

# LAPWiN: Location-Aided Probing for Protecting User Privacy in Wi-Fi Networks

Yu Seung Kim, Yuan Tian, Le T. Nguyen, and Patrick Tague  
Carnegie Mellon University  
Email: {yuseungk, yt, lenguyen, tague}@cmu.edu

**Abstract**—Efficient Wi-Fi probing has been demonstrated to leak sensitive user information. During the probing process, Wi-Fi clients transmit the names of previously known Wi-Fi access points (APs) in plaintext. An eavesdropper can easily collect the information leaked by this Wi-Fi probing process to mount numerous attacks, such as fake AP or revealing hidden APs, or to breach users’ privacy. Since APs are often named after the location, business, or affiliation of the host, an attacker can learn about nearby users and infer social connections. In this work, we propose to reduce the privacy risk while simultaneously decreasing the network connection time by eliminating unnecessary probe requests, most notably those requests sent to networks that are not in proximity of the device. We present a location-aided Wi-Fi probing mechanisms called LAPWiN to achieve these improvements. We demonstrate how LAPWiN can be implemented by modifying a widely used network manager and evaluate the performance and achievable privacy gains.

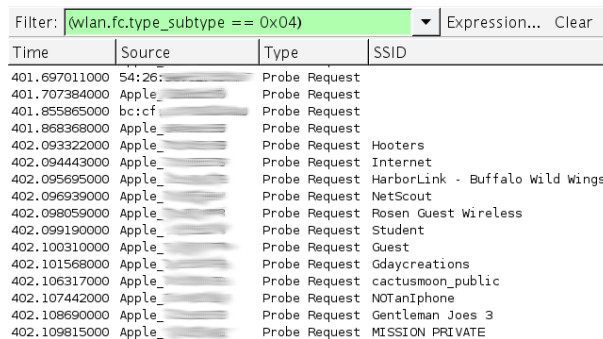
## I. INTRODUCTION

Diverse services have recently emerged over widely deployed Wi-Fi networks. In addition to standard Internet connectivity, Wi-Fi networks can, for example, assist with device localization, either in conjunction with GPS or independently using a location service provider [1]. Similarly, the Wi-Fi infrastructure can collect information from mobile devices to enable tracking and provide location-based targeted marketing to retailers or surveillance capabilities to building managers [2]. Most of these services rely on control frames recorded from Wi-Fi devices, exchanged for the purpose of initiating connections quickly and efficiently. Although varying by vendor implementation, many modern Wi-Fi-enabled mobile devices emit these control frames even in sleep mode, unbeknownst to users. Most of these control frames fall under the category of management frames, as defined in the Wi-Fi standard [3]. Moreover, since these frames are used in the initial stage of network association, they are not encrypted by management frame protection [4].

Based on the availability of Wi-Fi control frames to nearby listeners during the network association procedure, researchers have recently identified a variety of possible threats to users’ personal privacy [5], [6], [7]. In particular, during the process of active scanning, preferred by most mobile devices due to relatively short connection time, the Wi-Fi device will broadcast probe request frames to the set of preferred networks. Each probe request frame includes the service set identification (SSID) of a previously associated access point (AP), along with the MAC address of the device and other information, in order to elicit a probe response from the target AP. While this active probing mechanism incurs an energy cost to broadcast

the probe requests, it drastically reduces the connection time by increasing the scan speed. In contrast, the passive scanning mechanism requires no transmission, but the device must listen for a period of time on each Wi-Fi channel in an attempt to receive beacon frames occasionally sent by the APs.

Due to the lack of frame protection of Wi-Fi probing messages, an eavesdropper can easily recover a list of device MAC addresses and the SSIDs included in probe requests. This capability enables numerous attacks. For example, since many Wi-Fi connection managers check only the SSID of a candidate AP to connect with, an attacker can easily set up a fake AP using an SSID that the device is seeking [5]. Observing this SSID information in the probe request frame is also an easy way for an attacker to reveal hidden APs which do not publicize their SSIDs. In this work, we focus on the fact that SSIDs broadcast by mobile devices contain human-understandable information, including location names and context details, enabling an attacker to learn private details about the users of the devices and breach personal privacy. We illustrate an example of this type of information in Fig. 1 for data collected at a security conference.



Time	Source	Type	SSID
401.697011000	S4:26:...	Probe Request	
401.707384000	Apple_...	Probe Request	
401.855865000	bc:cf:...	Probe Request	
401.868368000	Apple_...	Probe Request	
402.093322000	Apple_...	Probe Request	Hooters
402.094443000	Apple_...	Probe Request	Internet
402.095695000	Apple_...	Probe Request	HarborLink - Buffalo Wild Wings
402.096939000	Apple_...	Probe Request	NetScout
402.098059000	Apple_...	Probe Request	Rosen Guest Wireless
402.099190000	Apple_...	Probe Request	Student
402.100310000	Apple_...	Probe Request	Guest
402.101568000	Apple_...	Probe Request	Gdaycreations
402.106317000	Apple_...	Probe Request	cactusmoon_public
402.107442000	Apple_...	Probe Request	NOTanIphone
402.108690000	Apple_...	Probe Request	Gentleman Joes 3
402.109815000	Apple_...	Probe Request	MISSION PRIVATE

Fig. 1: Frames captured at a security conference reveal the SSIDs of networks previously used by nearby users, enabling an attacker to infer users’ past behaviors.

While previous research has demonstrated several vulnerabilities due to the use of Wi-Fi probing, we propose to address the issue by changing the behavior of the probing mechanism that creates the vulnerabilities in the first place. Intuitively, the value of active probing is in broadcasting the SSIDs of networks that the devices prefers to connect to. However, if the device broadcasts an SSID for a network that is not present near the device, this value is lost, so the only outcome is revealing the users’ preference. If we can eliminate the unnecessary SSID broadcasts without significantly affecting

connection times, the privacy risk can be reduced. Moreover, in reducing the unnecessary broadcasts, it may be possible to even *decrease the connection time while simultaneously improving privacy*. Toward this goal, we propose *location-aided probing in Wi-Fi networks*, or *LAPWiN*, to prevent these unnecessary broadcasts. Using LAPWiN, the device sends probe requests only for APs which are both known and geographically nearby, thus minimizing exposure of information about previously connected APs from the user’s device, enhancing both the **privacy** and **efficiency** of Wi-Fi connection management.

Our proposed LAPWiN mechanism is also practically deployable because it requires modification only on the Wi-Fi client side and supports Wi-Fi devices with and without explicit positioning capabilities. Our efforts in this paper are summarized in the following contributions.

- We propose a novel location-aided probing mechanism in Wi-Fi networks, providing improved privacy, better performance, and practical deployment.
- We implement LAPWiN for proof-of-concept by modifying a widely used open source network connection manager in Linux-based platforms and evaluate its performance via analysis and practical testing.

The remainder of this paper is structured as follows. In Section II, we detail the models and background information used in our study of Wi-Fi probing schemes. We propose our LAPWiN mechanism in Section III and explain its operation in various situations. In Section IV, we describe the implementation of LAPWiN and evaluate its performance via experiments. In Section V, we summarize related studies on privacy issues from Wi-Fi probing and early defenses. Finally, we conclude the paper in Section VI.

## II. WI-FI PROBING AND THREATS

Wi-Fi probing process is designed to provide user with a fast and convenient method to connect to Wi-Fi APs, and user privacy is not taken as first priority in the standard. We first render the assumptions used in the problem of our interest and background of common Wi-Fi probing techniques. We then analyze the vulnerabilities of current Wi-Fi probing techniques in terms of user privacy and detail the attacker’s model.

### A. Model Assumptions and Definitions

Throughout this paper, we assume that Wi-Fi devices operate in the **infrastructure mode** (*i.e.*, Wi-Fi clients are connected to AP) for ease of discussion. However, our approach can be easily extended without loss of generality, since the probing procedure is similarly used in distributed types of Wi-Fi modes such as ad hoc mode or mesh mode. **Wi-Fi APs less likely change their geographical location**, but network administrators can sometimes move them for the management purpose. In contrast to fixed Wi-Fi APs, **Wi-Fi clients are mobile**. Mobile Wi-Fi clients such as smartphones and tablets frequently switch between different APs as they move around, while the nomadic type of devices such as laptops occasionally move to another location. **Wi-Fi clients may have positioning capabilities or not**. Rapid advance in location technology enables various location sources such

as GPS, cellular communication, and even location service providers rampant and available. Still, however, there are many legacy Wi-Fi devices without these means.

In the IEEE 802.11 WLAN standard [3], Wi-Fi probing is divided into two types: **active scan** and **passive scan**. Active scan again has two different modes, **direct probe** and **broadcast probe**. A Wi-Fi client using the former broadcasts probe request frames containing the SSID of APs to which it has ever connected before. The previously associated SSIDs are usually stored in the local storage of Wi-Fi client and are updated whenever a Wi-Fi client connects with new APs. A Wi-Fi station waits during a certain period to collect the probe response frames responded by surrounding APs, and then switches to the next frequency channel. Broadcast probe (aka wildcard probe) carries the empty SSID field in probe requests. The APs receiving broadcast probe respond with probe response, and thus broadcast probe is used for finding new APs as well as previously connected APs. Note that many APs also have an option to deny to respond to the broadcast probe for security purpose. Owing to their benefits, it is common to use hybrid approaches combining both in many implementations. Also, in many cases the Wi-Fi probing procedure occurs not only in the initial connection phase, but also in the associated phase, in order to prepare alternative Wi-Fi networks for the situation where the current Wi-Fi network becomes unavailable. In contrast to active scan, passive scan non-invasively waits during a certain period to listen to the periodic beacons from surrounding APs. Since a Wi-Fi client should wait enough to listen to each channel, passive scan is much slower than other scan methods, thus not preferred by many implementations.

### B. Threat Model

Although the direct probe is preferred due to its fast scanning speed, it can pose serious privacy breach by revealing the SSID list of user’s Wi-Fi devices [8]. As illustrated in Fig. 2a, the Wi-Fi client  $C$  is direct probing to search the previously connected APs by sending out the SSID list in plain text with its MAC address. In this case, an attacker  $E$  can passively eavesdrop the  $C$ ’s whole SSID list, not only the SSID ‘SFO-WiFi’ of nearby AP  $A$ . It is possible for an attacker to use this information for launching the *Karma* attack [5] or revealing the SSIDs of hidden APs. However, in this study, we focus on privacy breach from the conventional direct probe.

1) *User identification*: The SSID information captured by an attacker can be used to identify the Wi-Fi device user. The small clue about a target user can be combined with the captured SSID information to track a user. Suppose that an attacker knows that the target user is a professor in *UW* University, a frequent visitor of *Hooters*, and his hometown is *Zurich*. If an attacker observes that a Wi-Fi device sends probe requests with the SSIDs such as *UWNetwork*, *Hooters*, and *ZurichAirport* as shown in Fig. 2b, it should be easily guessed as the signal from the target user’s.

We introduce a metric to evaluate the traceability (distinguishability) in this situation. Let us first denote the union set of SSID lists of all Wi-Fi devices as  $\mathbb{S}$  and the set of Wi-Fi devices as  $\mathbb{D}$  in a given geographical area. The Wi-Fi device  $d \in \mathbb{D}$  sends the probe requests containing the SSID list

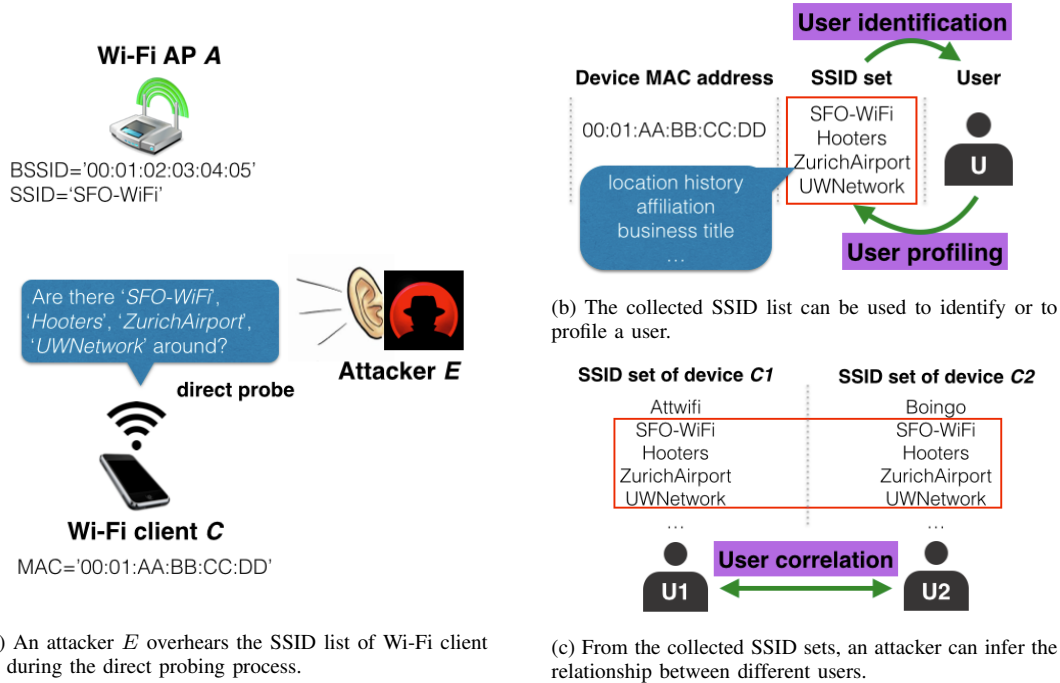


Fig. 2: Illustrated are the network model of conventional Wi-Fi probing and potential risks in the process.

$s_d \in \mathbb{S}$ . Note that there is no duplicated SSID in  $s_d, \forall d \in \mathbb{D}$ . Let us also denote the probability that an attacker randomly selects the SSID list  $s_d \subset \mathbb{S}$  as  $\mathcal{O}(s_d)$  (i.e., the occurrences in the total set). An attacker observing the SSID list  $s_d$  of  $d$  can exactly pinpoint  $d$  among all devices in  $\mathbb{D}$  if  $s_d$  is unique in  $\mathbb{S}$  (i.e.,  $\mathcal{O}(s_d) = 1/|\mathbb{D}|$ ). Conversely, if all devices in  $\mathbb{D}$  have the same SSID list (i.e.,  $\mathcal{O}(s_d) = 1, \forall d \in \mathbb{D}$ ), the probability that an attacker can detect the target device by observing the SSID list will be  $1/|\mathbb{D}|$  (i.e., random guess).

Using the definition of entropy in Information Theory [9], we represent the uniqueness of SSID sets as

$$\mathcal{H}(\mathcal{O}) = - \sum_{s_d \subset \mathbb{S}, d \in \mathbb{D}} \mathcal{O}(s_d) \log_2 \mathcal{O}(s_d). \quad (1)$$

The uniqueness of SSID sets is inversely proportional to the privacy level of Wi-Fi devices. The lower bound of  $\mathcal{H}(\mathcal{O})$  is 0 (perfect privacy), and the upper bound of  $\mathcal{H}(\mathcal{O})$  is  $\log_2 |\mathbb{D}|$  (no privacy). We aim to develop a defense decreasing this uniqueness of SSID sets in Wi-Fi probing.

2) *User profiling*: The SSID itself carries abundant meaningful information related to user's privacy. Many SSIDs are generally named after the current location, the business title, and so on. The name of business included in the SSID can be sometimes critical to others. The cautious might not want that their preferences inferred from frequently visiting businesses are exposed to others without their intention (e.g., even for the targeted marketing purpose). It is demanding to manually manage the SSID information stored in their devices. Even worse, some mobile platforms do not provide a way to manually flush the SSID list in itself.

An attacker thus can exploit the SSID information of Wi-Fi users to collect more critical user privacy. Of course, the significance level of each SSID varies with the context. Ignoring the semantics of SSIDs for ease of analysis, we can directly use the number of SSIDs broadcast from a Wi-Fi client device to measure the privacy level against user profiling.

3) *User correlation*: An attacker is also able to correlate the SSID information to infer the relationship between different users or devices. For example, it is possible to find other Wi-Fi devices which belong to a user or other users sharing the similar Wi-Fi usage pattern (e.g., family members, coworkers, roommates, etc.). We can define the *correlated pair of devices*, if the devices  $d_i$  and  $d_j, \forall i, j \in \mathbb{D}$  satisfy

$$|s_{d_i} \cap s_{d_j}| \geq \alpha, \quad (2)$$

where  $|\cdot|$  is the cardinality, and  $\alpha$  is the correlation threshold, which is defined as a constant or decided with the size of SSID lists  $s_{d_i}$  and  $s_{d_j}$ . For instance, if we assume most users use one Wi-Fi network at their home and another at their workplace,  $\alpha$  is set to  $\alpha = 2$ . For the users traveling many places and using their devices,  $\alpha$  can be defined as

$$\alpha = \beta \cdot \min(|s_{d_i}|, |s_{d_j}|), 0 < \beta < 1, \quad (3)$$

where  $\beta$  is the relative correlation coefficient varying with the number of commonplace SSIDs and so on.

### III. PREVENTING PROBING LEAKAGE

In this section, we propose a novel location-aided Wi-Fi probing mechanism, referred to as *LAPWiN*, and discuss how it addresses the issues in previous works. The design goals of *LAPWiN* are summarized as follows.

- LAPWiN should not (or maximally reduce to) hamper the usability of Wi-Fi devices. The usability includes the successful connection rate to an AP, the average connection speed, the recovery to the original scheme when it fails, and so on.
- LAPWiN should be practically deployable. It should at best support the legacy devices. It should neither require special hardware or significant changes in existing protocols.

#### A. Location-Aided Wi-Fi Probing

So as to prevent the attackers exploiting information leaked from the existing Wi-Fi probing mechanism, it requires to minimize the exposure of SSID from users' Wi-Fi devices. If a Wi-Fi client knows the context about the APs to connect to, it does not need to probe with all APs in its local storage. As such a context, we use the *locations* of Wi-Fi clients and Wi-Fi APs to be found. Fig. 3 illustrates the idea of location-aided Wi-Fi probing. Wi-Fi clients using LAPWiN filters the list of SSIDs of nearby APs by proximity testing. Understanding the current location context, they attempt to probe only the nearby APs, thus not revealing its whole SSID list to an attacker. Moreover, all nearby Wi-Fi clients will send the similar SSID list so that an attacker cannot easily identify an individual user.

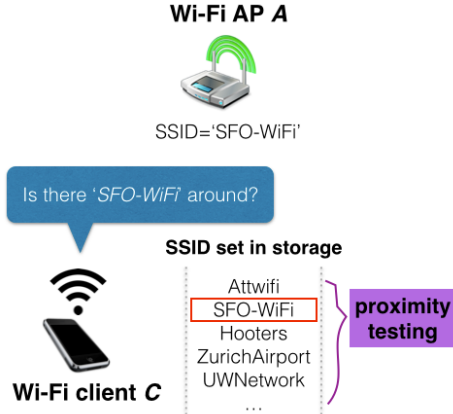


Fig. 3: A Wi-Fi client understanding the current location context can probe with only the SSIDs of nearby APs, thus not exposing the whole SSID list.

Since Wi-Fi APs generally does not provide its location information to the clients, a Wi-Fi client using LAPWiN should log its current location with the associated AP and reuse it for the proximity testing of next visit. Many mobile Wi-Fi devices can utilize a variety of positioning methods such as GPS, cellular network, and even accelerometer. Also, many applications running on these devices periodically or interactively update the current location of device. LAPWiN can thus use the last known location or update the location if it is too old to be used.

Fig. 4 illustrates the proximity testing of LAPWiN with a 2D map. We denote the location of the Wi-Fi AP  $A$  as  $\mathcal{L}(A)$ , the location of the Wi-Fi client  $C$  at time  $t$  as  $\mathcal{L}(C;t)$ , and the location uncertainty of  $C$  as  $a_{C;t}$ .<sup>1</sup> The Wi-Fi client  $C$  connects

<sup>1</sup>The location uncertainty is reported with the location coordinate by API in most mobile platforms. It is represented as distance unit.

to the Wi-Fi AP  $A$  at the time  $t_1$ . In an ideal case, the wireless coverage boundary of  $A$  is represented as the circle  $l_0$  with the radius  $c$ . While  $C$  knows that it is located in the circle  $l_1$ , of which the center is at  $\mathcal{L}(C;t_1)$  with the radius  $a_{C;t_1}$ , it does not know  $\mathcal{L}(A)$ . Thus,  $C$  can only estimate that  $\mathcal{L}(A)$  is within the circle  $l_2$ , which has a center at  $\mathcal{L}(C;t_1)$  with the radius  $a_{C;t_1} + c$ . We can define a wireless coverage of  $A$  as  $l_3$ , which has a center at  $\mathcal{L}(C;t_1)$  with the radius less than  $2 \cdot (c + a_{max})$ , where  $a_{max}$  is the maximum location uncertainty.

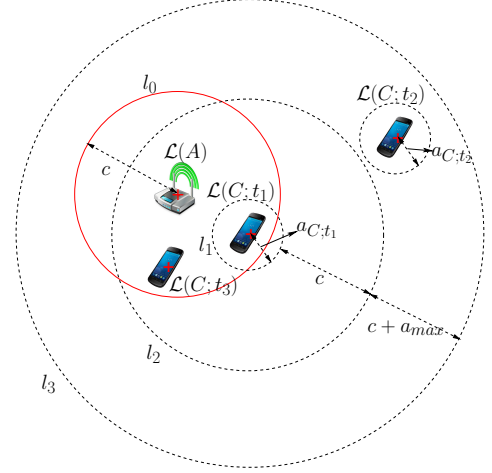


Fig. 4: The operation of proximity testing in LAPWiN is depicted with a 2D map. The positioning capable Wi-Fi client  $C$  estimates  $l_3$  as the wireless coverage of the AP  $A$ . The red solid circles represent the real wireless coverages of APs, while the black dotted circles represent estimated location or wireless coverage.

Whenever  $C$  revisits this area and determines that it is located in  $l_3$ , it will probe with the SSID of  $A$ . If  $C$  is located at  $\mathcal{L}(C;t_3)$  at  $t_3$ , it will probe with the SSID of  $A$  and successfully connect to  $A$ . In this case, LAPWiN should update the SSID entry of  $A$  with the current location  $\mathcal{L}(C;t_3)$  for the next connection. Since the locations at each connection can be different, the new location can be added into the existing locations or can replace the previous one. The former method can be used for improving accuracy of estimating AP location by using multiple locations, while it requires additional computation and storage, thereby inappropriate in resource constrained Wi-Fi devices.

However, at  $\mathcal{L}(C;t_2)$ ,  $C$  will probe with the SSID of  $A$  although it cannot connect to it. We define this case as *false positive*, since  $C$  determines that it is close enough to  $A$  to send the SSID of  $A$  although it is not. In contrast, *false negative* is defined as the case the Wi-Fi client should probe with the SSID of  $A$  since it is located inside the coverage of the Wi-Fi AP, but it does not try to probe.

To fulfill the usability requirement in our design goal, we aim to minimize false negatives. In real practice, the wireless coverage of  $A$  cannot be shown as a regular circle due to wireless fading, irregular radiation pattern of antennas, etc. Thus,  $c$  should be large enough to make the estimated wireless coverage of  $A$  cover the real wireless coverage of  $A$ . While it will result in increasing false positive, an attacker cannot collect the entire SSID list of the target user's device unless

all APs specified in the SSID list are located near to an attacker.

On the other hand, it is not guaranteed that the Wi-Fi device always obtains the current location. The positioning capability may not work when the device cannot reach any location reference sources, or it may require too high energy consumption in an embedded device. Moreover, Wi-Fi devices such as laptop are generally not equipped with any positioning capability. To support these devices, LAPWiN can use an implicit notion of location in parallel. Due to the wide deployment of Wi-Fi networks, it is not uncommon to observe multiple APs in most residential areas. Therefore, the neighboring APs can be used for understanding the current location context.

We define a separate *pre-scan* to hear any neighboring APs. During the pre-scan phase, the LAPWiN device performs broadcast probe or passive scan, but its channel scanning duration is much shorter than the original scan. Given the set  $\mathbb{S}$  of all SSIDs in the local storage and the set  $\mathbb{P}$  of pre-scanned SSIDs, LAPWiN selects the SSID  $p \in \text{neigh}(s)$ ,  $\forall p \in \mathbb{P}$ ,  $\forall s \in \mathbb{S}$ , where  $\text{neigh}(s)$  is all SSIDs of neighboring APs of  $s$  including  $s$  itself.

Fig. 5 depicts the proximity testing of LAPWiN based on the implicit location context. At  $\mathcal{L}(C; t_1)$ ,  $C$  hears the APs  $A_1$ ,  $A_2$ , and  $A_3$ . Once  $C$  establishes a connection with  $A_1$ ,  $C$  records  $A_2$  and  $A_3$  into the SSID entry of  $A_1$ . If  $C$  revisits this area and is located inside the wireless coverage  $l_1$  of  $A_1$ , it probes with the SSID of  $A_1$  if it hears any of three APs. At  $\mathcal{L}(C; t_2)$ ,  $C$  probes with the SSID of  $A_1$  by hearing  $A_2$  (false positive). In this case, false positive increases as the wireless coverages of neighbor APs gets larger. However, an attacker still cannot capture the whole SSID list of Wi-Fi client without knowing all locations of APs.

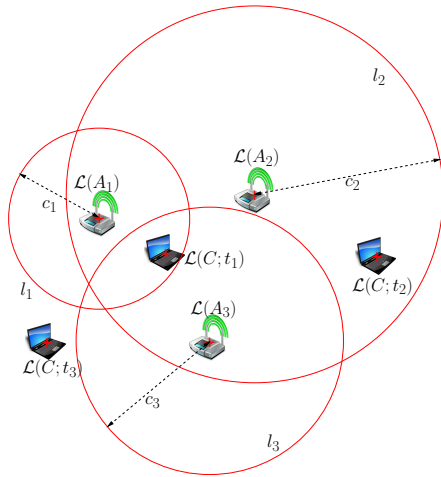


Fig. 5: The LAPWiN device  $C$  uses the neighbor APs  $A_2$ ,  $A_3$  to test the proximity to  $A_1$ .

Compared to the proximity testing based on explicit location context, this will increase false negative if not observing any neighboring APs: LAPWiN accidentally misses them or the neighboring APs change their configuration. Since it is hard to detect whether scanning failure is caused by LAPWiN or absence of neighbor APs, the recovery scheme should be also considered. When it keeps failing to find neighboring APs even after a certain number of attempts, the Wi-Fi client should

be able to recover to the original probing mechanism.

In contrast, there are commonplace SSIDs (e.g., *attwifi*, *boingo*) used by wireless service providers. This will bring an effect of increasing false positive, thereby giving an attacker more chances to collect the SSID list of Wi-Fi clients. To resolve this issue, the BSSID (commonly source MAC address of AP, which is unique) can be used for proximity test in LAPWiN instead of the SSID. One drawback of this solution is to raise false alarm whenever the BSSID of AP changes. For the sake of usability, it is not common to change the SSID compared to the BSSID. For example, when a network administrator upgrades or replaces the Wi-Fi AP hardware, the SSID is usually sustained. Another approach is maintaining a database of well known commonplace SSIDs. Putting a threshold in the number of observed neighboring APs tenses the proximity testing condition, thus decreasing false positive.

## B. Advanced Operations

1) *APs changing their location*: A network administrator may change the location of Wi-Fi APs as well as SSID for management purpose. LAPWiN will fail to find the uninstalled APs, and will try to use the original probing mechanism by the recovering policy. To prevent this process to be repeated, the corresponding SSID entry should be deleted when LAPWiN fails to connect to it many times by. Otherwise, a network administrator can explicitly notify the SSID change to users.

As defined in Section II-A we assume that the locations of Wi-Fi APs are fixed, but a certain type of APs keeps moving. They are installed in public transportations such as buses, subways, vessels, and airplanes to provide the Internet service. An individual can also carry the handheld type of or tethering Wi-Fi APs connected to wireless broadband backbones. Since this type of APs breaks the premise of LAPWiN relying on the fixed location of AP, a user should be able to configure the use of LAPWiN for each SSID entry. We will investigate the challenge with mobile APs and the more advanced probing mechanism in our future work.

2) *Hidden APs and APs not responding to broadcast probe*: Exposing the SSID of hidden AP is one of threats in Wi-Fi probing. Moreover, it is easily exposed by following authentication and association procedures. Unless we change the entire Wi-Fi connection protocols, it is impossible to solve this problem only by disguising the SSID in Wi-Fi probing.

To hide from the public or to protect DoS, APs may opt not responding to broadcast probe. They only allow the connections from the known clients. LAPWiN is based on direct probe, and it can thus still connect to this type of APs. Since, however, direct probe cannot search new APs, broadcast probe should be used together with LAPWiN in a hybrid manner in practice.

## IV. EVALUATION

In this section, we evaluate our proposed LAPWiN mechanism in terms of both performance and privacy, and detail our LAPWiN implementation.

### A. Analysis of LAPWiN Performance

To gauge the practicality of LAPWiN, we analyze the probing time and success rate of LAPWiN compared to other probing mechanisms. Since these measurements vary widely across positioning devices and Wi-Fi chipsets, largely due to proprietary variations in implementation such as channel scanning heuristics, we focus on theoretical performance bounds rather than hardware-specific measurement tests. We thus analyze probing times and success rates for the passive scan, active scan, and LAPWiN mechanisms.

We let  $N_c$  denote the number of channels to be scanned,  $T_p$  denote the time a Wi-Fi client stays on one channel, and  $T_{ps}$  denote the total passive scan time. A client scans each channel once in a scanning round, and thus  $T_{ps} = N_c T_p$ . Since many APs by default send beacons every 100 ms, the listening time per channel should satisfy  $T_p > 100$  ms, meaning that a passive scan can take several seconds.

Since a client may not be able to hear every frame from an AP, *e.g.*, a beacon frame or probe response, we further define the probability  $P_{ps}$  that a Wi-Fi client receives a frame from a particular AP after one round of scanning. Assuming a uniform probability  $p_l$  of frame loss/miss, this probability is given by

$$P_{ps} = \sum_{i=1}^{N_c} p_o(i) \left(1 - p_l^{\lfloor T_p/I_b \rfloor}\right), \quad (4)$$

where  $p_o(i)$  is the fraction of time the AP spends on the  $i^{\text{th}}$  channel (where  $\sum_{i=1}^{N_c} p_o(i) = 1$ ) and  $I_b$  is the beacon interval of the AP.

To analyze the active scan mechanisms, we note that a Wi-Fi client using direct probing may stop once it hears a response from a desired AP. Since many APs use the same SSID on multiple channels, a Wi-Fi client may need to scan all the channels if the best connection is sought. Due to the uncertainty in this process, we present both a lower bound  $\mathcal{L}(T_{dp})$  and upper bound  $\mathcal{U}(T_{dp})$  on the direct probing time  $T_{dp}$ . In our analysis, we suppose that a client waits  $T_a$  seconds for a response from nearby APs on each channel.<sup>2</sup> The upper and lower bounds are thus given by

$$\mathcal{U}(T_{dp}) = N_c T_a, \quad \mathcal{L}(T_{dp}) = d_r, \quad (5)$$

where  $d_r$  is the delay between sending a probe request by a Wi-Fi client and being responded by an AP, including the extra time spent on retransmitting the probe response due to frame loss.

If we assume that the number of available APs in the vicinity grows or that surrounding APs dynamically change their channel to avoid an inter-AP interference, the channel occupancy probability  $p_o$  will be uniform regardless of a channel ( $p_o = 1/N_c$ ). The loss probability  $p_l$  of probe request will also decrease as the number of responding APs gets larger (*i.e.*,  $p_l \rightarrow 0$ ). In these asymptotic cases, the average direct probing time  $\overline{T_{dp}}$  is derived as

$$\overline{T_{dp}} = \frac{1}{N_c} \sum_{i=1}^{N_c} ((i-1)T_a + d_r N_q) = \frac{N_c - 1}{2} T_a + d_r N_q, \quad (6)$$

<sup>2</sup>The delay of sending a probe request after switching to a channel is usually on the order of 10 – 100  $\mu\text{s}$ , which is much smaller than  $T_a$  in practice ( $T_a > 10$  ms in most implementations).

where  $N_q$  is the number of probe requests sent on a channel.

The broadcast probe is similar to the passive scan in the way that it searches all channels to find all available APs. However, with this approach the probe request is sent in order to get the fast response from the nearby APs. Because mixing the broadcast probe with the directed probe is a common practice, we use the same waiting time  $T_a$  on a channel for the broadcast probe without defining a different notation. The same applies to the hybrid probe since it also requires a Wi-Fi station to scan all channels. The broadcast probing time  $T_{bp}$  and the hybrid probing time  $T_{hp}$  are therefore defined as

$$T_{bp} = T_{hp} = N_c T_a. \quad (7)$$

Multiple APs will contend for channel reservation with high probability to send the probe response upon the received broadcast probe request. Thus, the response delay in the broadcast probe and the hybrid probe will be larger than the one in the directed probe as the number of responding APs increases.

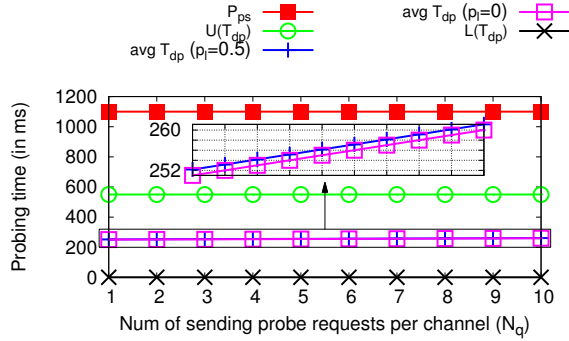
Given the maximum number of retransmission  $N_r$  for a probe response by AP, the success probability  $P_{as}$  of a Wi-Fi client finding an AP after one active scanning round can be represented as

$$P_{as} = \sum_{i=1}^{N_c} p_o(i) (1 - p_l^{N_r + 1}), \quad (8)$$

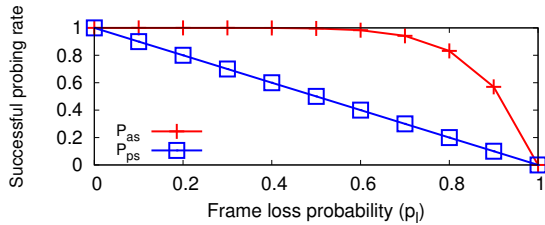
which is the same for any of the mentioned active scanning methods.

LAPWiN is based on direct probe, and therefore the probing time and successful probing rate of LAPWiN are similar to the direct probe, as long as the location information required for LAPWiN is available. If this information is not already available, the probing time of LAPWiN will be extended depending on the type of positioning devices used. In case of implicit proximity testing, the pre-scanning time will be added to the total probing time. If the pre-scan uses broadcast probe and waits for  $d_r$  to hear only the first probe response, the added time can be minimized at  $N_c d_r$ . Due to the neighboring APs, the successful probing rate of LAPWiN based on implicit proximity testing increases with the number of received neighboring APs.

Fig. 6 depicts the comparison of probing time and successful probing rate among different probing mechanisms by setting each parameter to values commonly used in practice. As the number of the sent probe requests per channel  $N_q$  increases, average direct probing time slowly increases as shown in Fig. 6a. Since LAPWiN brings the effect that reduces the number of sent probe requests per channel after filtering SSIDs, it slightly outperforms the standard direct probe in probing time, assuming no new localization occurs. In Fig. 6b, active scan shows more robust performance than passive scan as frame loss probability  $p_l$  increases in terms of successful probing time. This is because AP retransmits probe responses, which are unicast frames, when they get lost. Meanwhile broadcast frames such as beacons and probe requests are not retransmitted in Wi-Fi networks. LAPWiN also follows the same performance as active scan here.



(a) Comparison of probing time:  $T_{bp} = T_{hp} = \mathcal{U}(T_{dp})$ . All non-zoomed graphs are flat.



(b) Comparison of successful probing rate

Fig. 6: Performance comparison of various probing mechanisms is shown. We set  $T_p = 100$  ms,  $T_a = 50$  ms,  $d_r = 1$  ms,  $I_b = 100$  ms,  $N_c = 11$ ,  $p_o(i) = 1/11$  (uniform distribution), and  $N_r = 7$ .

### B. Implementation and Privacy Evaluation

To validate the feasibility of LAPWiN, we modify the *wpa\_supplicant* [10], a commonly used Linux-based network connection management software. The official Android open source project [11] provides source code for *wpa\_supplicant\_8*. Our implementation is based on a GPS-enabled *Nexus 7* tablet running *Android 4.2 (Jelly Bean)*.

We have taken several measurements to choose the various design parameters in the LAPWiN implementation. Namely, the connection threshold used in LAPWiN requires specification of the maximum reachable distance  $c$  between a Wi-Fi AP and client and the accuracy  $a_{max}$  of the location information. In an outdoor scenario, we measured an average reachable distance  $c$  of 78 meters and an average location accuracy  $a_{max}$  of 16 meters. In an indoor scenario, we measured an average reachable distance  $c$  of 44 meters and an average location accuracy  $a_{max}$  of 192 meters. In order to design a LAPWiN configuration that works in both environments, we choose a conservative threshold of  $2(c + a_{max}) = 472$  meters corresponding to the indoor measurements.

As previously described, we note that this choice of threshold is closely related to false positive cases in which the client device emits a probe message for an AP not in range and false negative cases in which the client device does not emit a probe for a nearby AP. Clearly, there is a trade-off between these false positive and false negative cases, with an infinite threshold representing the extreme trade-off made in current Wi-Fi systems, essentially attaining 0 false negatives but nearly 100% false positives. We have intentionally chosen the threshold to be significantly larger than the expected connection range to

minimize false negative cases (not emitting a probe for an AP in range) which delay the connection process.

Next, we investigate the reasonable parameters for the neighboring APs based proximity testing. We collect the beacon frames at eight different places: residence, enterprise office, hotel, restaurant, cafe, and conference site. Each place is distant from one another at least 1 kilometer away, and any APs are thus not physically shared. Fig. 7 shows the number of BSSIDs per SSID at each place with the CDF graphs. The hotels and the conference site use a few SSIDs having a large number of BSSIDs to serve many clients. The APs in the enterprise office also have many BSSIDs per SSID, and more SSIDs are provided to the users having various purposes. Due to the maintenance issue previously mentioned, we do not use the BSSID information for proximity testing. The observed number of SSIDs at each place is 6 at minimum and 124 at maximum with the average of 34.125. If we set the required number of neighboring SSIDs to determine the proximity to higher than 6, LAPWiN will frequently fail to access the Wi-Fi network (*i.e.*, increase of false negative). By exhaustively pairing data sets from 8 different places (*i.e.*,  $\binom{8}{2}$  cases in total), we found one SSID is shared in three cases, a pair of SSIDs are shared in two cases, and no SSID is shared in the other cases. The shared SSIDs at different places are ‘GoogleWiFi’, ‘GoogleWiFiSecure’, ‘xfinitywifi’, ‘CableWiFi’, and ‘linksys’. Thus, setting the required number of neighboring SSIDs for proximity testing to be smaller than 2 will cause the increase of false positive in LAPWiN. Again, since we are cautious about the usability in probing, we set the threshold value to 3 in this situation to minimize false negative.

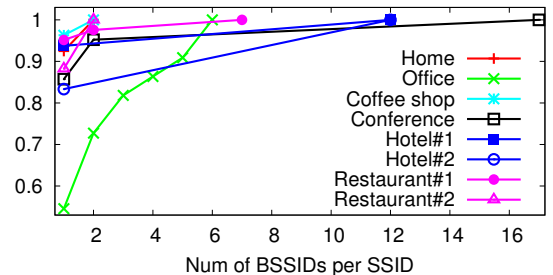


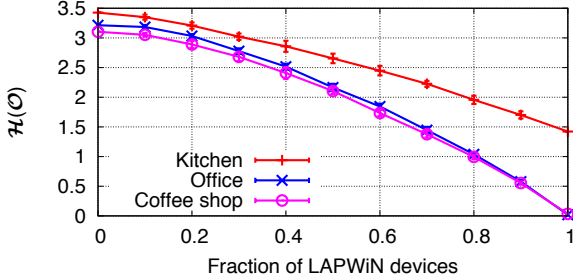
Fig. 7: The beacon frames at eight different places are captured to investigate the reasonable parameters for proximity testing based on neighboring APs. Each CDF graph shows the number of BSSIDs per SSID at each place.

Applying the parameters to our LAPWiN implementation, we evaluate the resulting privacy protections. We collected data at three different locations: a kitchen area on campus, an office environment with heavy Wi-Fi usage, and a coffee shop in an urban area. By passively capturing probe request frames using *Wireshark* [12], we observed 154 users and 266 unique SSIDs in a kitchen area, 423 users and 445 unique SSIDs in an office, and 182 users and 279 unique SSIDs in a coffee shop.

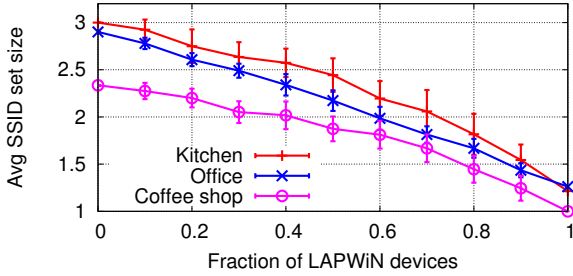
In the first experiment, we evaluate the uniqueness measure defined in Section II-B for the cases when a subset of devices use LAPWiN to prevent **user identification**. Fig. 8a shows the decrease of uniqueness  $\mathcal{H}(\mathcal{O})$  as more devices use LAPWiN. In all of three data sets, the uniqueness measure decreases as the fraction of LAPWiN devices increases. The lower the

uniqueness score the more difficult it is for an attacker to disambiguate between the devices. The kitchen data captures many testing mobile devices which have only one SSID of our campus AP. Thus, the kitchen data set with LAPWiN is not as unique as the other data sets.

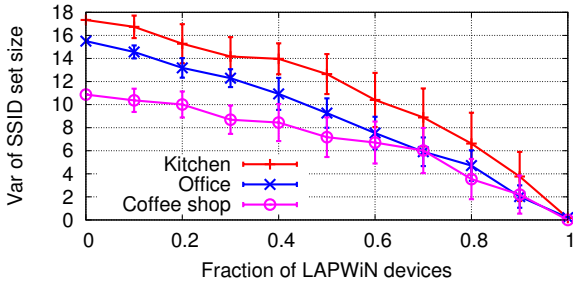
One of the main concerns of Wi-Fi probing is the fact that the list of revealed SSIDs often contain user's private



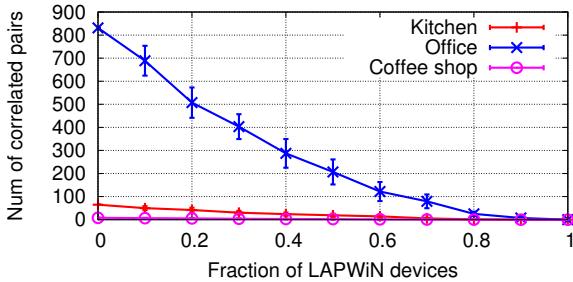
(a) Uniqueness of SSID sets per fraction of LAPWiN devices. We randomly selects each fraction of devices 10 times: we plot the average and the standard deviation from the 10 trials at each point by using the error bars ( $m \pm \sigma$ ).



(b) Average SSID set size after running 10 trials with different fraction of LAPWiN devices



(c) Variance of SSID set size for each case of (b)



(d) Number of correlated device pairs: we set  $\alpha = 2$  by merging all APs shared in each location into one.

Fig. 8: Privacy evaluation of LAPWiN for three different data sets is shown.

information. Therefore, the larger is the list of SSIDs the more information is used for **user profiling**. Many of the collected SSIDs carry names of hotels (e.g., Hyatt Regency Monterey, The Palms Guests), airports (McCarran WiFi, ZurichAirport), restaurants (Hooters, HarborLink - Buffalo Wild Wings) or even people (e.g., Mark's Guest Network, Justine's Network). Fig. 8b shows the average SSID set size depending on the fraction of LAPWiN devices and Fig. 8c shows the variance of SSID set size for each case. Both figures show that by using LAPWiN users reduce the SSID set size by sending out only relevant SSIDs, which makes safe from potential attackers.

In the third experiment, we count the number of correlated devices pairs to measure the effect of LAPWiN in terms of **user correlation**. We set the correlation threshold  $\alpha$  to 2 by merging all the Wi-Fi APs shared in each location into one. Fig. 8d shows the number of correlated pairs with the fraction of LAPWiN devices. The office data set shows the much larger correlation compared to the other data sets, because more users generates more pairs and many users use multiple devices in a workplace. On the other hand, in a coffee shop there are less number of devices used by each user and relatively weaker correlation exists among users. Overall, by applying LAPWiN we can largely minimize the number of correlated pairs.

## V. RELATED WORK

### Extracting user information from Wi-Fi probing

Many researchers have studied on identifying devices by exploiting Wi-Fi probing procedure. Desmond *et al.* and Franklin *et al.* use probing frequency and delay of active scanning to identify devices [6], [7], while it is also possible to mimic such features [13]. Greenstein *et al.* proposed approaches to use persistent link-layer address, list of known networks (SSIDs), and other protocol and physical layer characteristics to identify users [14]. Klasnja *et al.* conducted a user study to identify various privacy concerns in Wi-Fi use [15]. Husted *et al.* designed attacks to track users by using mobile devices in network as triangulators [16]. Cunche *et al.* proposed using probe requests to identify the relationships between users [17], [18]. Marco *et al.* generated social graph of smartphone users by analyzing their probe requests [19]. They further built the demographics about language usage and vendor adoption from the captured data set.

### Early defenses

We summarize the comparison among different probing mechanisms in terms of privacy and usability in Table I. All probing mechanisms except for direct probe are safe from the three privacy risks. In terms of connection speed, LAPWiN is fast as direct probe, since it is based on the same probing method, whereas broadcast probe and passive scan require more time to wait to hear all available APs. Meanwhile, broadcast probe and passive scan can find new APs, while others do not have this capability. Lastly, LAPWiN and direct probe can probe hidden APs and APs not responding to broadcast probe. Thus, for achieving both privacy and usability LAPWiN should be used with broadcast probe in a hybrid manner as the conventional direct probe commonly does.

Several mechanisms such as identifier obfuscation and encryption of probing messages proposed to mitigate the



TABLE I: Comparison of different probing mechanisms in term of privacy and usability

	LAPWiN	Direct probe	Broadcast probe	Passive scan
Safe from user identification?	✓		✓	✓
Safe from user profiling ?	✓		✓	✓
Safe from user correlation ?	✓		✓	✓
Probing speed	fast	fast	medium	slow
Can find new APs?			✓	✓
Can find special APs?*	✓	✓		

\*: hidden APs and APs not responding to broadcast probe.

privacy leakage of network discovery process. With identifier obfuscation, a Wi-Fi client avoids using its real link layer address in probing procedure, instead uses a pseudonym [20]. However, pseudonym provides the high level of anonymity only when there are enough number of other users. Obfuscating the protocol fields of management frames is another approach to protect Wi-Fi user privacy [21]. In practice, there are still many legacy Wi-Fi devices prohibiting to change the link layer address. Lindqvist *et al.* proposed a Wi-Fi probing protocol encrypting probing messages [8]. A Wi-Fi client sends the probe request with empty SSID, and an AP receiving the probe request responds with the SSID encrypted by the pre-shared key. This crypto-based protocol not only intrinsically imposes key management issue, but leads to extra computation overhead. As already reviewed, LAPWiN performs equal to or better than original probing, while other mechanisms require extra processing in client and/or AP.

Besides, early defenses are limited in terms of usability. Because users are not familiar with memorizing meaningless SSID numbers and the false connection requests will increase by mistake, obfuscating SSIDs severely impairs the Wi-Fi user experience. Unlike others, LAPWiN does not require any extra users' action, thus achieving high usability.

Last but not least, while identifier obfuscating and encrypting probing messages require considerable modification in both AP and client, thus making its real deployment impractical, LAPWiN only requires modification in only Wi-Fi clients. Table II summarizes the comparison with early defenses.

TABLE II: Comparison of LAPWiN with legacy defenses

	LAPWiN	SSID obfus.	Identifier obfus. [21], [20]	Encrypting probing msgs [8]
Performance	High	Medium	Medium	Medium
Usability	High	Low	Medium	High
Modified components	Client	AP	Both	Both

## VI. CONCLUSION

We revisited the potential threats in the current Wi-Fi probing and defined three risks: user identification, user profiling, and user correlation. To prevent these risks, we proposed a novel Wi-Fi probing mechanism LAPWiN utilizing location context. We implemented LAPWiN on the Android platform and evaluated its effectiveness in terms of both privacy and usability. LAPWiN requires modification only in Wi-Fi clients without existing protocols or special hardware, thus enabling

a practical deployment. We also reviewed the legacy defenses and compared our approach with them.

## REFERENCES

- [1] Skyhook, "Skyhook." [Online]. Available: <http://www.skyhookwireless.com>
- [2] Cisco Systems Inc., "Cisco mobility service engine." [Online]. Available: <http://www.cisco.com/en/US/products/ps9742/index.html>
- [3] *IEEE Std 802.11-2012, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society Std., 2012.
- [4] *IEEE Std 802.11w-2009, Amendment 4: Protected Management Frames*, IEEE Computer Society Std.
- [5] Wirelessdefence.org, "Karma Attack." [Online]. Available: <http://www.wirelessdefence.org/Contents/KARMAMain.htm>
- [6] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee, "Identifying unique devices through wireless fingerprinting," in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 46–55.
- [7] 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 *Proc. 15th USENIX Security Symposium*, 2006, pp. 167–178.
- [8] J. Lindqvist, T. Aura, G. Danezis, T. Koponen, A. Myllyniemi, J. Mäki, and M. Roe, "Privacy-preserving 802.11 access-point discovery," in *Proceedings of the second ACM conference on Wireless network security*, ser. WiSec '09. New York, NY, USA: ACM, 2009, pp. 123–130.
- [9] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001.
- [10] Linux WPA/WPA2/IEEE 802.1X Supplicant. [Online]. Available: [http://hostap.epitest.fi/wpa\\_supplicant/](http://hostap.epitest.fi/wpa_supplicant/)
- [11] Android Open Source Project. [Online]. Available: <http://source.android.com/>
- [12] Wireshark. [Online]. Available: <http://www.wireshark.org/>
- [13] Y. Liu and P. Ning, "Mimicry attacks against wireless link signature and defense using time-synched link signature," 2011.
- [14] B. Greenstein, R. Gummadi, J. Pang, M. Y. Chen, T. Kohno, S. Seshan, and D. Wetherall, "Can ferris bueller still have his day off? protecting privacy in the wireless era," in *Proceedings of 11th Workshop on Hot Topics in Operating Systems (HotOS XI)*, San Diego, CA, USA, 2007.
- [15] P. Klasnja, S. Consolvo, J. Jung, B. M. Greenstein, L. LeGrand, P. Powledge, and D. Wetherall, "'when i am on wi-fi, i am fearless': Privacy concerns & practices in eeryday wi-fi use," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1993–2002.
- [16] N. Husted and S. Myers, "Mobile location tracking in metro areas: malnets and others," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 85–96.
- [17] M. Cunche, M. A. Kaafar, and R. Boreli, "I know who you will meet this evening! linking wireless devices using wi-fi probe requests," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*. IEEE, 2012, pp. 1–9.
- [18] M. Cunche, M.-A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in wi-fi probe requests," *Pervasive and Mobile Computing*, no. 0, pp. –, 2013.
- [19] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa, "Signals from the crowd: Uncovering social relationships through smartphone probes," in *IMC '13: Proceedings of the 2013 ACM conference on Internet measurement conference*. ACM, 2013.
- [20] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. New York, NY, USA: ACM, 2007, pp. 246–257.
- [21] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall, "Improving wireless privacy with an identifier-free link layer protocol," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 2008, pp. 40–53.