802.11 User Fingerprinting

Jeffrey Pang* Ben Greenstein† Ramakrishna Gummadi‡
Srinivasan Seshan* David Wetherall†§

*Carnegie Mellon University †Intel Research Seattle
‡University of Southern California §University of Washington

jeffpang@cs.cmu.edu benjamin.m.greenstein@intel.com gummadi@usc.edu
srini@cmu.edu djw@cs.washington.edu

ABSTRACT

The ubiquity of 802.11 devices and networks enables anyone to track our every move with alarming ease. Each 802.11 device transmits a globally unique and persistent MAC address and thus is trivially identifiable. In response, recent research has proposed replacing such identifiers with pseudonyms (i.e., temporary, unlinkable names). In this paper, we demonstrate that pseudonyms are insufficient to prevent tracking of 802.11 devices because *implicit identifiers*, or identifying characteristics of 802.11 traffic, can identify many users with high accuracy. For example, even without unique names and addresses, we estimate that an adversary can identify 64% of users with 90% accuracy when they spend a day at a busy hot spot. We present an automated procedure based on four previously unrecognized implicit identifiers that can identify users in three real 802.11 traces even when pseudonyms and encryption are employed. We find that the majority of users can be identified using our techniques, but our ability to identify users is not uniform; some users are not easily identifiable. Nonetheless, we show that even a single implicit identifier is sufficient to distinguish many users. Therefore, we argue that design considerations beyond eliminating explicit identifiers (i.e., unique names and addresses), must be addressed in order to prevent user tracking in wireless networks.

Categories and Subject Descriptors:
C.2.1 Computer-Communication Networks: Network Architecture and Design

General Terms: Measurement, Security

Keywords: privacy, anonymity, wireless, 802.11

1. INTRODUCTION

The alarming ease with which third parties can track our every move has drawn the concern of the popular media [1, 2], the United States government [22, 40], and technical standards bodies [17]. The fear is that we are sacrificing our *location privacy* due to the ubiquity of wireless devices that disclose our locations, identities, or both. Though this fear has focused on large scale wireless systems, such as cellular phone networks, the capability to track user location in such systems has typically been limited to service providers that are legally bound to protect our privacy. In contrast, the low cost of 802.11 hardware and ease of access to network monitoring software—all that is required for someone to locate others nearby and eavesdrop on their traffic—enable *anyone* to track users. Furthermore, although the popular press raised awareness about tracking threats posed by emerging wireless technologies, such as RFID [13], no such campaign has been waged to educate users about 802.11 devices and networks, which pose the same threats *today*.

The best practices for securing 802.11 networks, embodied in the 802.11i standard [16], provide user authentication, service authentication, data confidentiality, and data integrity. However, they do not provide anonymity, a property essential to prevent location tracking. For example, it is trivial to track an 802.11 device today since each device advertises a globally unique and persistent MAC address with every frame that it transmits. To mask this identifier, researchers have proposed applying *pseudonyms* [14, 18] (i.e., temporary, unlinkable names) by having users periodically change the MAC addresses of their 802.11 devices.

In this paper, we demonstrate that pseudonyms are insufficient to provide anonymity in 802.11. Even without a unique address, characteristics of users' 802.11 traffic can identify them implicitly and track them with high accuracy. An example of such an *implicit identifier* is the IP address of a service that a user frequently accesses, such as his or her email server. In a population of several hundred users, this address might be unique to one individual; thus, the mere observation of this IP address would indicate the presence of that user. Of course, in a wireless network that employs link-layer encryption, IP addresses would not be visible to an eavesdropper. However, other implicit identifiers would remain and these identifiers can be used in combination to identify users accurately.

This paper quantifies how well a passive adversary can track users with four implicit identifiers visible to commodity hardware. We thereby place a *lower bound* on how accurately users can be identified implicitly, as more implicit identifiers and more capable adversaries exist in practice. We make the following contributions:

- We identify four previously unrecognized implicit identifiers: network destinations, network names advertised in 802.11 probes, differing configurations of 802.11 options, and sizes of broadcast packets that hint at their contents.
- We develop an automated procedure to identify users. This procedure allows us to quantify how much information implicit identifiers, both alone and in combination, reveal about several hundred users in three empirical 802.11 traces.

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.

- Our evaluation shows that users emit highly discriminating implicit identifiers, and, thus, even a small sample of network traffic can identify them more than half (56%) of the time in public networks, on average. Moreover, we will almost never mistake them as the source of other network traffic (1% of the time). Since adversaries will obtain multiple traffic samples from a user over time, this high accuracy in traffic classification enables them to track many users with even higher accuracy in common wireless networks. For example, an adversary can identify 64% of users with 90% accuracy when they spend a day at a busy hot spot that serves 25 concurrent users each hour.
- To our knowledge, we are the first to show with empirical evidence that design considerations beyond eliminating explicit identifiers, such as unique names and addresses, must be addressed to protect anonymity in wireless networks.

In Section 2 we illustrate the power of implicit identifiers with several real examples. Section 3 covers related work. Section 4 explains our experimental methodology. Section 5 describes our empirical 802.11 traces. Section 6 analyzes how well 802.11 users can be identified using each implicit identifier individually. Section 7 examines how accurately an adversary can track people using these implicit identifiers in public, home, and enterprise networks. We conclude in Section 8.

2. THE IMPLICIT IDENTIFIER PROBLEM

How significantly do implicit identifiers erode location privacy? Consider the seemingly innocuous trace of 802.11 traffic collected at the 2004 SIGCOMM conference, now anonymized and archived for public use [31]. Interestingly, hashing real MAC addresses to pseudonyms is also the best practice for anonymizing traces such as this. Unfortunately, implicit identifiers remain and they are sufficient to identify many SIGCOMM attendees. For example:

Implicit identifiers can identify us uniquely. One particular attendee's laptop transmitted requests for the network names "MIT," "StataCenter," and "roofnet," identifying him or her as someone probably from Cambridge, MA. This occurred because the default behavior of a Windows laptop is to actively search for the user's preferred networks by name, or Service Set Identifier (SSID). The SSID "therobertmorris" perhaps identifies this person uniquely [26]. A second attendee requested "University of Washington" and "djw." The last SSID is unique in the SIGCOMM trace and suggests that this person may be University of Washington Professor David J. Wetherall, one of our coauthors. More distressingly, Wigle [39], an online database of 802.11 networks observed around the world, shows that there is only one "djw" network in the entire Seattle area. Wigle happens to locate this network within 192 feet of David Wetherall's home.

Implicit identifiers remain even when counter measures are employed. Another SIGCOMM attendee transferred 512MB of data via BitTorrent (this user contacted hosts on the typical BitTorrent port, 6881). A request for the SSID "roofnet" [32] from the same MAC address suggests that this user is from Cambridge, MA. Suppose that this user had been more stealthy and changed his or her MAC address periodically. In this particular case, since the user had not requested the SSID during the time he or she had been downloading, the MAC address used in the SSID request would have been different from the one used in BitTorrent packets. Therefore, we would not be able to use the MAC address to explicitly link "roofnet" to this poor network etiquette. However, the user does access the same SSH and IMAP server nearly every hour and was the only user at SIGCOMM to do so. Thus, this server's address is an implicit identifier, and knowledge of it enables us to link the user's sessions together.

Now suppose that the network employed link-layer encryption scheme, such as WPA, that obscures network addresses. Even then, we could link this user's sessions together by employing the fact that, of the 341 users that sent 802.11 broadcast packets, this was the only one that sent broadcast packets of sizes 239, 245, and 257 bytes and did so repeatedly throughout the entire conference. Furthermore, the identical 802.11 capabilities advertised in each session's management frames improves our confidence of this linkage because these capabilities differentiate different 802.11 cards and drivers. Prior research has shown that peer-to-peer file sharing traffic can be detected through encryption [42]. Thus, even if pseudonyms and link-layer encryption were employed, we could still implicate someone in Cambridge.

Implicit identifiers are exposed by design flaws. These examples illustrate three shortcomings of the 802.11 protocol beyond exposing explicit identifiers, none of which is trivially fixed. These shortcomings afflict not only 802.11 but many wireless protocols, including Bluetooth and ZigBee.

Identifying information exposed at higher layers of the network stack is not adequately masked. For example, even with encryption, packet sizes can be identifying. Padding, decoy transmissions, and delays may hide information exposed by size and timing channels, but increase overhead. For example, Sun *et al.* [34] found that 8 to 16 KB of padding is required to hide the identity of web objects. The performance penalty due to this overhead would be especially acute in wireless networks due to shared nature of the medium.

Identifying information during service discovery is not masked. 802.11 service discovery can not be encrypted since no shared keys exist prior to association. This raises the more general problem of how two devices can discover each other in a private manner, which is expensive to solve [4]. This problem arises not only when searching for access points, but also when clients want to locate devices in ad hoc mode, such as when using a Microsoft Zune to share music or a Nintendo DS to play games with friends.

Identifying information exposed by variations in implementation and configuration is not masked. Each 802.11 implementation typically supports different 802.11 features (e.g., supported rates) and has different timing characteristics. This problem is difficult to solve due to the inherent ambiguity of human specifications and manufacturers' and network implementers' desire for flexibility to meet differing constraints.

Balancing the costs involved in rectifying these shortcomings with the incentives necessary for deployment is itself a challenge. Nonetheless, rectifying these flaws at the protocol level is important so that users need not limit their activities in order to protect their location privacy. By measuring the magnitude with which each flaw contributes to the implicit identifier problem, our study provides insight into the proper trade-offs to make when correcting these design flaws in future wireless protocols. In the short term, our study may give guidance to individuals that are willing to proactively hide their identity in existing wireless networks.

In the remainder of this paper, we examine how these shortcomings impact the location privacy of a large number of users in different 802.11 networks and demonstrate that the examples described in this section are not isolated anomalies.

3. RELATED WORK

The challenge of hiding a user's identity has been examined in three different contexts: location privacy, identity hiding designs, and the study of other implicit identifiers. In this section, we describe the previous work in each of these areas.

Location Privacy. Location privacy has recently received significant attention, most notably in the RFID [13] and pervasive computing [7] fields. The concern is that location-aware applications, which use GPS and other positioning technologies, might reveal this information in undesirable ways. However, location privacy is threatened even by devices that do not explicitly track location. Since 802.11 users usually associate with access points that are less than tens of meters away, knowing the access point that a user is associated with gives away a coarse estimate of his location, such as his home or workplace. Moreover, systems that can employ multiple monitoring locations can use wireless signal strength to obtain an even more accurate estimate of a user's location [6, 35]. An added complication is that wireless devices are rapidly becoming integral parts of our daily lives. A resulting trend, which is evident from examining databases of access point locations [39], is the increasing availability of service, which is increasing the number of location tracking opportunities. Unfortunately, identifying individual users is often trivial since the 802.11 devices that they use are uniquely named by their MAC addresses.

Identity Hiding. Pseudonyms are widely used in systems, such as the GSM cellular phone network [15] to hide user identities. Gruteser *et al.* [14] and Jiang *et al.* [18] proposed using pseudonyms within 802.11 networks, and Stajano *et al.* [41] proposed a similar mechanism for Bluetooth. Using pseudonyms is a necessary first step to make tracking in these networks more difficult. However, we show that it is insufficient to protect location privacy because *implicit identifiers* can be sufficient to track users in many real scenarios.

Implicit Identifiers. Fingerprinting devices using implicit identifiers is not a new concept. For example, Franklin *et al.* [11] showed that it is possible to fingerprint device drivers using the timing of 802.11 probes. In contrast, our work attempts to pin down actual user identities rather than selecting among a few dozen drivers.

Kohno *et al.* [21] showed that devices could be fingerprinted using the clock skew exposed by TCP timestamps. We introduce new implicit identifiers that are useful in identifying users and, in contrast to TCP timestamps, three of our identifiers are still visible in wireless networks using link-layer encryption. Moreover, Kohno *et al.* note that one limitation of their work is that an adversary can not passively obtain timestamps from devices running the most prevalent operating system, Windows XP. For example, in two of our empirical traces, only 32% and 15% of the users sent TCP timestamps. All our identifiers have much at least 55% coverage.

Padmanabhan and Yang [29] explored fingerprinting users with "clickprints," or the paths that users take through a website. Their techniques rely on data from many user sessions collected at actual web servers. Our techniques can be employed passively by anyone with a wireless card without even associating to a network. These three research efforts compliment ours, since the procedure we develop for identifying users enables an adversary to use these implicit identifiers in combination with ours, yielding even more accurate user fingerprints. None of these previous efforts offer a formal method to combine multiple pieces of evidence. Moreover, to our knowledge, we are the first to evaluate the how well users are identified by implicit identifiers observed in empirical wireless data.

Implicit identifiers also reveal identity in other contexts. Security tools like `nmap` [12] and `p0f` [28] leverage differences in network stack behaviors to determine a device's operating system. Keystroke dynamics have been shown to accurately identify users [24, 33]. The timing and sizes of Web transfers often uniquely identify websites, even when transmitted over encrypted channels [8, 34]. Finally, there has been a large body of research in identifying applications from implicit identifiers in encrypted traffic [19, 20, 25, 42, 43]. Like many of these techniques which succeed in classifying applications accurately, we use a Bayesian approach.

4. EXPERIMENTAL SETUP

This section describes the evaluation criteria we use to determine how well several implicit identifiers can be used to track users.

The Adversary. Strong adversaries, such as service providers and large monitoring networks, obviously pose a large threat to our location privacy. However, the significance of the threat posed by 802.11 is that *anyone* that wishes to track users can do so.

Therefore, we consider an adversary that runs readily available monitoring software, such as `tcpdump` [37], on one or more laptops or on less conspicuous commodity 802.11 devices [3]. We further restrict adversaries by assuming that their devices listen passively. That is, they never transmits 802.11 frames, not even to associate with a network. This means that the adversary *can not be detected* by other radios. The adversary deploys monitoring devices in one or more locations in order to observe 802.11 traffic from nearby users. By considering a weak adversary, we place a lower bound on the accuracy with which users can be tracked, as stronger adversaries would be strictly more successful.

The Environments. An adversary's tracking accuracy will depend on the 802.11 networks he or she is monitoring. Since implicit identifiers are not perfectly identifying, it will be more difficult to distinguish users in more populous networks. In addition, different networks employ different levels of security, making some implicit identifiers invisible to an adversary. We consider the three dominant forms of wireless deployments today: public networks, home networks, and enterprise networks.

Public networks, such as hot spots or metro-area networks [27], are typically unencrypted at the link-layer. Although many public networks employ access control—for example, to allow access to only a provider's customers—most do so via authentication above the link-layer (e.g., through a web page) and by using MAC address filtering thereafter. Very few use 802.11i-compliant protocols that also enable encryption. Hence, identifying features at the network, link, and physical layers would be visible to an eavesdropper in such an environment. Unfortunately, this is the most common type of network today due to the challenge of secure key distribution.

Home and small business networks are small, but detecting when specific users are present is increasingly challenging due to the high density of access points in urban areas [5]. In addition, these networks are more likely to employ link-layer encryption, such as WEP or WPA, because the set of authorized users is typically known and is small. In cases where link-layer encryption is employed, an eavesdropper will not be able to view the payloads of data packets. However, features that are derived from frame sizes or timing, which are not masked by encryption, or from 802.11 management frames, which are always sent in the clear, remain visible.

Finally, security conscious enterprise networks are likely to employ link-layer encryption. Moreover, if the only authorized devices on the network are provided by the company, there will be less diversity in the behavior of wireless cards. For example, Intel corporation issues similar corporate laptops to its employees. We consider a enterprise network where only one type of wireless card and configuration is in use, so users can not be identified by differences in device implementation. However, features derived from the networks that users visit or the applications and services they run remain visible.

The Monitoring Scenario. We assume that users use different pseudonyms during each wireless session in each of these environments, as Gruteser *et al.* [14] propose. As a result, explicit identifiers can not link their sessions together. Sessions can vary in length, so we assume that every hour, each user will have a different pseudonym. We define a *traffic sample* to be one user's network traffic observed during one hour.

Although it is possible for users to change their MAC addresses more frequently, this is unlikely to be very useful in practice because other features, such as received signal strength, can link pseudonyms together at these timescales [6, 35]. Moreover, changing a device's MAC address forces a device to re-associate with the access point and, thus, disrupts active connections. In addition, it may require users to revisit a web page to re-authenticate themselves, since MAC addresses are tied to user accounts in many public networks. Users are unlikely to tolerate these annoyances multiple times per session.

Of course, the ability to link traffic samples together does not help an adversary detect a user's presence unless the adversary is also able to link at least one sample to that user's identity. In Section 2, we showed that identity can sometimes be revealed by correlating implicit identifiers with out-of-band information, such as that provided by the Wigle [39] location database. However, if the adversary knows the user he wishes to track, he can likely obtain a few traffic samples known to come from that user's device. For example, an adversary could obtain such samples by physically tracking a person for a short time. We assume the adversary is able to obtain this set of *training samples* either before, during, or after the monitoring period. Our results show that on average, only 1 to 3 training samples are sufficient to track users with each implicit identifier (see Section 6.2.3). The monitor itself collects samples that the adversary wants to test, which we call *validation samples*.

Evaluation Criteria. There are a number of questions an adversary may wish to answer with these validation samples. Who was present? When was user U present? Which samples came from user U? Essential to answering all these questions is the ability to classify samples by the user who generated them. In other words, given a validation sample, the adversary needs to answer the following question for one or more users U:

Question 1 *Did this traffic sample come from user U?*

Section 6 evaluates how well an adversary can answer this question with each of our implicit identifiers.

To demonstrate how well implicit identifiers can be used for tracking, we also evaluate the accuracy in answering the following:

Question 2 *Was user U here today?*

This question is distinct from Question 1 because an adversary can observe many traffic samples at any given time, any one of which may be from the target user U. In addition, a single affirmative answer to Question 1 does not necessitate a affirmative answer to Question 2 because an adversary may want to be more certain by obtaining multiple positive samples. Section 7 details the interaction between these questions and evaluates how many users can be tracked with high accuracy in each of the 802.11 networks described above.

5. WIRELESS TRACES

We evaluate the implicit identifiers of users in three 802.11 traces. We consider `sigcomm`, a 4 day trace taken from one monitoring point at the 2004 SIGCOMM conference [31], `ucsd`, a trace of all 802.11 traffic in U.C. San Diego's computer science building on November 17, 2006 [10], and `apt`, a 19 day trace monitoring all networks in an apartment building, which we collected. All traces were collected with `tcpdump`-like tools and only contain information that can be collected using standard wireless cards in monitor mode. The `ucsd` trace is the union of observations from multiple monitoring points. IP and MAC addresses are anonymized but are consistent throughout each trace (i.e., there is a unique one-to-one mapping between addresses and anonymized labels). Link-layer encryption (i.e., WEP or WPA) was not employed in either the `sigcomm` or `ucsd` network and neither trace recorded application packet payloads. In our analysis, we show that implicit identifiers remain even when we emulate link layer encryption and that we do not need packet payloads to identify users accurately. The `apt` trace only recorded broadcast management packets due to privacy concerns; hence, we only use it to study the one implicit identifier that is extracted from these packets.

We distinguish unique users by their MAC address since it is not currently common practice to change it. To simulate the effect of using pseudonyms, we assume that every user has a different MAC address each hour. Hence, we have one sample per user for each hour that they are active. To simulate the training samples collected by an adversary, we split each trace into two temporally contiguous parts. Samples from the first part are used as training samples and the remainder are validation samples. We choose a training period in each trace long enough to profile a large number of users. For the `sigcomm` trace, the training period covers the time until the end of the first full day of the conference. For the `ucsd` trace, the training period covers the time until just before noon. We skip one hour between the training and validation periods so user activities at the end of the training period are less likely to carry over to the validation period. For the `apt` trace, the training period covers the first 5 days. We consider a user to be present during an hour if and only if she sends at least one data or 802.11 probe packets during that time; i.e., if the user is actively using or searching for a wireless network.[1]

Table 1 shows the relevant statistics about each trace. Note that since can we only compute accuracy for users that were present in both the training and validation data, those are the only users that we profile. Therefore, results in this paper refer to 'Profiled Users' as the total user count and not 'Total Users.'

6. IMPLICIT IDENTIFIERS

In this section, we describe four novel implicit identifiers and evaluate how much information each one reveals. Our results show that (1) many implicit identifiers are effective at distinguishing individual users and others are effective at distinguishing groups of users; (2) a non-trivial fraction of users are trackable using any one highly discriminating identifier; (3) on average, only 1 to 3 training samples are required to leverage each implicit identifier to its full effect; and (4) at least one implicit identifier that we examine accurately identifies users over multiple weeks.

[1]We ignore samples that only contain other 802.11 management frames, such as power management polls. Including samples with these frames would not appreciably change the characteristics of the `sigcomm` workload, but would double the number of total "users" in the `ucsd` workload. This is because many devices observed in the `ucsd` trace were never actively using the network; we ignore these idle devices.

	sigcomm		ucsd		apt	
	training	validation	training	validation	training	validation
Duration (hours)	37	54	10	11	119	345
Total Samples	1974	3391	587	1240	638	1473
Frames Per Sample (median)	289	284	1227	1128	57	92
Total Users	377	412	225	371	97	196
Profiled Users	337	337	153	153	39	39
Samples Per Profiled User (mean)	5.5	9.1	3.1	4.7	14.7	32.2
Users Per Hour (mean)	53	64	59	113	5	4

Table 1—Summary of relevant workload statistics and parameters. The duration reports only hours with at least one active user.

6.1 Identifying Traffic Characteristics

Network Destinations. We first consider netdests, the set of IP <address, port> pairs in a traffic sample, excluding pairs that are known to be common to all users, such as the address of the local network's DHCP server. There are several reasons to believe that this set is relatively unique to each user. It is well known that the popularity of web sites has a Zipf distribution [9], so many sites are visited by a small number of users. In fact, in the `sigcomm` and `ucsd` training data, each <address, port> pair is visited by 1.15 and 1.20 users on average, respectively. The *set* of sites that a user visits is even more likely to be unique. In addition, users are likely to visit some of the same sites repeatedly over time. For example, a user generally has only one email server and a set of bookmarked sites they check often [36].

An adversary could obtain network addresses in any wireless network that does not enable link layer encryption. Even if users sent all their traffic through VPNs, the case for several users in the `sigcomm` trace, the IP addresses of the VPN servers would be revealing. No application or network level confidentiality mechanisms, such as SSL or IPSec, would mask this identifier either.

SSID Probes. Next we consider ssids, the set of SSIDs in 802.11 probes observed in a traffic sample. Windows XP and OS X add the SSID of a network to a preferred networks list when the client first associates with the network. To simplify future associations, subsequent attempts to discover *any* network will try to locate this network by transmitting the SSID in a probe request. As we observed in Section 2, SSID names can be distinguishing.[2] In addition, probes are never encrypted because active probing must be able to occur before association and key agreement.

There are two practical issues that limit the use of ssids as an implicit identifier. First, the preferred networks list changes each time a user adds a network, and thus a profile may degrade over time. Second, clients transmit the SSIDs on their preferred networks lists only when attempting to discover service. Therefore, clients may not probe for distinguishing SSIDs very often. While this is true, our results show that when distinguishing SSIDs are probed for, they can often uniquely identify a user. Since all users in the monitoring area are likely to use the SSIDs of the networks being monitored, these SSIDs are not distinguishing and we do not include them in the ssids set.

Broadcast Packet Sizes. We now consider bcast, the set of 802.11 broadcast packet sizes in each traffic sample. Many applications broadcast packets to advertise their existence to other machines on the local network. Due to the nature of this function, these packets often contain naming information. For example, in our traces, we observed many Windows machines broadcasting NetBIOS naming advertisements and applications such as FileMaker and Microsoft Office advertising themselves.

Application	Port	Number of Sizes
wireless driver or OS	NA	14
DHCP	67	14
sunrpc	111	1
NetBIOS	138	7
groove-dpp	1211	1
Microsoft Office v.X	2222	1
FileMaker Pro	5003	7
X Windows	6000	1

Table 2—A list of the most unique broadcast packets observed in the `sigcomm` trace. The third column shows the number of packet sizes that were emitted by at most 2 users.

Since these packets vary in length, their sizes can reveal information about their content even if the content itself is encrypted. Packet sizes alone appear to distinguish users almost as well as <application, size> tuples. For example, in the `sigcomm` trace, there are only 16% more unique tuples than unique sizes. Table 2 lists the most unique broadcast packet sizes we observed and the application port that generated them. Broadcast packets are sent to a known broadcast MAC address; thus, an adversary can distinguish them from other traffic even if link encryption is employed and the adversary is not granted network privileges. This set would remain identifying even when user behavior changes because most broadcast packets are emitted automatically.

Two types of broadcast packets, standard DHCP requests and power management beacons, are common to all users, since a device must send a DHCP request in order to obtain an IP address and sends power management beacons when in low power mode. Thus, we do not include these packets' sizes in the bcast set. These packets have distinct sizes (336 and 36 payload bytes, respectively) so they can be filtered even when link-layer encryption is enabled.

MAC Protocol Fields. Finally, we consider fields, the specific combination of 802.11 protocol fields visible in the MAC header that distinguish a user's wireless card, driver, and configuration. The fields included are the 'more fragments,' 'retry,' 'power management,' and 'order,' bits in the header, the authentication algorithms offered, and the supported transmission rates. Some card configurations can be more or less likely to emit different values in each of these fields, so they can distinguish users with different wireless cards. Although this identifier is unlikely to distinguish users uniquely, it can be combined with others to add more evidence. Moreover, many of these fields are available in any 802.11 packet, so they can almost always assist in identification. Furthermore, the likelihood of any particular field combination is unlikely to change for a user unless she obtains a new wireless device or

[2] A recent patch [23] to Windows XP allows a user to disable active probing, but it remains enabled by default because disabling it would break association in networks where the access point does not announce itself. In addition, revealing probes or beacons are still required for devices to discover each other in ad hoc mode.

driver; thus, fields should remain identifying over long time periods.

6.2 Evaluating User Distinctiveness

To show much information each identifier reveals, we now evaluate how accurately an adversary can answer Question 1 (see Section 4) using each implicit identifier.

6.2.1 *Methodology*

We construct a classifier C_U for each user U in our traces. Given a traffic sample s, C_U returns "Yes" if it believes the sample came from user U and "No" otherwise. We use a naïve Bayes classifier due to its effectiveness in application traffic classification [25, 42, 43]. More sophisticated classifiers exist, but this simple one is sufficient to demonstrate that implicit identifiers are a problem. Specifically, from each traffic sample, we extract a vector of features $(f_1, \ldots, f_m)$. In our case, $m \leq 4$, one feature per implicit identifier present in the sample. Each of our features has a different source, so we assume that they are independent. For each feature f_i, we estimate the posterior probability distribution $\Pr[s \text{ has } f_i | s \text{ is from } U]$ and the prior probability distribution $\Pr[s \text{ has } f_i]$ from training data. We are interested in $\Pr[s \text{ is from } U | s \text{ has } f_1, \ldots, f_m] =$

$$\frac{\prod_i^m (\Pr[s \text{ has } f_i | s \text{ is from } U]) \cdot \Pr[s \text{ is from } U]}{\prod_i^m \Pr[s \text{ has } f_i]} .$$

We classify a sample as being from U if and only if this value is greater than a threshold T. We also estimate the prior $\Pr[s \text{ is from } U]$ from training data, though this could also be based on a priori knowledge of how frequently the adversary believes his target will be present.

Feature Generation. To compute these probabilities, we must convert each of our implicit identifiers into a categorical or real-valued feature. We treat the fields identifier as a categorical feature by having each field combination represent a different value. Each of the other three identifiers is defined as a *set* of discrete *elements*; e.g., netdests is a set of network addresses. The following procedure describes how this set is converted into a real-valued feature that measures how similar it is to the target user's expected set.

We first construct a profile set, $Profile_U$, comprising all the elements in the union of all training samples for user U. To obtain a numeric value from the set of elements from a sample s, Set_s, we use a weighted version of the Jaccard similarity index [38] of the profile and the sample sets. The Jaccard index of two sets computes $J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$. However, some elements in each set are more discriminating than others (i.e., those that we observe in fewer users' traffic). Hence, we weight each element e by $w(e)$, the inverse of the number of users that accessed it. We learn these weights from the training data. Hence, given the profile $Profile_U$, the feature we compute for sample s is:

$$feature_U(s) = \frac{\sum_{e \in Profile_U \cap Set_s} w(e)}{\sum_{e \in Profile_U \cup Set_s} w(e)} .$$

This value quantifies how similar the set seen in the sample is to the user's profile. Since this procedure computes a real-valued feature, we estimate the probability distributions using a density estimator. We use the default estimator in the R statistical package [30], which uses multiple Gaussian kernels.

6.2.2 *Accuracy Metrics*

Implicit identifiers are not perfectly identifying. Therefore, to evaluate Question 1, we quantify the *accuracy* of our classifier.

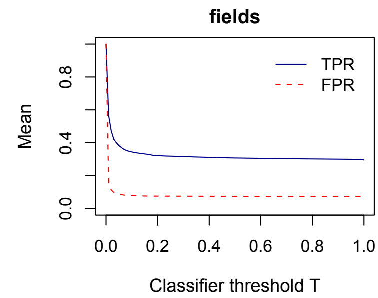


Figure 1—Mean TPR and FPR as the classifier threshold T is varied for fields.

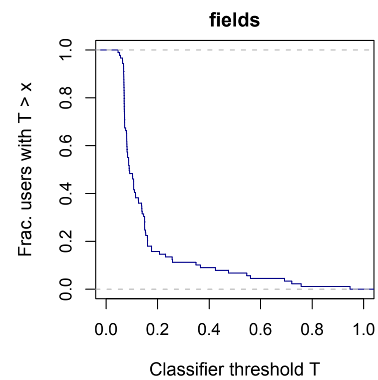


Figure 2—CCDF of classifier thresholds T that achieve FPR = 0.01 for different users

Accuracy has two components: (1) the true positive rate (TPR), or the fraction of validation samples that user U generates that we correctly classify, and (2) the false positive rate (FPR), or the fraction of validation samples that user U does not generate that we incorrectly classify. The former tells us how often U's traffic will identify her, while the later tells us how often we will mistake U as the source of other traffic. We measure accuracy with TPR and FPR instead of just precision (i.e., the fraction of all samples classified correctly) because the vast majority of samples are negative (i.e., not from the target user). Hence, classifiers that mark a larger fraction samples as negative would score higher in precision even if they marked the same fraction of true positives incorrectly.

Trainable Users. When evaluating each identifier, we consider only those users that have at least one training sample that contain it, since we can't build profiles for those with no such samples. Table 3 shows the number of profiled users that exhibit each feature in the training period. Each implicit identifier is exhibited by a different subset of users. In both workloads, each implicit identifier is exhibited by a majority of profiled users. The fraction of users that exhibited the ssids feature is lower in the ucsd workload (55% vs 81%) because fewer users sent SSID probes to search for a network. This may be because many ucsd users already established a high preference for the UCSD network, having used it previously. sigcomm users were all new to the SIGCOMM network and initiated broader searches for their preferred networks before association.

Classifier Thresholds. We evaluate each classifier across several thresholds T in order to determine the trade-off between TPR and FPR. As T increases, FPR and TPR decrease because the classifier

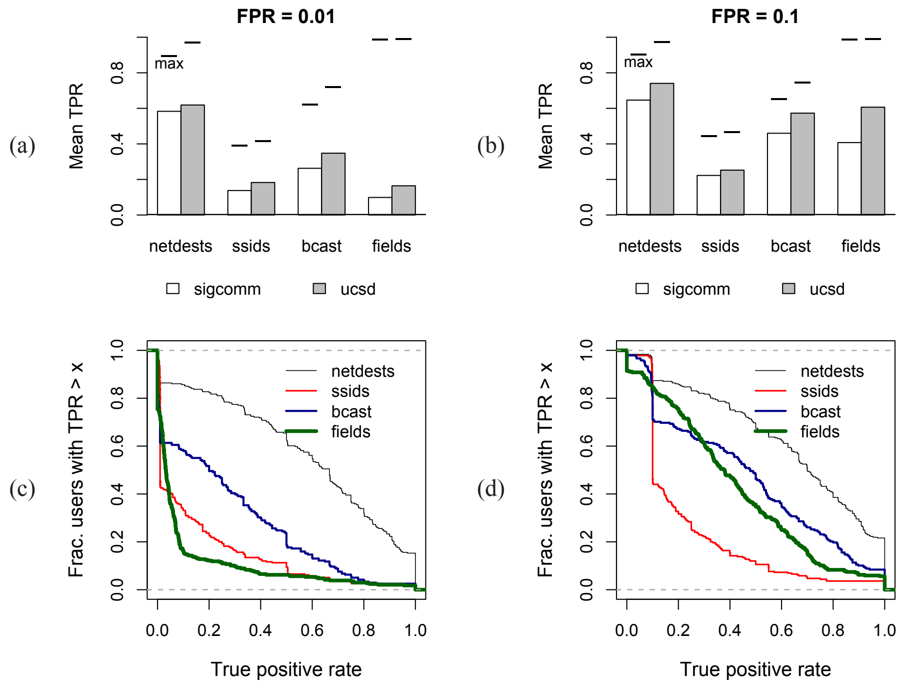


Figure 3—Classification accuracy using each feature. The top two graphs show the mean achieved TPR for (a) FPR = 0.01 and (b) FPR = 0.1. The line above each bar show the maximum expected TPR given a perfect classifier on that feature. The bottom two graphs show a CCDF of the achieved TPR on `sigcomm` users for (c) FPR = 0.01 and (d) FPR = 0.1.

	Fraction of users trainable	
	`sigcomm`	`ucsd`
netdests	0.89	0.84
ssids	0.81	0.55
bcast	0.70	0.65
fields	1.00	1.00

Table 3—The fraction of profiled users that we could train using each feature.

requires more evidence that a user is present in order to answer positively. This is exemplified in Figure 1 for the classifier using the fields feature. We assume that an adversary desires a target FPR, such as 1 in 100, and chooses a threshold T based on that target. Ideally, the target FPR would be low. Due to variance in each user's training data, an adversary may need to use different thresholds to achieve the same FPR for different users. This is exemplified in Figure 2, which shows a complementary cumulative distribution function (CCDF) of thresholds that achieve FPR = 0.01 for each user's classifier using the fields feature. An adversary would train a different classifier for each user that he is tracking. In practice, an adversary would have to select T without a priori knowledge of the FPR achieved on the validation data. In Section 7.1, we show that an adversary can select T to achieve a desired FPR without this knowledge when using multiple features in combination.

6.2.3 Results

In order to examine the characteristics of each individual implicit identifier, we now focus on the TPR achieved for different FPR targets using each identifier in isolation.

Mean Accuracy. Figure 3(a) and (b) shows the mean TPR achievable with each implicit identifier in isolation for FPR = 0.01 and FPR = 0.1, respectively. For example, when using netdests, we can identify samples from the average user in both workloads about 60% of the time for FPR = 0.01. The line above each bar indicates the maximum expected TPR that a perfect classifier would achieve on that implicit identifier—i.e., a classifier that always classifies a sample correctly if it has that implicit identifier, but guesses randomly otherwise. This line is below 1.0 because some validation samples do not contain a particular implicit identifier and, hence, even a perfect classifier on this identifier would not do better than random guessing on those samples. For example, many samples have no SSID probes and, thus, are missing the ssids identifier.

Figure 3(a) shows that the average user sometimes emits an implicit identifier that is highly distinguishing. netdests, ssids, and bcast all achieve moderate TPRs (about 60%, 18%, and 30%, respectively) even for a very low FPR (1%). The lower TPR for ssids is expected, since users usually only emit distinguishing SSIDs when they are searching for a network. Indeed, the theoretical maximum TPR achievable by a perfect classifier is only about 40%. Also, as expected, fields is not able to identify many samples on its own since it only distinguishes wireless cards and drivers.

Figure 3(b) shows that the TPR for fields improves to 40% and 60% when FPR = 0.1, for the `sigcomm` and `ucsd` workloads, respectively. Thus, the fields identifier is good at classifying users into groups, and can aid in identifying users in those cases when no unique identifier is observed. This is expected, since fields only distinguishes wireless cards and divers. The TPR of the other three features improves much less dramatically when we increase the allowable FPR from 0.01 to 0.1. This is because most of the other implicit identifiers either uniquely identify a user, or are not identifying at all. Thus, the TPR gains observed when we increase FPR are mostly due to less conservative random guessing on the remaining samples.

This effect can be seen in Figure 4, which shows the variation in mean TPR and FPR across classification thresholds for `sigcomm`

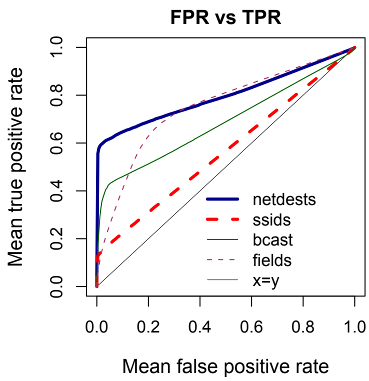


Figure 4—The mean achieved TPR and FPR for `sigcomm` users as we vary the classification threshold T using each feature alone. The $x = y$ line shows how well random guessing would perform.

users. The $x = y$ line shows how well random guessing is expected to perform. The TPR of all the features except for fields grows roughly linearly toward 1.0 after the initial spike, which is the effect that progressively less conservative random guessing would have.

For all features, users in the `ucsd` workload are slightly more identifiable than those in the `sigcomm` trace. This is probably because there are more total users in the `sigcomm` workload and, thus, a higher likelihood that two users exhibit the same traits. We examine the effect population size has on tracking in Section 7.2.

Variation Across Users. Accuracy for some users is better than others. Thus, Figure 3(c) and (d) shows a CCDF of achieved TPR over all users in the `sigcomm` workload, for FPR = 0.01 and FPR = 0.1, respectively. For example, consider netdests when FPR = 0.01. In this case, 65% of users achieve a TPR of at least 50%.

Each of the first three implicit identifiers distinguishes some users very often. Figure 3(c) shows that 65%, 11%, 24% of users have samples that are identified at least half of the time with an FPR of only 0.01 using netdests, ssids, and bcast, respectively. This implies that a non-trivial number of users are trackable even if only one of these features is available.

Nonetheless, when FPR = 0.1, 12%, 53%, and 29% of users have a TPR of at most 0.1 as well using netdests, bcast, and ssids, respectively (see Figure 3(d)). This means that our classifier does not perform any better than random guessing on these users. These users are simply not identifiable. For example, for the netdests feature, this means that these users only visited popular destinations during the training period or did not revisit any site in the subsequent days. This result also implies that the mean TPR shown in Figure 3(a) and (b) actually underestimates the TPR for the users that are identifiable at all, since this fraction of non-identifiable users drags the mean down. We conclude that there is a large variation in user distinctiveness.

Training Sample Sensitivity. To explore the variability in classifier accuracy for different users, we examine whether users observed more often during the training period are more identifiable. Figure 5 shows the mean TPR achieved for FPR = 0.01 for sets of `sigcomm` users with different numbers of training samples. The error bars show 95% confidence intervals, which are negligible for most points.

Figure 5 shows that the mean TPR noticeably increases with more training samples for netdests and bcast. For netdests, TPR stabilizes after 3 training samples. The TPR of ssids and fields does not change dramatically with more training samples, probably because these identifiers are generated without user interaction and, thus, are nearly always identical when emitted. Artifacts near the right hand side of each graph, such as large confidence intervals, are mostly due to small sample sizes for those points. We conclude that an adversary can build a more accurate classifier with more samples, but needs very few to build one that is useful.

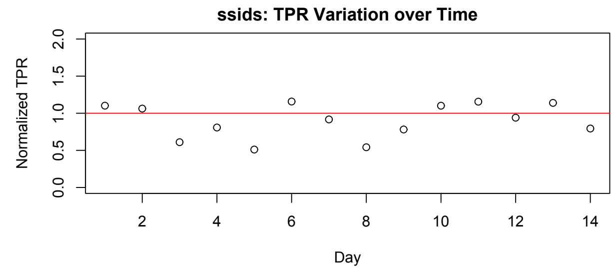


Figure 6—Accuracy over time. Normalized mean TPR on each day in the `apt` trace for FPR = 0.01. Each TPR value is normalized to the mean TPR for the entire period, evaluated over the users present during that day. The mean TPR for the entire period over all profiled users is 42%.

Accuracy Over Time. One concern is that the accuracy of ssids may degrade over time since a user's preferred networks list can change. Figure 6 shows how the mean TPR varies over two weeks in the `apt` trace, the only trace of that duration, fixing FPR = 0.01. Each value is normalized by the mean TPR for the entire period. Even after two weeks, normalized values are close to 1, which suggests that the SSIDs that users emit are relatively stable over time.

7. WHEN CAN WE BE TRACKED?

In this section, we evaluate how accurately an adversary can answer Question 1 and Question 2 in each of the wireless environments described in Section 4. The previous section evaluated how well an adversary could use implicit identifiers independently to determine whether a sample came from a given user, but in practice, an adversary would not be restricted to using identifiers in isolation.

Without link-layer encryption, public networks reveal features both at the link and network layers. In contrast, home networks that employ encryption reveal only link-layer features. Encrypted enterprise networks comprised of homogeneous devices might reveal only link-layer features that vary due to application and user behavior; features that vary due to driver- and card-level differences provide no useful information since they would not vary. Therefore, we evaluate each environment with the following features visible to an adversary:

- Public network: netdests, ssids, fields, bcast.
- Home network: ssids, fields, bcast.
- Enterprise network: ssids, bcast.

Since measurements from these environments can be difficult to obtain due to legal and ethical restrictions, we use our analysis of the `sigcomm` trace to estimate answers to these questions. In all three scenarios, we consider users with devices that will have a different pseudonym each hour of the day as in our analysis in the previous section.

Many users in both the `sigcomm` and `ucsd` traces expose implicit identifiers of all four types, so we conjecture that populations in other environments are unlikely to differ substantially beyond

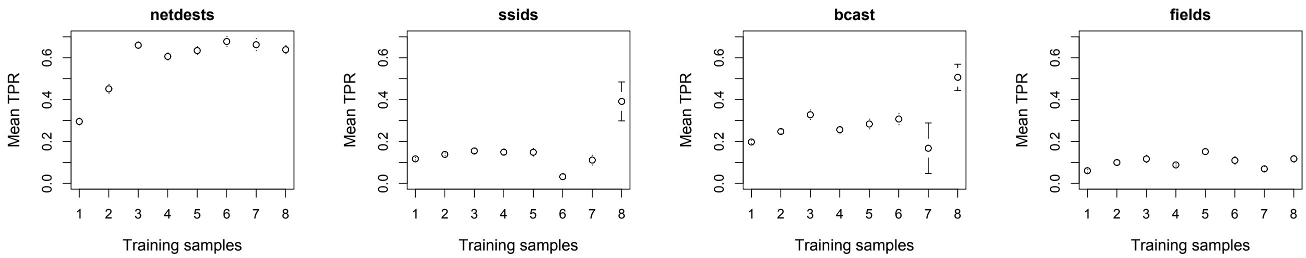


Figure 5—Sensitivity to the number of training samples for each feature. The mean TPR achieved for FPR = 0.01 for `sigcomm` users with different numbers of training samples. The error bars indicate 95% confidence intervals.

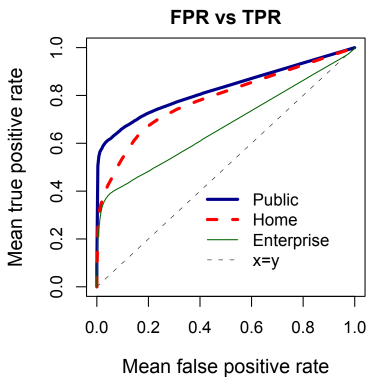


Figure 7—Classification accuracy for Question 1 if `sigcomm` users where in typical public, home, and enterprise networks.

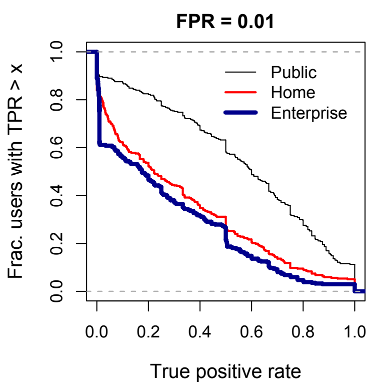


Figure 8—CCDF of TPR for Question 1 if `sigcomm` users were in a typical public, home, or enterprise network for FPR = 0.01.

the identifiers available. The population sizes will differ, however, so we vary the population size in our experiments. Enterprise networks may be more homogeneous, but the identifiers we consider vary due to user behavior and the applications that they run. ssids will remain distinguishing as long as users visit other networks with their devices, and bcast will remain distinguishing as long as laptops run Windows and use or search for different names, since a large number of broadcast packets are due to NetBIOS.

7.1 Q1: Did this Sample come from User U?

First, we evaluate how well an adversary can answer Question 1 using features in combination. Since all profiled users had at least

	% users with FPR error < 0.01	
	median error	90th percentile error
Public	97%	82%
Home	80%	64%
Enterprise	79%	68%

Table 4—Stability of classifier threshold T across different validation sub-samples. The percentage of users that have FPR errors that are less than 0.01 away from the target FPR of 0.01.

one training sample with each of the four features in our training sets, we can evaluate the accuracy on *all* profiled users, not just a fraction, as was the case when using individual features (see Table 3).

Figure 7 shows how accurately we can answer Question 1 for the average user when varying the threshold T in each of our three environments. Figure 8 shows the CCDF of TPR achieved for users in public, home, and enterprise networks for several FPR = 0.01.

When more features are visible, classification accuracy is better. In public networks, user samples are identified 56% of the time with a very low FPR (1%), on average. This TPR is slightly lower than that observed for netdests in Figure 3(a) because here we are considering all users, not only the 89% that exhibited netdests in their training samples. The average TPR in home and enterprise networks is 31% and 26%, respectively, when FPR = 0.01. Figure 8 shows that when FPR = 0.01, 63%, 31%, and 27% of users are identifiable at least 50% of the time in public, home, and enterprise networks, respectively. As expected, users are more identifiable in environments with more features.

Selecting the Classifier Threshold. As mentioned in Section 6.2.2, an adversary would have to select a classifier threshold T to achieve a desired target FPR. In practice, he would have to select the threshold without knowing a priori the resulting FPR of the validation data. Instead, an adversary would have to choose a T that achieves a target FPR in *previous* samples he has collected (e.g., as part of training). Therefore, in order to achieve the desired accuracy, the adversary requires that the T chosen in this manner achieves approximately the FPR target in yet unknown validation data.

To test whether this requirement is met, we ran the following experiment on the `sigcomm` workload: An adversary selects T that achieves FPR = 0.01 on a random 20% subsample of the validation data and tests whether the same T achieves a similar FPR in a different random 20% subsample. We perform 10 trials of this experiment per user and measure the absolute FPR errors, i.e., the difference between the achieved FPR and the target FPR. Table 4 shows the number of users that have median and 90th percentile errors that are less than 0.01 away from the target FPR. 79-97%

of users in all scenarios have errors less than 0.01 away from the target most of the time. This suggests that an adversary would be able to select T that achieves an FPR very close to a desired target in most circumstances.

7.2 Q2: Was User U here today?

Now we consider Question 2. We consider an adversary that wants to accurately detect the presence of a user during a particular 8 hour work day. In this section, we answer the following two questions: (1) How many users can be detected with high confidence? (2) How often does a user have to be active in order to be detected?

7.2.1 Methodology

Accuracy Estimation. Consider an environment with N users present each hour during an eight hour day. User U operates a laptop during $active$ different hours this day and thus an adversary obtains $active$ samples from U. The adversary also obtains up to N samples each hour from the other users.

Suppose an adversary would like to determine whether U is present during this day with a TPR of at least TPR_{target} and an FPR of no more than FPR_{target}. In section 6.2.1, it was shown that an adversary could use features in combination to answer whether a particular traffic sample came from U with a moderate TPR (tpr_{Q1}) and a very low FPR (fpr_{Q1}), on average. Unfortunately, even a very low fpr_{Q1} could result in the misclassification of a sample because during an eight hour day, there would be up to $8N$ opportunities to do so. Therefore, to boost the adversary's accuracy, he could answer Question 2 affirmatively only when multiple samples are classified as being from U.

Specifically, suppose the adversary only answers Question 2 affirmatively when at least one sample from $belief$ different hours is classified as from U. That is, he believes U is present during at least $belief$ different hours. If we assume that the observations made during each hour are independent, when U is active during at least $active \geq belief$ hours,

$$TPR_{target} \geq \Pr[X \geq belief],$$

where X is a binomial random variable with parameters $n = active$ and $p = tpr_{Q1}$. In addition,

$$FPR_{target} \leq \Pr[Y \geq belief],$$

where Y is a binomial random variable with parameters $n = 8$ and $p \leq 1 - (1 - fpr_{Q1})^N$, the probability that at least 1 sample not from U during one hour is misclassified. We show below that the independence assumption is not unreasonable.

In order for an adversary to answer Question 2 with TPR_{target} and FPR_{target}, he would determine if there exists a threshold T for U's classifier that would satisfy these constraints. In the process, he would also determine the minimum number of hours that U would have to be active ($active$). For example, when all four features are available, we show that quite a few users can be detected when they are active for several hours even if an adversary desires 99% accuracy (i.e., $TPR_{target} \geq$ 99% and $FPR_{target} \leq$ 1%).

Dependence. The constraints above assume that the observations made during each hour are independent. That is, the likelihood of observing a true or false positive is not dependent on the adversary's past observations. The following analysis of the `sigcomm` trace shows that there is some dependence in reality, but that the dependence is small.

There are two primary concerns. The first concern is that our classifier may often confuse user U with another user Q, so that if

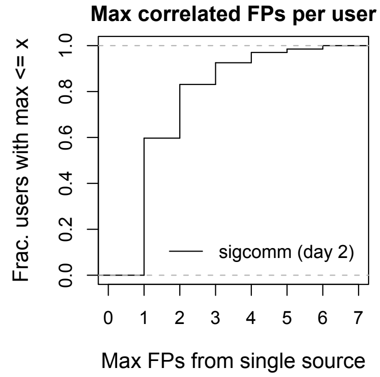


Figure 9—Limited dependence in the `sigcomm` trace. CDF of the maximum number of false positives (FPs) generated by any one user for each user.

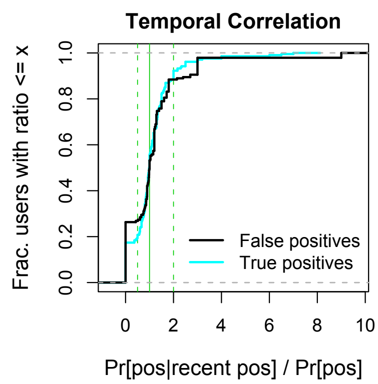
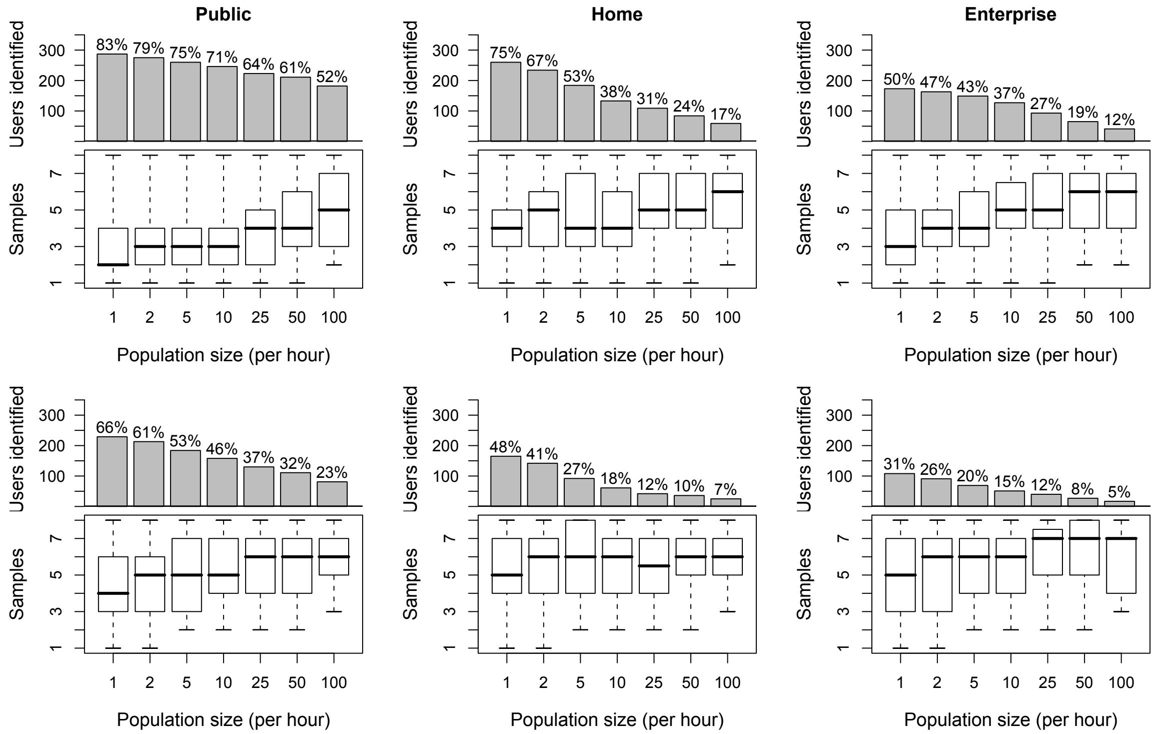


Figure 10—Limited dependence in the `sigcomm` trace. CDF of how much more likely a true or false positive is given that one was observed recently.

Q is active, then the false positive rate will be high regardless of the number of hours that the adversary samples. This concern is mitigated by two factors that add randomness to the sampling process: 1) users enter and depart from the environment and 2) user behavior is variable to begin with. Consider our classifier on all features using a classification threshold $T = 0.5$. Figure 9 shows, for each user that exhibits any false positives during the second full day of the `sigcomm` trace, the maximum number of false positives that are contributed by any other single user. From this cumulative distribution function (CDF), we see that for 60% of users, no single other user is responsible for more than 1 false positive, and for over 95%, no single user is responsible for more than 3 false positives. Therefore, most of the time the two factors mentioned prevent a large number of false positives from being correlated to a single user. In addition, since the user set is relatively static at a conference, there is likely to be more churn in the population of most other environments, further reducing the dependence.

The second concern is that there may be temporal locality in either true or false positive samples. For example, we might expect that a user is much more likely to exhibit a particular feature if he has done so in the recent past. If temporal correlation was substantial then the ratio

$$\frac{\Pr[\text{positive} \mid \text{positive in the last } t \text{ hours}]}{\Pr[\text{positive}]}$$



(a) 90%

(b) 99%

Figure 11— The number of of users detectable and the number of hours they must be active to be detected with (a) 90% accuracy and (b) 99% accuracy. The x-axis in each graph varies the population size. The top portion shows the number and percentage of users it is possible to detect. The bottom portion shows a box plot of the number of hours during which they must be active to be detected. That is, the thick line through the middle of each box indicates the median, the ends of each box demark the middle 50% of the distribution, and the whiskers indicate the minimum and maximum values.

would be much larger or smaller than 1. Figure 10 shows a CDF of this ratio for each users' true and false positives when $t = 2$ using the same classifier as above. For true positives, we only consider times during which the user is active. For false positives, we only consider the active 9 hours of the last 2 days of the conference since false positives are obviously less likely to occur when fewer people are present. If there was no temporal correlation, we would obtain a vertical line at $x = 1$. We note that 60 and 70% of users' true and false positives are within a factor of 2 of this line, meaning that if a true (false) positive was seen in the last two hours we are no more than 2 times more or less likely to observe another true positive than otherwise. Moreover, given the small number of positives for each user, much of this variation is probably due to randomness. Therefore, temporal dependence is small.

7.2.2 Results

Figure 11 shows the number of users detectable and the number of hours they must be active to be detected with (a) 90% accuracy, (b) 99% accuracy. The x-axis in each graph varies the number of users present each hour. The top half of each graph shows the number of users an adversary can detect and, above each bar, the percentage of profiled users the number represents. The bottom half of each graph shows a box plot of the number of hours during which these users must be active to be detected. That is, the thick line within each box shows the median number of hours a detectable user has to be active to be detected, while the ends of each box demark the first and third quartiles. The whiskers mark the minimum and maximum.

For example, part (a) shows the results if the adversary desires an accuracy of 90% (i.e., $TPR_{target} \geq$ 90% and $FPR_{target} \leq$ 10%). Consider the public networks figure. The fourth bar from the left in top part shows that when there are 10 users present per hour, we can detect 71% of users if they are active during all 8 hours when present. The box and whiskers just below that in the bottom part shows that shows that most of these users do not need to be active all 8 hours to be detected. Of the 71% of users that can be detected, 75% of them only need to be active for 4 hours to be detected, 50% for at most 3 hours, and 25% for at most 2 hours.

Conclusions. We make two overall conclusions. First, an adversary can successfully combine multiple implicit identifiers from a few samples to detect many users in common networks with high accuracy. The majority of users can be detected with 90% accuracy when active often enough in public networks with 100 concurrent users or less. At least 27% of users are detectable with 90% accuracy in all of the networks when there are 25 concurrent users or less. This implies that many users can be detected with high confidence in small to medium sized networks regardless of type if they are active often enough. Even in large networks with 100 users, 12% to 52% remain detectable.

Second, some users are detectable with very high accuracy. Even if an adversary desires 99% accuracy, the fraction of detectable users is between 12% and 37% in all networks with 25 users when they are active often enough. Therefore, even applying existing best network security practices will fail to protect the anonymity of a non-trivial fraction of users.

Indeed, several usage patterns in home and enterprise networks make detection more likely than the overall results suggest. In home networks, very few users are likely to be active during each hour. For example, even when monitoring all the networks in our apt trace, we only observed 4 users per hour, on average. Therefore, the results closer to the left side of each graph are more representative of home environments. Since users of a enterprise network are probably employees, they are more likely to be active for the entire observation period. Thus, the top half of each graph is probably a good estimation of the fraction of users that an adversary can detect on a typical day.

8. CONCLUSIONS AND FUTURE WORK

In this paper we demonstrated that users can be tracked using *implicit identifiers*, traffic characteristics that remain even when unique addresses and names are removed. Although we found that our technique's ability to identify users is not uniform—some users do not display any characteristics that distinguish themselves from others—most users can be accurately tracked. For example, the majority of users can be tracked with 90% accuracy when active often enough in public networks with 100 concurrent users or less. Some users can be tracked with even higher accuracy. Therefore, pseudonyms are insufficient to provide location privacy for many users in 802.11 networks.

Moreover, our results showed that even a single implicit identifier, such as netdests, ssids, or bcast, can be highly discriminating and that an adversary needs only 1 to 3 samples of users' traffic to track them successfully, on average. Therefore, we argue that addressing all the shortcomings outlined in Section 2 is critical to improving the anonymity of wireless protocols. We are designing mechanisms to resolve these issues in 802.11.

Finally, we note that by considering a subset of all possible implicit identifiers and a weak, passive adversary, our results only place a lower bound on the accuracy with which users can be tracked. We are continuing our effort to uncover implicit identifiers exposed in 802.11, such as those exposed by timing channels. In addition, we would like to evaluate the accuracy of our implicit identifiers over longer time scales and across different locations, since this study's analysis is limited by the duration and location of our traces.

9. ACKNOWLEDGEMENTS

We thank David Andersen, Tadayoshi Kohno, and the anonymous reviewers for their valuable comments and suggestions. We thank CRAWDAD [31] for making the sigcomm trace available and Yu-Chung Cheng for providing us with the ucsd trace.

10. REFERENCES

[1] Big boss is watching. CNET News.com, Sept. 2004. http://news.com.com/Big+boss+is+watching/2100-1036_3-5379953.html.
[2] Wireless location tracking draws privacy questions. CNET News.com, May 2006. http://news.com.com/Wireless+location+tracking+draws+privacy+questions/2100-1028_3-6072992.html.
[3] Wi-fi hacking, with a handheld PDA. ZDNet, Feb. 2007. http://blogs.zdnet.com/security/?p=19.
[4] ABADI, M., AND FOURNET, C. Private authentication. *Theor. Comput. Sci. 322*, 3 (2004).
[5] AKELLA, A., JUDD, G., SESHAN, S., AND STEENKISTE, P. Self-management in chaotic wireless deployments. In *MobiCom* (2005).
[6] BAHL, P., AND PADMANABHAN, V. N. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM* (2000).
[7] BERESFORD, A. R., AND STAJANO, F. Location privacy in pervasive computing. *IEEE Pervasive Computing 2*, 1 (2003).
[8] BISSIAS, G., LIBERATORE, M., JENSEN, D., AND LEVINE, B. N. Privacy vulnerabilities in encrypted http streams. In *Proc. Privacy Enhancing Technologies Workshop* (2005).
[9] BRESLAU, L., CAO, P., FAN, L., PHILLIPS, G., AND SHENKER, S. Web caching and Zipf-like distributions: Evidence and implications. In *INFOCOM* (1999).
[10] CHENG, Y.-C., BELLARDO, J., BENKO;, P., SNOEREN, A. C., VOELKER, G. M., AND SAVAGE, S. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. *SIGCOMM CCR 36*, 4 (2006).
[11] FRANKLIN, J., MCCOY, D., TABRIZ, P., NEAGOE, V., RANDWYK, J. V., AND SICKER, D. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proc. USENIX Security Symposium* (2006).
[12] FYODOR. Nmap network security scanner. http://insecure.org/nmap/.
[13] GARFINKEL, S., JUELS, A., AND PAPPU, R. RFID privacy: An overview of problems and proposed solutions. *IEEE Security and Privacy 3*, 3 (2005).
[14] GRUTESER, M., AND GRUNWALD, D. Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitative analysis. *ACM Mobile Networks and Applications 10*, 3 (2005).
[15] HAVERINEN, H., AND SALOWEY, J. Extensible authentication protocol method for global system for mobile communications (GSM) subscriber identity modules (EAP-SIM), 2006. IETF RFC 4186. http://www.ietf.org/rfc/rfc4186.txt.
[16] IEEE 802.11i-2004 amendment to IEEE std 802.11, 2004.
[17] IETF geographic location/privacy working group charter. http://www.ietf.org/html.charters/geopriv-charter.html.
[18] JIANG, T., WANG, H., AND HU, Y.-C. Preserving location privacy in wireless LANs. In *Proc. ACM MobiSys* (2007).
[19] KARAGIANNIS, T., BROIDO, A., FALOUTSOS, M., AND CLAFFY, K. Transport layer identification of P2P traffic. In *IMC* (2004).
[20] KARAGIANNIS, T., PAPAGIANNAKI, K., AND FALOUTSOS, M. BLINC: multilevel traffic classification in the dark. *SIGCOMM CCR 35*, 4 (2005).
[21] KOHNO, T., BROIDO, A., AND CLAFFY, K. C. Remote physical device fingerprinting. In *Proc. IEEE Symposium on Security and Privacy* (2005).
[22] Location privacy protection act of 2001. U.S. Senate bill.
[23] MICROSOFT. Wireless Client Update for Windows XP with service pack 2, http://support.microsoft.com/kb/917021.
[24] MONROSE, F., AND RUBIN, A. Authentication via keystroke dynamics. In *Proc. ACM Computer and communications security* (1997).
[25] MOORE, A. W., AND ZUEV, D. Internet traffic classification using bayesian analysis techniques. In *SIGMETRICS* (2005).
[26] MORRIS, R. http://pdos.csail.mit.edu/~rtm/.
[27] MUNIWIRELESS. http://www.muniwireless.com/.
[28] p0f. http://freshmeat.net/projects/p0f/.
[29] PADMANABHAN, B., AND YANG, Y. Clickprints on the web: Are there signatures in web browsing data? http://knowledge.wharton.upenn.edu/papers/1323.pdf, 2006.
[30] R DEVELOPMENT CORE TEAM. *R: A Language and Environment for Statistical Computing*, 2006. http://www.R-project.org.
[31] RODRIG, M., REIS, C., MAHAJAN, R., WETHERALL, D., ZAHORJAN, J., AND LAZOWSKA, E. CRAWDAD data set uw/sigcomm2004 (v. 2006-10-17). http://crawdad.cs.dartmouth.edu/meta.php?name=uw/sigcomm2004, Oct. 2006.
[32] Roofnet. http://pdos.csail.mit.edu/roofnet/.
[33] SONG, D. X., WAGNER, D., AND TIAN, X. Timing analysis of keystrokes and timing attacks on ssh. In *USENIX Security Symposium* (2001).
[34] SUN, Q., SIMON, D. R., WANG, Y.-M., RUSSELL, W., PADMANABHAN, V. N., AND QIU, L. Statistical identification of encrypted web browsing traffic. In *Proc. IEEE Symposium on Security and Privacy* (2002).
[35] TAO, P., RUDYS, A., LADD, A. M., AND WALLACH, D. S. Wireless LAN location-sensing for security applications. In *WiSE* (2003).
[36] TAUSCHER, L., AND GREENBERG, S. How people revisit web pages: empirical findings and implications for the design of history systems. *Int. J. Hum.-Comput. Stud. 47*, 1 (1997).
[37] tcpdump. http://www.tcpdump.org/.
[38] VAN RIJSBERGEN, C. *Information Retrieval*. Butterworths, 1979.
[39] WIGLE. http://www.wigle.net/.
[40] Wireless privacy protection act of 2005. U.S. House bill.
[41] WONG, F.-L., AND STAJANO, F. Location privacy in bluetooth. In *ESAS* (2005).
[42] WRIGHT, C., MONROSE, F., AND MASSON, G. On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research* (Aug. 2006).
[43] ZANDER, S., NGUYEN, T., AND ARMITAGE, G. Automated traffic classification and application identification using machine learning. In *LCN* (2005).