a generative one.

CRF is a special type of undirected graphic model. Let $\mathcal{C}$ denote the entire set of cliques, which are fully connected subgraphs, in the graph. A clique $C \in \mathcal{C}$ contains variables from X, denoted X_C, and also variables from Y, denoted Y_C. For a generic CRF, the goal is to learn the following conditional distribution from the training data:

$$p(Y|X) = \frac{1}{Z(X)} \prod_{C \in \mathcal{C}} \psi_C(Y_C, X_C), \tag{1}$$

where $Z(X)$, sometimes called the partition function, is a normalization factor and is given by:

$$Z(X) = \sum_{Y} \prod_{C \in \mathcal{C}} \psi_C(Y_C, X_C). \tag{2}$$

Furthermore, ψ_C is a real-valued potential function on clique C; a commonly used function is:

$$\psi_C(Y_C, X_C) = \exp(\sum_i \lambda_i f_i(Y_C, X_C)), \tag{3}$$

where f_i is a feature function and λ_i is the weight of feature function f_i.

There is a special type of CRF models, called *linear-chain CRF* models, which are particularly useful for solving sequence labeling problems. Linear-chain CRF models are conditionally trained as linear chains, instead of generic undirected graphical models. In Figure 2, we show a linear-chain CRF, where the node representing X is not generated from the model. In this work, we used CRFsuite [31], a linear-chain CRF implementation for parameter estimation and inference, and defined the state feature function and transition feature function as boolean functions, which are similar to those in Sutton and McCallun's book [44]. It is worth noting that many methods have been proposed to train linear-chain CRF models and use them for inference. Due to space limitation, we refer interested readers to the literature for more thorough treatment on the topic of CRF [44], [24], [46].

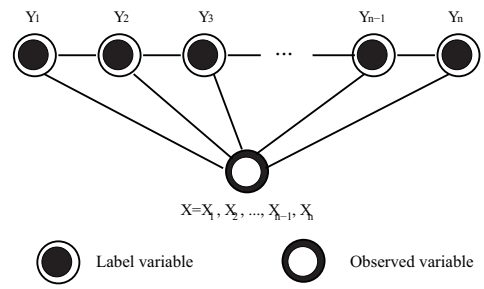

Fig. 2. Illustration of linear-chain CRF

VI. Experimental Evaluation

In this section, we evaluate the effectiveness of the CRF-based method for correlation attacks. We use the user association log extracted from the SNMP log collected at Dartmouth College between January 4, 2010 and March 6, 2010, which in total covered 62 days corresponding to one academic term. In the original dataset, there were 19,579 distinct MAC addresses, which contributed to 3,076,318 association records. Because the WLAN at Dartmouth College is an open network, any one physically at the campus site can use this network for free, and thus a great portion of MAC addresses belong to visitors who have appeared in the logs for only a short period of time. As training CRF models for these transient users would be difficult due to insufficient data, we filtered out those users who were active in fewer than 45 days during this 62-day period, and the resulting dataset still contained 79.67% of the user association records with 4,285 distinct users and 1,364 distinct APs. We used this reduced dataset for the experiments below. All the experiments were performed on four commodity PCs, which took around four days to finish.

We use the *Minimum Size of Candidate Identifier Set* (MSCIS) as the metric to measure the attack efficiency. Consider the relaxed correlation attack problem with a sanitized user association dataset $\mathcal{L}_s$ and an observed sequence of AP association records $\mathcal{Q}$. For each ID_i where $1 \leq i \leq N_s$ in $\mathcal{L}_s$, we compute score w_i according to Step 4 in the CRF-based method. Suppose that ID_j is the user ID of the victim who generated $\mathcal{Q}$. The MSCIS is defined as the number of user IDs whose scores are no smaller than w_j. MSCIS establishes an upper bound on how many candidate user IDs need be considered in order to contain the victim's user ID in the sanitized dataset. Note that if a user has the same score as the victim's (i.e., w_j), his ID should also be counted into MSCIS.

We perform 10-round *leave-one-out* experiments. The 62-day user association log is partitioned into 10 bins of approximately the same length for each user. In the j-th round ($1 \leq j \leq 10$), we use the j-th bin of each user's association records as the testing dataset ($\mathcal{L}_u$) and the remaining nine as the training dataset ($\mathcal{L}_s$) to build the CRF models. The results shown below are the 10-round averages.

To set up a baseline case for comparison, we developed a simple distance-based method described as follows:

Step 1. For each user in $\mathcal{L}_s$, we build a time vector each day that contains how much time this user spent at

each AP. The length of a time vector is equal to the total number of unique APs in the trace, and the number of time vectors for a given user is equal to the number of days that the user appeared in $\mathcal{L}_s$.

Step 2. Similarly, we compute a set of daily time vectors for each user in $\mathcal{L}_u$.

Step 3. For each user in $\mathcal{L}_u$, we compute the Euclidean distance between each of her time vectors and every user's time vectors in $\mathcal{L}_s$, to obtain an average score for every user in $\mathcal{L}_s$.

Step 4. For each user in $\mathcal{L}_u$, we sort the scores derived from Step 3 in non-descreasing order to obtain a sorted list of user IDs in $\mathcal{L}_s$, then compute the MSCIS for each user in $\mathcal{L}_u$.

Figure 3 compares the results of the CRF-based method and the distance-based method. The sanitization is done by anonymizing only the MAC addresses but leaving the other fields intact (other sanitization strategies will be examined in Section VII). When the length of $\mathcal{Q}$ is 5-6 days, the CRF-based method significantly outperforms the distance-based method in attack efficiency: 73.21% of the 4,285 users can be pinpointed exactly from $\mathcal{L}_s$; for 80.12% of the users, their MSCIS is no more than 2, meaning that the victim's ID appears among the top two candidates according to the CRF-based method; for 99.72% of the users, their MSCIS is no more than 20. Hence, using the CRF-based method, the adversary could almost surely narrow down the victim's possible user ID into a set of 20 candidates from the user association dataset with more than 4,000 users.

By tuning the length of $\mathcal{Q}$ to different values (from 5-6 days to 2 or 3 days), we show how the amount of auxiliary knowledge affects the attack efficiency. Clearly, reducing the auxiliary knowledge available to the attacker (shorter $\mathcal{Q}$) degrades the performance of the attack. However, even in the worst case here that the length of $\mathcal{Q}$ is only two days, the adversary still can pinpoint her identity exactly from $\mathcal{L}_s$ with probability 61.67%, and for 98.51% of the users, he can narrow down her identity in $\mathcal{L}_s$ to only 20 candidates. From the attacker's perspective, this is favorable because he needs to know a victim's association activities for only a short period to launch the correlation attack effectively.

VII. Mitigation Strategies

In the previous section, we play an adversary's role and evaluate the effectiveness of the CRF-based correlation attack under different amount of auxiliary information. As a network trace publisher in real life and the host of the CRAWDAD website [10], we are also interested in how well standard sanitization measures can prevent such privacy breaches.

Generally speaking, there are four categories of approaches to anonymizing datasets to protect privacy: *suppression-based* methods remove information from the data, *generalization-based* methods coarsen the level of information released in the data, *perturbation-based* methods add noise into the data, and *permutation-based* methods swap sensitive associations between entities [7]. Because the information provided in a user

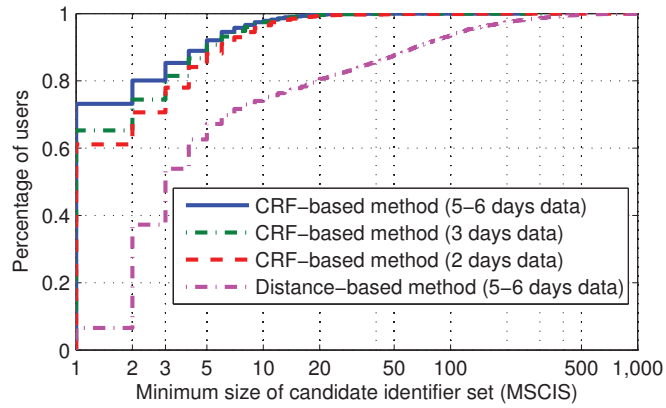

Fig. 3. Relationship between the attack performance and the amount of auxiliary information

association record is already limited, removing any field in it would make a released dataset hard to use. On the other hand, the identity information in released AP association records has been anonymized and thus swapping identity information between different users does not prevent correlation attacks discussed in this work. Therefore, in the following we focus on analyzing the effectiveness of generalization-based and perturbation-based methods in mitigating correlation attacks.

A. Generalization

Recall that the AP-naming scheme in the user association logs uses a hierarchical structure: building ID, floor level, and AP serial number. Hence, it is natural to apply generalization on the AP names. We consider two generalization schemes here: one keeping only the building information of each AP, and the other keeping both the building ID and the floor level. Using these two generalization schemes, we obtain two anonymized datasets and then apply the CRF-based method to launch correlation attacks against them. The results, together with results from CRF without any generalization, are depicted in Figure 4. All the experiments in Section VII work on the same sanitized dataset $\mathcal{L}_s$ and unsanitized dataset $\mathcal{L}_u$ (with 5-6 days) as those in the previous section.

It is clear that applying generalization-based anonymization techniques helps mitigate correlation attacks. For instance, keeping the building and floor level information, the probability of pinpointing the exact user is reduced from 73.21% to 70.92%, and the probability of having the victim appear among the top five candidates is reduced from 92.09% to 83.64%; keeping only the building information, the top one and top five ratios are further reduced to 64.78% and 74.10%, respectively. On the other hand, because keeping only the AP's building information is the best we can do to generalize AP names, we can see only limited effectiveness of generalization-based schemes in mitigating correlation attacks on user association logs. As a further step, one may consider anonymizing the building information, such as using a one-way function to rename them. Its effectiveness is, however, still questionable as Yoon's work [48] has shown that it is easy to re-identify

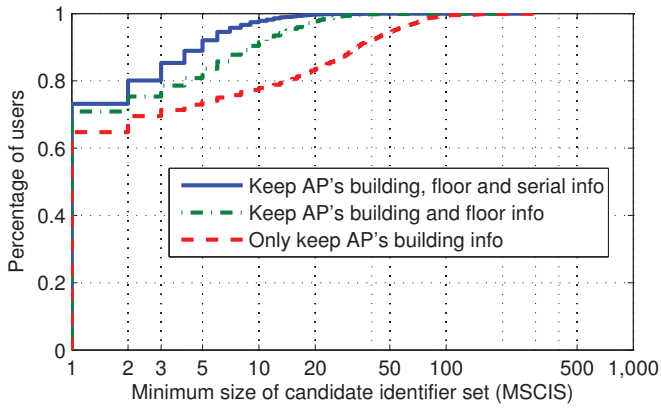

Fig. 4. Effectiveness of generalization-based mitigation against the proposed correlation attack

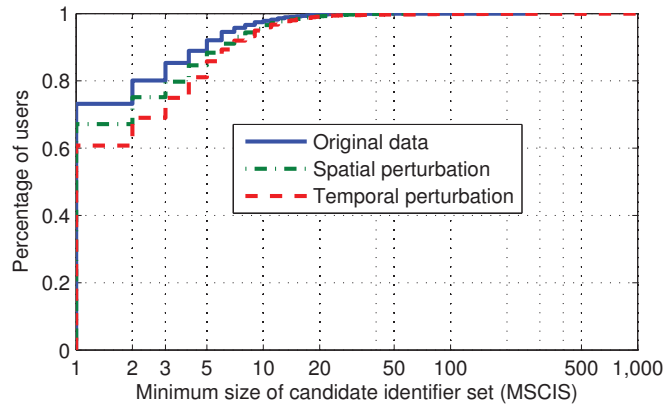

Fig. 5. Effectiveness of perturbation-based mitigation against the proposed correlation attack

the real building information even though they have been anonymized in our previously published trace [16].

B. Perturbation

Perturbation is another commonly used technique for data sanitization. Its key idea is to add some noise into the original dataset such that user privacy can be preserved while the usability of the dataset is still ensured. Based on the characteristics of the user association logs, we consider two perturbation methods: spatial perturbation and temporal perturbation.

- The *spatial perturbation* method changes the AP information in the original dataset as follows. Let $\mathcal{S}_i$ denote the sequence of user ID_i's AP association records, sorted in increasing order of starting timestamps. For each record R_j in $\mathcal{S}_i$, we change the AP in R_j to the AP in R_{j-1} with probability 15%, change it to the AP in R_{j+1} with probability 15%, or keep it intact with probability 70%.
- The *temporal perturbation* method changes the start and end timestamps in the original dataset as follows. For each AP association record, we add Gaussian noise with mean 0 and standard deviation 3600 seconds to its start and end timestamps. During the process of adding noise, we do it sequentially on each user's AP association records and ensure that the starting timestamp of the current AP association record is always greater than the end timestamp of the previous AP association record after noise is added.

The effectiveness of both methods in mitigating correlation attacks is illustrated in Figure 5. Not surprisingly, both methods make it more difficult for the adversary to launch correlation attacks. Using spatial perturbation, the probability of pinpointing the exact user is reduced from 73.21% to 67.14%, and the probability of having the victim appear among the top five candidates is reduced from 92.09% to 88.03%. On the other hand, if temporal perturbation is applied, the top one and top five ratios are reduced to 60.77% and 85.83%, respectively.

Considering the results in Figures 4 and 5, we conclude that for all the mitigation techniques evaluated, their effectiveness in mitigating CRF-based correlation attacks is rather limited. For instance, none of these methods is able to reduce the probability of pinpointing the exact user ID below 55%. Although adding more noise in the perturbation-based methods can further constrain the adversary's capability in launching correlation attacks, it may also damage the usability of the released user association datasets. In our future work, we shall consider the detailed use of user association logs (e.g., predicting mobility of wireless users [42]) and further explore the tradeoff between their usability and privacy.

VIII. Conclusion

User association logs collected from real-world WLANs have played an important role in understanding these networks. Sharing them with the public, however, poses potential risks to the privacy of the users involved. In this work, we show that people's association behaviors form implicit signatures for individual users. When combined with auxiliary information, such signatures can help reveal the true identities of anonymized IDs in a sanitized WLAN user association log. On a pessimistic note, standard anonymization techniques, such as generalization and perturbation, are unable to mitigate such CRF-based correlation attack effectively. The results from this work call for a more thorough study of potential privacy risks when wireless user association logs are shared with the public.

References

[1] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker. Using wireless physical layer information to construct implicit identifiers. In *Proceedings of HotPETS 2008*, July 2008.

[2] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pages 217–228. IEEE Computer Society, 2005.

[3] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava. Class-based graph anonymization for social network data. In *Proceedings of the VLDB Endowment*, volume 2, pages 766–777. VLDB Endowment, 2009.

[4] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the ACM Conference on Wireless Network Security (WiSec)*, pages 56–61. ACM Press, 2008.

[5] T. Brekne, A. Årnes, and A. Øslebø. Anonymization of IP traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies. In *Proceedings of the International Symposium on Privacy Enhancing Technologies (PET)*, pages 179–196. Springer-Verlag, 2005.

[6] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy-preserving data publishing. *Foundations and Trends in Databases*, 2(1–2):1–167, 2009.

[7] G. Cormode and D. Srivastava. Anonymized data: generation, models, usage. In *Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1015–1018. ACM, 2009.

[8] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter. On web browsing privacy in anonymized netflows. In *Proceedings of the USENIX Security Symposium*, pages 1–14. USENIX, 2007.

[9] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, and M. K. Reiter. Playing Devil's advocate: Inferring sensitive information from anonymized network traces. In *Proceedings of the Annual Symposium on Network and Distributed System Security (NDSS)*. IEEE Press, February 2007.

[10] Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD). http://www.crawdad.org/.

[11] E. C. R. de Oliveira and C. V. N. de Albuquerque. NECTAR: a DTN routing protocol based on neighborhood contact history. In *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC)*, pages 40–46. ACM, 2009.

[12] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[13] C. Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12. Springer, 2006.

[14] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of USENIX Security*. USENIX, 2006.

[15] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith. Composition attacks and auxiliary information in data privacy. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 265–273. ACM, 2008.

[16] Tristan Henderson, David Kotz, Ilya Abyzov, and Jihwang Yeo. CRAWDAD trace set dartmouth/campus/snmp (v. 2004-11-09). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus/snmp, November 2004.

[17] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 161–171. ACM Press, 2007.

[18] W. J. Hsu, D. Dutta, and A. Helmy. Mining behavioral groups in large wireless LANs. In *Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 338–341. ACM, 2007.

[19] M. Kim and D. Kotz. Periodic properties of user mobility and access-point popularity. *Journal of Personal and Ubiquitous Computing*, 11(6):465–479, August 2007. Special Issue of papers from LoCA 2005.

[20] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[21] T. Kudoh and Y. Matsumoto. Use of support vector learning for chunk identification. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 142–144, 2000.

[22] U. Kumar and A. Helmy. Human behavior and challenges of anonymizing WLAN traces. In *Proceedings of GLOBECOM*, 2009.

[23] U. Kumar, N. Yadav, and A. Helmy. Gender-based feature analysis in campus-wide WLANs. *SIGMOBILE Mobile Computing and Communications Review*, 12(1):40–42, 2008.

[24] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[25] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[26] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 24–85, 2006.

[27] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.

[28] K. Mehta, D. Liu, and M. Wright. Location privacy in sensor networks against a global eavesdropper. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2007.

[29] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, pages 111–125. IEEE Press, 2008.

[30] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pages 173–187. IEEE Computer Society, 2009.

[31] N. Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/, 2007.

[32] Y. Ouyang, Z. Le, D. Liu, J. Ford, and F. Makedon. Source location privacy against laptop-class attacks in sensor networks. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks (SecureComm)*, pages 1–10. ACM, 2008.

[33] L. Overlier, T. Brekne, and A. Arnes. Non-expanding transaction specific pseudonymization for IP traffic monitoring. In *Proceedings of the Cryptology and Network Security (CANS)*, pages 261–273. Springer-Verlag, 2005.

[34] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 88–93. ACM, 2004.

[35] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 99–110. ACM Press, 2007.

[36] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, 2006.

[37] L. Rabiner and B. H. Juang. Fundamentals of speech recognition. In *Prentice Hall Signal Processing Series*. Prentice-Hall, Inc., 1993.

[38] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, February 1989.

[39] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International, 1998.

[40] S. Sarat and A. Terzis. On the detection and origin identification of mobile worms. In *Proceedings of the 2007 ACM Workshop on Recurring Malcode (WORM)*, 2007.

[41] A. Slagell, K. Lakkaraju, and K. Luo. FLAIM: A multi-level anonymization framework for computer and network logs. In *Proceedings of the USENIX Large Installation System Administration Conference (LISA)*, December 2006.

[42] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next cell predictors with extensive Wi-Fi mobility data. *IEEE Transactions on Mobile Computing*, 5(12):1633–1649, December 2006.

[43] B. Sun, F. Yu, K. Wu, Y. Xiao, and V. C. M. Leung. Enhancing security using mobility-based anomaly detection in cellular mobile networks. *IEEE Transactions on Vehicular Technology*, 55(3):1385–1396, 2006.

[44] C. Sutton and A. McCallum. *Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006.

[45] K. Tan, G. Yan, J. Yeo, and D. Kotz. A correlation attack against user mobility privacy in a large-scale WLAN network (extended abstract). In *Proceedings of the Mobicom S3 workshop*, 2010.

[46] H. M. Wallach. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania CIS, 2004.

[47] G. Yan, S. Eidenbenz, and B. Sun. Mobi-watchdog: you can steal, but you can't run! In *Proceedings of the Second ACM Conference on Wireless Network Security (WiSec)*, pages 139–150. ACM, 2009.

[48] J. Yoon, B. D. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 177–190. ACM, 2006.

[49] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the 1st ACM SIGKDD Workshop on Privacy, Security, and Trust in KDD (PinKDD)*, 2007.