2011

Defending Against Traffic Analysis in Wireless Networks Through Traffic Reshaping

Fan Zhang
University of Nebraska-Lincoln, fzhang2@unl.edu

Wenbo He
University of Nebraska-Lincoln, wenbohe@engr.unl.edu

Xue Liu
McGill University, xueliu@cs.mcgill.ca

2011 31st International Conference on Distributed Computing Systems

Defending Against Traffic Analysis in Wireless Networks Through Traffic Reshaping

Fan Zhang*†, Wenbo He* and Xue Liu‡
*Department of Electrical Engineering, University of Nebraska-Lincoln, NE, USA
†Department of Electronics and Information, Huazhong University of Sci. & Tech., Wuhan, China
‡School of Computer Science, Mcgill University, Quebec, Canada
Email: fzhang2@unl.edu, wenbohe@engr.unl.edu, xueliu@cs.mcgill.ca

***Abstract*—Traffic analysis has been exploited by attackers to threaten user privacy in wireless networks. As an example, a user's online activities may be exposed to strangers, even if the traffic is encrypted. However, the existing defense mechanisms against traffic analysis, such as packet padding and traffic morphing, are inefficient because they add noise traffic to blur the traffic features, therefore introducing significant overhead. In this paper, we propose the *traffic reshaping* technique to thwart traffic analysis. It creates multiple virtual media access control (MAC) interfaces over a single wireless card, dynamically schedules packets over these interfaces, thereby reshaping the packet features on each virtual interface. Hence, features of the original traffic are obscured and unavailable for the adversary to infer users' online activities. Unlike the existing solutions, *traffic reshaping* enhances privacy protection without incurring overhead in items of adding noise traffic. We evaluate the performance of *traffic reshaping* through trace-based experiments. The results show that *traffic reshaping* is effective and efficient in defending against the traffic analysis attacks.**

***Keywords*-Traffic Reshaping, Traffic Analysis, Privacy, Users' Online Activities, Virtualization**

I. Introduction

Due to the shared-medium nature of wireless links, adversaries can easily eavesdrop on the traffic from and to a specific user. Even if the traffic is encrypted, traffic features are still exposed to adversaries, and the user may suffer from traffic analysis attacks. Even worse, it may cause many upper-layer side-channel information leaks, which discovered in various online applications, such as web browsing [1], [2], video-streaming [3], and voice-over-IP (VoIP) applications [4], [5].

Traffic analysis extracts identifiable traffic features, such as packet size, frequency of a packet and the packet interarrival time, from traffic flows, and then associates the features with certain facts or secrets. Machine learning techniques, such as Support Vector Machine (SVM), Neural Network (NN), Bayesian techniques and Hidden Markov Models (HMM), can be used to enhance the accuracy of traffic analysis. Recent studies show that through traffic analysis an adversary can identify user's online activities (e.g., web-browsing, chatting, online gaming, downloading, uploading, online video and BitTorrent (BT)) [6] and glean what other users are browsing [1] in a few seconds with high accuracy. It turns out that traffic analysis has been a severe threats to user privacy in wireless networks.

A commonly used technique to defend against traffic analysis is packet padding [1], [2] (e.g., padding all packets to the same length), which usually incurs significant communication overhead, hence it is not an ideal solution. Traffic morphing [7], which modifies packet sizes to morph the network traffic from one class to another, is proposed to defend against traffic analysis in VoIP and web-browsing applications. But the communication overhead in terms of the increased payloads, reported from 15.4% to 38.9% [7], are not negligible.

It is very challenging to defend against traffic analysis effectively and efficiently. In this paper, we propose a novel approach, *traffic reshaping*, to prevent adversaries from inferring users' online activities through traffic analysis. *Traffic reshaping* creates multiple virtual MAC interfaces over a single wireless card, dynamically assigns packets over these interfaces, thereby changing the packet features on each virtual interface. Since *traffic reshaping* does not use packet splitting and reassembling, unlike the existing approaches (e.g., packet padding and traffic morphing), it does not incur additional overhead for noise traffic. The only message overhead introduced by *traffic reshaping* is for configuring virtual interfaces. Hence, *traffic reshaping* achieves better efficiency and performs well in defending against traffic analysis. Furthermore, *traffic reshaping* is a MAC layer solution and transparent to high level protocols. We evaluate the performance of *traffic reshaping* through trace-based experiments. The results show that the accuracy of traffic analysis decreases from 83.24% to 43.69% when the eavesdropping duration is 5 seconds. When the eavesdropping duration is extended to 1 minute, the accuracy remains unchanged as 44.49%, as compared with that of 91.86% under the situation without *traffic reshaping*.

The remainder of this paper is organized as follows. We present the background of our work in Section II. We then describe the detailed design of *traffic reshaping* against traffic analysis in Section III. In Section IV, we evaluate the *traffic reshaping* through real trace-based experiments. Section V discusses the implications and Section VI summarizes the related work. Finally, we conclude the paper in Section VII.

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.77

II. Background

A. Attack Model

The shared-medium nature of wireless links poses a great threat to user privacy. It is easy for an adversary to keep monitoring traffic traces from and to a specific user with sniffer software (e.g., Wireshark, Aircrack-ng) in current local area networks (WLANs) settings. Based on these traffic traces, an attacker is able to identify traffic features and use traffic analysis to link the features to certain facts or secrets.

As an example, the analysis based on the traffic features collected in a few seconds in the MAC layer is able to yield accurate estimation of users' online activities (i.e., the particular network application or service that a user is running), no matter what encryption schemes are used [6]. A user's online activities is regarded as highly private and sensitive, since the user usually do not want other persons in the same WLAN to track what they are doing on the Internet (e.g., web-browsing, chatting, online gaming and downloading, etc.). In addition, it is more risky that an adversary performs further attacks to get more sensitive personal information, such as which websites or contents a particular user is reading.

Traffic features, such as *average packet interarrival time*, *average packet size* and *packet size distribution*, can be used to profile users' actual online activities. For example, chatting and gaming are low traffic applications with smaller packets. Downloading and uploading are high traffic applications with large packet size in downlink and uplink, respectively. Also, online video demonstrates a relatively stable data rate and browsing contains bursty traffic. Figure 1 shows the packet size probability of distribution function (PDF) of seven popular online applications measured in residential environments[1] when the applications receive packets from the AP. It is obvious that traffic features can be employed to classify most applications.

Various traffic classification techniques, such as SVM and NN algorithms, Bayesian techniques, HMM, have been extensively studied. According to [6], the adversaries can accurately tell which online applications are active through SVM and NN algorithms. The accuracy reaches around *80%* when the eavesdropping duration is *5 seconds*. If eavesdroppers monitor the traffic for *one minute*, the classification accuracy is higher than *90%* and even achieves *100%* accuracy in most of the situations.

B. Existing Defense Against Traffic Analysis

The research defending against traffic analysis can be categorized into the following groups.

Traffic padding and packet padding are presented in [1], [2], [8], [9] to counter traffic analysis. Although these approaches may alleviate the problem, they are usually inefficient and incur high overheads. According to [7], padding the packets to the Maximum Transmission Unit (MTU) length of 1500 bytes incurs an overhead of 156.5%, and the adversary is still able to perform accurate classification with accuracy 86.2%.

[1]The received signal strength indicator is around -50dBm in the measurement.

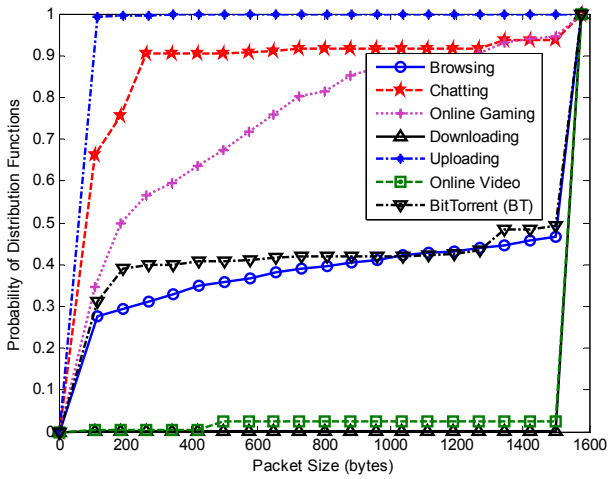


Figure 1. Packet size PDF of seven popular applications on receiver's side

Traffic morphing is proposed in [7] to thwart traffic analysis by modifying one class of traffic to look like another class. Hence, traffic morphing reduces the accuracy of traffic classification while incurring much less overhead than packet padding. Results in [7] show that the traffic morphing reduces the VoIP classifier's accuracy from 71% to 54% with 15.4% overhead on average. Likewise, the accuracy of the web classifier is reduced from 98.4% to 63.4% with 38.9% overhead. The overhead of traffic morphing for VoIP and HTTP applications, is not negligible. In addition, traffic morphing only changes the packet size, hence other features may still be sufficient for classification.

Identifier-free approaches [10], [11], which conceal the identifiers (i.e., MAC addresses) of users, can be utilized to prevent the adversary from associating the traffic features with the user's identity, thereby preserving the user's privacy in wireless networks. However, the physical layer measurements on traffic statistics (e.g., received signal strength indicator (RSSI) values) allow the adversary to link the packets with a specific user [12]. On the other hand, identifier-free approaches require to encrypt all the packets, including the packet header, control and management frames, thus the overhead of encryption and key managements can not be overlooked.

Frequency hopping changes the frequency of the communication channel periodically. It was designed to defeat frequency jamming and has the potential of preventing the adversary from obtaining the whole traffic traces from a user [13], thereby mitigating traffic analysis attacks.

Pseudonym schemes [14], [15] randomly change the MAC address of a user, so that adversary cannot track the entire traffic stream between the user and the AP. However, both frequency hopping and pseudonym schemes are insufficient to prevent traffic analysis attacks [6], [16], [10], because they do not obscure the traffic features when the traffic is partitioned over a single frequency channel or a specific MAC address. Hence, a single partition (i.e., piece of traffic trace) may release enough sensitive information for the adversary to perform traffic analysis accurately. For example, since pseudonym schemes only change MAC addresses each session or when idle, all the packets sent under one pseudonym are still linkable [10].

Physical space security and jamming approaches aim to reduce the number of packets that can be overheard by an eavesdropper. Lakshmanan et al. [17] and Sheth et al. [18] demonstrate that using directional antennas to focus transmissions within a secure physical space and jamming [19] have been suggested as methods to mitigate an eavesdropper's ability to overhear wireless packets. An intelligent jamming strategy deployed at potential eavesdropper locations can effectively raise the noise level to neutralize eavesdroppers, but jamming will also interfere with legitimate communications and degrade the network's performance [12].

In summary, the inefficiency of above existing approaches shows the following shortcomings. (1) Besides packet size, other traffic features (e.g., packet interarrival time) can still be used for traffic analysis. (2) The approaches partitioning the traffic over different frequency channels (i.e., frequency hopping) or using different MAC addresses (i.e., pseudonym) are at a coarse granularity, so the individual partitions of traffic may still lead to information leaks. Further, the traffic partitioning algorithms are naive and do not change the traffic features in a single partition. If the adversary accumulates the traffic traces in discrete time intervals, it is as if the adversary is monitoring all traffic in a smaller time scale. (3) Communication overhead (e.g., in padding and traffic morphing) or operational overhead (e.g., in identifier-free approaches) cannot be ignored.

In this paper, we use *traffic reshaping* to overcome above shortcomings and show that it is able to significantly improve the traditional defense against traffic analysis over the wireless links.

III. Traffic Reshaping

A. Overview

The goal of *traffic reshaping* is to enhance privacy protection by preventing information leaks without incurring noticeable overhead. To obscure traffic features without padding, we *reshape* the original traffic by dividing its packets into multiple sub-flows. Let each sub-flow be transmitted on a virtual wireless link and show partial patterns of the original traffic. Then attackers only get parts of traffic information, which may cause the misidentification of traffic analysis attacks.

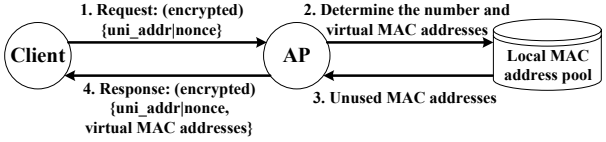


Figure 2. Configuration

Traffic reshaping executes in the MAC layer, hence, we only need to modify wireless device driver to support it. The modification includes two aspects.

First, we need to design communication protocols to support multiple virtual interfaces. We virtualize multiple virtual interfaces on Multiband Atheros Driver for WiFi (*MadWifi*). MadWifi is a popular WLAN driver and has the capability of creating multiple virtual MAC interfaces over a single physical interface [20]. Virtual interfaces are configured with different MAC addresses, but work in the same channel and keep association with the same AP. In *traffic reshaping*, each interface is treated as a fully functional, regular network interface, but only one adapter is active at any given time. Communication protocols are described in Section III-B.

Second, *traffic reshaping* explores an optimal scheduling, referred as *reshaping algorithm*, to partition traffic over virtual interfaces at a fine granularity in real time. Accordingly, traffic features (e.g., packet size distribution, packet interarrival time) on individual MAC interfaces shows different traffic patterns. Since, *traffic reshaping* does not add new data into the wireless link, it avoids the overhead for noise traffic. The reshaping algorithm is depicted in Section III-C.

B. Communications Between AP and Virtual Interfaces

1) Configuration: In *traffic reshaping*, both the AP and clients must be modified to support multiple virtual interfaces. In our design, virtual MAC addresses are assigned by the AP. The process includes four steps which are depicted in Figure 2. (1) First of all, a wireless client sends out a message to request virtual interfaces and their corresponding MAC addresses. (2) Upon receiving the request, the AP first chooses the number of virtual interfaces to create, denoted as I, determined by the privacy requirement and the resource availability. (3) Then the AP chooses unused addresses from its MAC address pool. The MAC address pool returns unused MAC address to the AP. Because there are 48 bits in a MAC address, randomly chosen addresses has a low probability of collision in small networks due to the birthday paradox. If N is the number of MAC addresses in the WLAN, the collision probability is about $\left(1 - 2^{48}!/2^{48N}(2^{48} - N)!\right)$. (4) Finally, the AP sends a reply that includes both the nonce from the request packet and the assigned MAC addresses.

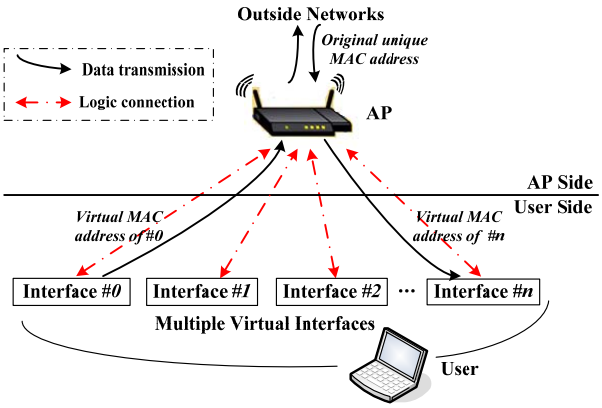


Figure 3. Data transmission between an AP and a user

The packets used in configuration are encrypted, thus the adversary does not know the mapping between the physical address and the virtual MAC addresses. When a wireless client receives the response from AP, it checks if the nonce corresponds to the request that it has sent. If so, it begins to virtualize multiple MAC interfaces and configure them with the corresponding MAC addresses. The AP is able to recycle and dynamically configure virtual MAC interfaces according to the change of resource availability and client requirements.

2) Data Transmission: When a client is ready to send a packet to the AP, it first adopts the reshaping algorithm (described in Section III-C) to determine a specific virtual interface. The virtual MAC interface (e.g., *Interface #0* shown in Figure 3) encapsulates an outgoing packet by filling the source address of the packet with its own MAC address, then the interface sends the packet to the AP. When the AP receives the packets, it should find the MAC Header and check the source address. If the address belongs to a virtual MAC interface, the AP needs to replace the source address by the unique physical MAC address of that wireless client. The MAC address translation should be done in order to circumvent the ARP protocol, hence the remote servers do not need any modifications.

When remote servers send packets to a client, packets are first sent to the AP which the client is associated with. AP first checks whether the destination uses virtual interfaces or not. If not, it sends the packet to the destination as usual. Otherwise, the AP employs the reshaping algorithm to determine a specific virtual interface and replaces the unique physical MAC address with the corresponding virtual MAC address (*Interface #n* shown in Figure 3). After that, the packet is sent to the client through that virtual interface. On the client side, the MAC layer of the client has been modified to receive all the packets whose destination address is one of its virtual MAC addresses. Then, it translates the virtual MAC address to the unique physical MAC address and sends the packets to the upper layers. The MAC address translation makes the modification in the MAC layer transparent to upper layers.

C. Reshaping Algorithm

1) Optimization Problem: The reshaping algorithm aims to obscure original traffic features by dispatching the packets to multiple virtual interfaces. Packets distributed into different interfaces look as if they are from independent applications. The reshaping algorithm is running on both the client and AP side. In detail, we denote the number of virtual interfaces as I. Packets scheduled to virtual interfaces are described as a set $S = (s_1, s_2, \cdots, s_k, \cdots, s_{\mathbb{N}}), (\mathbb{N} \to \infty)$. The reshaping algorithm is considered as a function to map a packet to a virtual interface in real time:

$$\mathscr{F}(s_k) = i, \quad i \in [1, I].$$

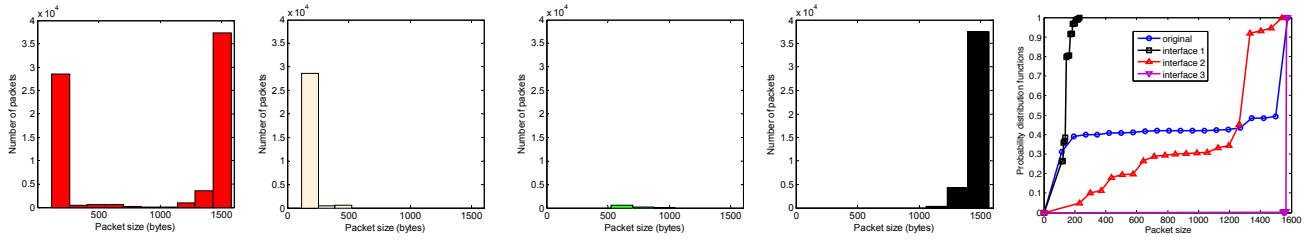
Let S^i represent the set of packets on interface i. S^i is a subset of S, and $\cup_i S^i = S$, $S^i \cap S^j = \emptyset$. For example, using the ***Random Algorithm (RA)***, a packet s_k is randomly scheduled on virtual interface i, i.e., $i = \mathrm{mod}(random[1, I])$. In ***Round-Robin (RR) algorithm***, packet s_k is scheduled on virtual interface $i = \mathrm{mod}[k, I]$.

In *traffic reshaping*, the interarrival time of packets over individual interfaces is automatically changed from the original traffic. Hence, we focus on changing the distribution of the packet size to deceive the adversaries.

First, define the packet size of s_k as $\mathscr{L}(s_k)$. The maximum packet size of set S is denoted as ℓ_{max}. Assume there are L possible packet size ranges, $\{(0, \ell_1], (\ell_1, \ell_2], \cdots, (\ell_{L-1}, \ell_L]\}$, where $\ell_L = \ell_{max}$. To describe the packet distribution of the original traffic, $P_j, (P_j \in [0, 1])$ is denoted as the probability of packet size which falls into a particular range $(\ell_{j-1}, \ell_j]$.

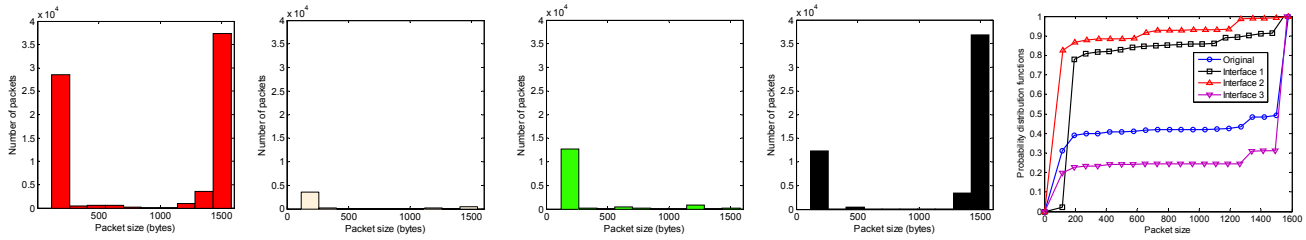
$$P_j = \Pr(\{s_k \in S : \mathscr{L}(s_k) \in (\ell_{j-1}, \ell_j]\}), \quad (\sum_{j=1}^{L} P_j = 1).$$

Similarly, p_j^i is denoted as the probability of packets whose size is among $(\ell_{j-1}, \ell_j]$ on the virtual interface i. We design the *reshaping algorithm* to alter the packet size distribution on each interface, so that the traffic is unidentifiable or looks like another application. The *target probability distribution* on interface i is denoted as ϕ^i, and $\phi^i = [\phi_1^i, \phi_2^i, \cdots, \phi_j^i, \cdots, \phi_L^i]$, where ϕ_j^i is defined as the target probability of packet sizes within $(\ell_{j-1}, \ell_j]$ on the virtual interface i. Preferably, p_j^i will close to ϕ_j^i, hence the reshaping algorithm can be formulated as an optimization problem with different *target probability distribution*:



(a) Original Traffic (b) Interface 1 (c) Interface 2 (d) Interface 3 (e) PDF

Figure 4. OR schedules a BT application by packet size ranges. Three ranges are $(0, 525]$, $(525, 1050]$, $(1050, 1576]$.



(a) Original Traffic (b) Interface 1 (c) Interface 2 (d) Interface 3 (e) PDF

Figure 5. OR schedules a BT application by packet sizes. A packet s_k is distributed to virtual interface $i = \mathrm{mod}[\mathscr{L}(s_k), I]$.

$$\begin{aligned} &\min \sum_{i=1}^{I} \sqrt{\left(\sum_{j=1}^{L} \left|\phi_j^i - p_j^i\right|^2\right)} \\ &\text{sub} \sum_{i=1}^{I} p_j^i \mathscr{N}(i) = P_j \mathbb{N}; \sum_{i=1}^{I} \mathscr{N}(i) = \mathbb{N}; \\ &\sum_{j=1}^{L} \phi_j^i = 1; \sum_{j=1}^{L} p_j^i = 1; \\ &\phi_j^i, p_j^i \in [0,1]; i \in [1, I]; j \in [1, L]; \end{aligned} \tag{1}$$

where $\mathscr{N}(i)$ is the number of packets on the interface i, and $\mathbb{N}$ is the total number of packets. *Traffic reshaping* optimality is desired, so that packets are scheduled to hide the original traffic features in the "best" way. Here, the "best" scheduling is defined by the target distribution, which disguises the original traffic as the target traffic on individual wireless interfaces. Hence, different reshaping algorithms over multiple virtual wireless interfaces can be designed to achieve different target distributions.

2) Algorithm Description: With multiple MAC layer interfaces, protection against traffic analysis is not automatically achieved. For example, under the naive algorithms, such as RA and RR, if an attack eavesdrops on any wireless interfaces for a longer time period, the attacker is able to collect enough traffic information to obtain users' online activities. Hence, efficient and satisfiable algorithms should be able to hide traffic features of the original traffic on individual wireless interfaces. Next, we propose an intelligent scheduling algorithm, named ***Orthogonal Reshaping (OR)***.

OR dispatches the packets with a certain size range to a specific virtual interface and makes the distribution of virtual interfaces unidentifiable. In OR, the packet size distribution of each virtual interface is *orthogonal* to each other. *orthogonal* means the *dot product* of two target distribution should be zero. It is described as follows.

$$\forall\{i_1, i_2 \in [1, I]\}, \phi^{i_1} \cdot \phi^{i_2} = \sum_{j=1}^{L} \left(\phi_j^{i_1} \phi_j^{i_2}\right) = 0 \tag{2}$$

Since $\phi_j^i \in [0, 1]$, the orthogonality leads to $\forall j \in [1, L]$, $\exists! i : \phi_j^i = 1$.

OR simplifies the selection of target distribution, and makes the online optimization to Equation (1) achievable by satisfying $p_j^i = \phi_j^i$. In this case, the optimal solution is achieved without knowing the future traffic.

Actually, OR can be regarded as a *hash function*, which maps multiple packet sizes to a certain virtual interface. Different mappings show different packet distributions. For example, $L = I$. Packets among each packet size range belong to a virtual interface. Assuming $I = 3, L = 3$, we divide the packet size into three ranges, which have similar length. Then, $\ell_1 = 525, \ell_2 = 1050, \ell_3 = 1576$. To hide a BitTorrent (BT) application, we dispatch the packets whose size less than ℓ_1 to interface 1, size within $(\ell_1, \ell_2]$ to interface 2, and others to interface 3. That means $\phi^1 = [1, 0, 0]$, $\phi^2 = [0, 1, 0]$ and $\phi^3 = [0, 0, 1]$. The packets of each interface are shown in Figure 4. We can see that different groups of packets are separated apart by OR and each interface only has the packets within the same range. The PDF of packet size for the original traffic and each virtual interface is shown in Figure 4(e). The distribution of each interface differs from each other and is also very

different from the original traffic.

In addition, the scheduling algorithms can be flexibly selected by users. In the above example, the traffic over each virtual interface has a small packet size range. Next, we give another example of OR to make the traffic have a large packet size range. This is a good property to prevent adversaries from telling if the *traffic reshaping* technique is being used. We set $L = \ell_{max}$. A packet s_k with the size $\mathscr{L}(s_k)$ is distributed to virtual interface i, $i = \mod[\mathscr{L}(s_k), I]$. We take the same BT application as an example. $I = 3$. The packets of each interface are shown in Figure 5. We see three virtual interfaces have very different traffic features. Through these two examples, we show that different scheduling policies may give different *traffic reshaping* results.

OR may be adopted on both the AP side and the client side. Each side can employ different parameters and dynamically change according to different applications. Further, we describe the packet distribution based on packet size in above algorithms. Other features may also be employed to characterize the distribution (e.g., number of packets).

3) Parameter Selection: From Equation (1), we can see that parameters L, I and ϕ_j^i need to be tuned dynamically for different applications. The selection rules are listed below.

Number of L. L is used to partition the distribution of packet size. Its selection is related to the reshaping algorithm. We can determine L according to privacy requirement and features of different applications. We observe that the main packet size of each application is distributed around two ranges: $[108, 232]$ and $[1546, 1576]$. So we can divide the packet size into three ranges: $(0, 232]$, $(232, 1540]$ and $(1540, 1576]$. So generally, we set $L \geq 3$.

Number of I. We denote the total number of MAC addresses in the WLAN as N. If the attacker has no additional information, the *privacy entropy* H is equal to $\log_2 N$ [14]. Hence if we increase I, N will be larger and users may get more privacy protection. But too many MAC addresses may cost more resources and increase operation costs. We have evaluated the effect of I on performance of reshaping algorithms in Section IV-C. Generally, $I = 3$ is enough for OR to perform well. In addition, I can be adjusted dynamically according to the privacy requirement and the resource availability.

Configuration of ϕ_j^i. ϕ_j^i is correlative with L. It should be design carefully to conceal traffic features of the original traffic and also prevent the adversary from linking multiple virtual interfaces with a certain user.

IV. Evaluation

We evaluate the performance of *traffic reshaping* through trace-based experiments, and the real traces are collected by *Intel Wireless WiFi Link 4965AGN* network cards with *Libpcap* library and *Proxim AP-2000 11b/g Cardbus Series (Atheros 5212 chipset)* network cards with the MadWifi driver (version madwifi-0.9.4 branch). The classification system in [6], including SVM and NN algorithms, is used to identify users' online activities. We validate the effectiveness and efficiency of OR and compare it with other defending schemes. Results show that *traffic reshaping*, which significantly reduces the classification accuracy without incurring additional overhead, performs better than packet padding and traffic morphing. We also compare the performance of four scheduling algorithms, including frequency hopping (FH) scheme[2], RR, RA and OR. Furthermore, the effect of parameters is evaluated.

Table I
Features on virtual interfaces (from AP to the user)

App.	Features (byte, second)	Original	OR		
			$i = 1$	$i = 2$	$i = 3$
br.	Avg. packet size	1013.2	134.0	780.6	1574.3
br.	Interarrival time	0.0284	0.0918	0.1087	0.0278
ch.	Avg. packet size	269.1	145.3	517.3	1576.0
ch.	Interarrival time	0.9901	1.1022	0.0687	0.0257
ga.	Avg. packet size	459.5	138.8	689.66	1575.3
ga.	Interarrival time	0.3084	0.4970	0.6899	0.4835
do.	Avg. packet size	1575.3	136.8	536.7	1576.0
do.	Interarrival time	0.0023	0.4242	0.5138	0.0023
up.	Avg. packet size	132.8	131.4	379.0	1576.0
up.	Interarrival time	0.0301	0.0302	0.0123	0.0965
vo.	Avg. packet size	1547.6	129.6	528.5	1576.0
vo.	Interarrival time	0.0119	0.3159	0.5493	0.0122
bt.	Avg. packet size	962.04	143.9	1062.5	1568.0
bt.	Interarrival time	0.0247	0.0634	0.2331	0.0486

Two metrics, *accuracy* and *false positive (FP)*, are employed to evaluate the performance of *traffic reshaping*. *Accuracy* is the percentage of correctly classified instances among the total number of instances, and *mean accuracy* is defined as overall average recognition probability of classifiers. Differently, *FP* reflects the percent of non-class X packets incorrectly classified as belonging to class X [22].

A. Scenarios

The accuracy of traffic classification in WLANs at home is higher than those in public areas and university campus [6]. Hence, if *traffic reshaping* is able to reduce the classification accuracy largely in home scenarios, it will perform better in public, enterprise and university campus WLANs. Therefore, we evaluate performance of *traffic reshaping* in home scenarios. We get traffic traces at home with *Comcast Internet* and *Time Warner Cable* as the Internet services. WLANs support 802.11a/b/g modes and the data rate may fluctuate from 1Mbps to 54Mbps. We examine seven applications, including web browsing, chatting, online game, downloading, uploading, online video and BT. We totally get more than 50 hours of traffic traces. By using the above traffic traces, we evaluate *traffic reshaping* through simulations. The eavesdropping duration (denoted as W) is

[2]We adopt FH by using VirtualWiFi [21] and channels are accessed in the order of 1,6,11. The active time period for each channel is 500ms.

Table II
ACCURACY OF CLASSIFICATION ($W = 5s$)

App.	Original (%)	FH (%)	RA (%)	RR (%)	OR (%)
br.	37.77	59.15	58.74	59.16	1.90
ch.	77.93	86.17	85.82	81.63	84.21
ga.	88.18	61.01	60.24	61.35	26.61
do.	99.88	98.26	95.59	94.25	99.95
up.	95.92	91.76	89.30	94.98	90.78
vo.	93.32	96.37	86.01	86.52	0.00
bt.	89.68	33.88	57.69	59.04	2.35
Mean	**83.24**	**75.23**	**76.20**	**76.70**	**43.69**

Table III
ACCURACY OF CLASSIFICATION ($W = 60s$)

App.	Original (%)	FH (%)	RA (%)	RR (%)	OR (%)
br.	72.94	72.59	76.72	77.90	0.57
ch.	85.29	81.09	67.67	64.89	93.86
ga.	93.74	79.71	81.36	81.67	23.64
do.	100.0	100.0	100.0	100.0	99.96
up.	95.92	91.76	89.30	94.98	90.78
vo.	100.0	100.0	100.0	100.0	0.00
bt.	95.14	93.63	96.44	97.02	2.61
Mean	**91.86**	**88.40**	**87.36**	**88.07**	**44.49**

used to represent the shortest time duration of traffic for classification each time.

B. Traffic Features

By default, we set the number of virtual interfaces, $I = 3$. Packet sizes are mostly divided into three ranges ($L = 3$): $(0, 232]$, $(232, 1540]$ and $(1540, 1576]$. Target distributions of different applications in OR are orthogonal ($\phi^1 = [1, 0, 0]$, $\phi^2 = [0, 1, 0]$ and $\phi^3 = [0, 0, 1]$).

By using the above configuration, we present the changes of features under *traffic reshaping* in Table I. The features are gotten according to the same processing as [6]. Because the eavesdropping duration W is mostly set at 5 seconds, the idle time without data transmission, which is beyond 5 seconds, is filtered out and is not calculated into the packet interarrival time. We see that the packet features of virtual interfaces greatly differ from their original values and are also different from each other. The average interarrival time is mostly larger than that of the original traffic.

C. Effectiveness

We use the same classification system as in [6], including SVM and NN techniques, to infer users' online activities and evaluate the performance of *traffic reshaping*. Features we employed in the classification are *number of packets*, *max/min/average/standard deviation of packet size*, and *packet interarrival time* in downlink and uplink. We present the highest classification accuracy based on these features.

Classification system in [6] can infer what a user is doing with the accuracy about 83.24% in 5 seconds. The accuracy achieves 91.86% when the eavesdropping duration, W, lasts for 1 minute. From Table II and Table III, we see that the accuracies of FH, RA and RR are all around 75%, which is close to the original result, 83.24%. Each application has similar accuracy. When W is extended to 1 minute, the accuracies of FH, RA and RR rise to about 88%. Because the main feature, "average packet size," is almost unchanged in FH, RA and RR, hence they do not bring down the classification accuracy of the original traffic. Therefore, FH, RA and RR can hardly prevent attackers from inferring the users' online activities.

OR decreases the classification accuracy quite markedly, shown in Table II and Table III. It has the lowest classification accuracy, 43.69%, among four algorithms. Furthermore, the accuracies in OR barely rise along with the increase of W. They nearly remain unchanged at 44.49% when W is set at 60 seconds.

From the results of the classification accuracy, we find that browsing, online video and BT applications are unidentifiable in OR. In contrast, chatting and downloading achieve high accuracy, even larger than the original case. The reason is that most packet sizes of applications are distributed around two ranges: $[108, 232]$ and $[1546, 1576]$, which look like chatting and downloading, respectively. Thus the classification tends to identifying the traffic in OR as chatting or downloading.

However, high accuracy does not mean an adversary is easy to detect the application. To describe it more clearly, we introduce FP, which denotes the percentage of members of other classes incorrectly classified as belonging to this class. The FP of classification is presented in Table IV. We see that the FP of OR, which is around 9%, is much larger than the original traffic. Also, the FP is nearly unchanged when the eavesdropping duration increases. High FP may cause many false identifications for the adversary. Chatting and downloading both have high FP than other applications. For instance, 34.77% of packets from other applications are regarded as downloading if the traffic is protected by OR. Among all of the applications, only uploading has high accuracy and very low FP. The reason is that uploading is the only application which has low traffic in downlink but high traffic in uplink, compared with other applications. In summary, OR performs better than other algorithms. It is effective to defend against traffic analysis on inferring users' online activities.

Table V describes the performance of OR when virtual interface I changes from 2 to 5. For $I = 2$, we set $L = 2$; and two packet size ranges are $(0, 1500]$ and $(1500, 1576]$. When $I = 5$, we set $L = 5$. The packet size is divided into five ranges: $(0, 232]$, $(232, 500]$, $(500, 1000]$, $(1000, 1540]$ and $(1540, 1576]$. We get the value of ϕ^i in OR by Expression 2. From Table V, we see that the accuracy decreases along with the increase of I, and the decrease is less and less evident. Hence, we generally set $I = 3$ and it is enough for OR to thwart the traffic analysis attack on users' online activities.

Table IV
FP OF CLASSIFICATION

App.	$W = 5s$		$W = 60s$	
	Original (%)	OR (%)	Original (%)	OR (%)
br.	2.73	1.91	1.51	2.30
ch.	2.21	21.01	1.45	19.73
ga.	3.29	3.55	1.86	1.54
do.	0.93	34.77	0.13	35.47
up.	0.02	0.00	0.00	0.00
vo.	1.05	0.44	0.30	0.00
bt.	9.32	4.00	4.25	5.72
Mean	**2.80**	**9.38**	**1.36**	**9.25**

Table V
ACCURACY CHANGES BY DIFFERENT VIRTUAL INTERFACES I

App.	OR (%)		
	$I = 2$	$I = 3$	$I = 5$
br.	2.82	1.90	1.52
ch.	91.63	84.21	90.35
ga.	56.83	26.61	17.24
do.	99.92	99.95	99.37
up.	95.59	90.78	90.53
vo.	0.00	0.00	0.00
bt.	2.47	2.35	0.49
Mean	**49.89**	**43.69**	**42.79**

D. Efficiency

We compare the efficiency of *traffic reshaping* with packet padding and traffic morphing. The results of our experiments, shown in Table VI, clearly illustrate the superiority of our technique over packet padding and traffic morphing. Both packet padding and traffic morphing attempt to hide traffic features by changing packet sizes. In packet padding, we pad all the packets to the maximum packet size (i.e., 1576 bytes). In contrast, we apply traffic morphing by modifying the packet of one application to look like another similar application. Specifically, we morph chatting to be gaming, disguise gaming as browsing, simulate browsing as BT, make BT look like online video, pad video to be downloading.

Table VI
EFFICIENCY COMPARISON ($W = 5s$)

App.	Accuracy (%) (Padding and Morphing)	Overhead (%) (Padding)	Overhead (%) (Morphing)
br.	31.37	55.55	28.67
ch.	72.15	485.74	54.62
ga.	71.68	242.96	128.42
do.	100	0.04	0
up.	95.92	0	0
vo.	91.81	1.84	1.83
bt.	37.54	63.82	62.52
Mean	**71.18**	**121.42**	**39.44**

As shown in Table VI, The overhead of packet padding, 121.42%, is unbearably high. Traffic morphing incurs 39.44% overhead, which is less than packet padding. With the above significant costs, the classification accuracy is still as high as 71.18%. The results are caused by using different ways to hide packet features. In this scenario, we use the traffic analysis attack based on the feature, the packet interarrival time. Since packet padding and traffic morphing only change the packet size, they have the same accuracy in terms of timing attack. In addition, the packet interarrival time can reveal enough information on users' online activities, even though all packets are padded as the same size. Hence, both packet padding and traffic morphing fail to preserve user privacy. Our defending method, *traffic reshaping*, achieves quite low accuracy, 43.69%, without additional communication overhead. In summary, compared with packet padding and morphing, *traffic reshaping* is a significant improvement in both privacy and overhead.

V. DISCUSSION

A. Against Power Analysis

As wireless signals fade with distance as they propagate over wireless medium, the same transmission will be received at different RSSI levels, depending on the distance between the transmitter and receiver. Adversaries may adopt wireless signal strength to infer a user's location and, therefore, associate packets to a specific user (or wireless card) [23][12]. To avoid vulnerability of power analysis, we can use the per-packet-based transmission power control (TPC) technique [24] to set a different transmission power for each packet. This fine granularity adjustment adds noises to RSSI values, therefore, we can disguise multiple virtual interface as multiple users in the same WLAN.

B. Scalability

As described in Section IV-C, if a client has three virtual interfaces, it is enough for *traffic reshaping* to thwart traffic analysis. Hence, it does not cost much for an AP to maintain all these virtual interfaces in a WLAN. In addition, AP can dynamically distribute and configure the virtual interfaces for each client according to the resource availability and privacy requirement. On the other hand, *traffic reshaping* does not incur overhead for noise traffic. The only message overhead introduced by *traffic reshaping* is for configuring virtual interfaces. The operation cost is also lightweight. The computational complexity of OR is $O(\mathbb{N})$, where $\mathbb{N}$ is the total number of packets through the AP or the client. Furthermore, *traffic reshaping* is compatible with standard WLAN protocols and transparent to users. In summary, *traffic reshaping* has good scalability and suitable for WLANs deployed in residential, hotspot and campus environments.

C. Compatibility with other techniques

Traffic reshaping is efficient in defending against traffic analysis. To make it more powerful, we can use it together with existing solutions. For instance, we use *traffic reshaping* together with traffic morphing on a virtual interface. In this case, the accuracy will be reduced further while incurring

much less overhead than traffic morphing. As an example, morphing chatting to look like gaming in a certain virtual interface, and modifying packet size of gaming to pretend browsing, then only downloading and uploading have the accuracy larger than 90%, others are close to zero. The mean accuracy will decrease to less than 28%. Furthermore, if we allow splitting packets of downloading and uploading into multiple smaller packets, the accuracy will be reduced even more, but it will sacrifice the network performance.

VI. Related Work

Side-channel Information Leaks: Side-channel leaks have threatened user privacy in context of secure shell (SSH) [25], keystroke dynamics [26], [27], web browsing [1], [2], [28], video-streaming [3], and VoIP [4][5]. Encrypted traffic does not prevent an attacker from discovering user privacy through traffic analysis. A straightforward traffic analysis attack against encrypted HTTP streams is presented in [29], [30] to identify the source of the traffic. Chen, et al. [1] find that the significant traffic distinctions of different websites help adversary to wiretap what the user is browsing. Moreover, the length of encrypted VoIP packets can be used to identify the phrases spoken within a call [5]. In a similar way, Zhang, et al. [6] utilize the traffic features to profile users' actual online activities accurately.

Regarding the defense, high-level mitigation policies, such as packet padding are likely to be inefficient or incur prohibitively high overheads [1]. Traffic morphing [7] only defends against traffic analysis based on packet size, which may be easily overridden by other features. Its design also leads to significant overhead. So an efficient defense against the side-channel information leaks is a future research topic with strong practical relevance.

Privacy of WiFi Networks: Due to the shared-medium nature, WiFi communication poses a great challenge on user privacy. Recent privacy preserving in WiFi networks has mainly focused on location privacy and user identification [16], [14]. From public available databases of WiFi networks [31], it is feasible to tracking users' location through the analysis of the log files. Furthermore, adversaries may adopt wireless signal strength in multiple monitoring locations to obtain an accurate estimation of a user's location and motion [23], [32]. Franklin et al. [33] show that it is possible to fingerprint drivers by using the timing of 802.11 probes.

Regarding the defense, Gruteser et al. [15] and Jiang et al. [14] propose to use pseudonyms within WiFi networks to hide or frequently change user identities. But pseudonyms change MAC address at a coarse granularity, so the individual partitions of traffic may still lead to information leaks. A wireless identifier-free link layer protocol proposed in [10] obscures all explicit identifiers from all transmitted bits to improve privacy, but the overhead of encryption and key management cannot be overlooked.

Virtualization in WiFi Networks: Researchers have introduced virtualization into WiFi networks, which mainly focuses on improving network performance. A fat virtual AP [34] is an 802.11 driver that provides aggregated bandwidth of available APs, and balances their loads. VirtualWiFi [21] or MultiNet [35] uses a network hopping scheme to switch the wireless card across multiple APs of wireless LANs with one MAC address in order to improve network throughput. The PeerBoost [36] system uses the coexistence of infrastructure and ad hoc mode over the one wireless card to maximize the throughput of WiFi networks. SoftRepeater [37], is a practical, deployable system, in which stations cooperatively address the rate anomaly problem by using VirtualWiFi. Different from the above papers, *traffic reshaping* with different MAC addresses and dynamically dispatches traffic flows to multiple infrastructure-based links optimally.

VII. Conclusions

In this paper, we propose *traffic reshaping* to protect users' online privacy. It creates multiple virtual MAC interfaces, dynamically dispatches traffic flows among these interfaces, and reshapes different traffic features on each virtual interface to hide those of the original traffic. Since *traffic reshaping* does not use packet padding, it thwarts traffic analysis without additional overhead for noise traffic. We evaluate the performance of *traffic reshaping* through trace-based experiments. It performs better than packet padding and traffic morphing in defending against traffic analysis. The results show that the accuracy of classification decreases from 83.24% to 43.69% when the eavesdropping duration is 5 seconds. When eavesdropping duration is extended to 60 seconds, the accuracy is reduced from 91.86% to 44.49%. Therefore, *traffic reshaping* is an efficient way to defend against traffic analysis.

References

[1] S. Chen, R. Wang, X. Wang, and K. Zhang. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 191–206, 2010.

[2] Q. Sun, D.R. Simon, Y. Wang, W. Russell, V.N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of IEEE Symposium on Security and Privacy*, 2002.

[3] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, and T. Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of USENIX Security Symposium*, 2007.

[4] C.V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted VoIP traffic: Alejandra y roberto or alice and bob. In *Proceedings of USENIX Security Symposium*, 2007.

[5] C.V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Proceedings of IEEE Symposium on Security and Privacy*, 2008.

[6] F. Zhang, W. He, X. Liu, and P. Bridges. Inferring users' online activities through traffic classification. In *Proceedings of WiSec*, 2011.

[7] C.V. Wright, S.E. Coull, and F. Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *Proceedings of NDSS*, 2009.

[8] RFC4949, August, 2007. Internet Security Glossary, Version 2.

[9] W. Stallings. *Cryptography and network security*, volume 3. Prentice Hall New Jersey, 2003.

[10] B. Greenstein, D. Mccoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceeding of MobiSys*, 2008.

[11] Y. Fan, B. Lin, Y. Jiang, and X. Shen. An efficient privacy-preserving scheme for wireless link layer security. In *Proceedings of GLOBECOM*, pages 4652–4656, 2008.

[12] B. Greenstein D. Grunwald K. Bauer, D. McCoy and D. Sicker. Physical layer attacks on unlinkability in wireless LANs. In *Proceedings of PETS*, 2009.

[13] W. Hu, D. Willkomm, L. Chu, M. Abusubaih, J. Gross, G. Vlantis, M. Gerla, and A. Wolisz. Dynamic frequency hopping communities for efficient ieee 802.22 operation. *IEEE Communications Magazine*, 45:80–87, 2007.

[14] T. Jiang, H.J. Wang, and Y. Hu. Preserving location privacy in wireless LANs. In *Proceedings of MobiSys*, pages 246–257, 2007.

[15] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis. *ACM Mobile Networks and Applications*, 10(3):315–325, 2005.

[16] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proceedings of MobiCom*, pages 99–110. ACM Press, 2007.

[17] S. Lakshmanan, C.L. Tsao, R. Sivakumar, and K. Sundaresan. Securing wireless data networks against eavesdropping using smart antennas. In *Proceedings of ICDCS*, 2008.

[18] A. Sheth, S. Seshan, and D. Wetherall. Geo-fencing: Confining wi-fi coverage to physical boundaries. In *Proceedings of Pervasive Computing*, 2009.

[19] I. Martinovic, P. Pichota, and J.B. Schmitt. Jamming for good: design and analysis of a crypto-less protection for WSNs. In *Proceedings of WiSec*, 2009.

[20] http://madwifi-project.org/.

[21] http://research.microsoft.com/en-us/projects/virtualwifi/.

[22] T.T. Nguyen and G. J. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials*, 10(1-4):56–76, 2008.

[23] J. Wilson and N. Patwari. See through walls: Motion tracking using variance-based radio tomography networks. *IEEE Transactions on Mobile Computing*, 2010.

[24] K. Kowalik, M. Bykowski, B. Keegan, and M. Davis. Practical issues of power control in IEEE 802.11 wireless devices. In *Proceedings of International Conference on Telecommunications*, pages 1–5. IEEE, 2008.

[25] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: automated construction of application signatures. In *Proceedings of the 2005 ACM SIGCOMM workshop on mining network data*, pages 197–202. ACM, 2005.

[26] F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *Proceedings of Computer and Communications Security*, pages 48–56, 1997.

[27] X. Song, D. Wagner, S. David, and X. Tian. Timing analysis of keystrokes and timing attacks on SSH. In *Proceedings of USENIX Security Symposium*, 2001.

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

[29] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine. Privacy vulnerabilities in encrypted HTTP streams. In *Proceedings of Privacy Enhancing Technologies Workshop*, pages 1–11, 2005.

[30] M. Liberatore and B. Levine. Inferring the source of encrypted http connections. In *Proceedings of Computer and Communications Security*, 2006.

[31] http://crawdad.cs.dartmouth.edu/.

[32] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach. Wireless LAN location-sensing for security applications. In *Proceedings of WiSE*, 2003.

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

[34] S. Kandula, K. C. Lin, T. Badirkhanli, and D. Katabi. Fatvap: Aggregating ap backhaul capacity to maximize throughput. In *Proceedings of NSDI*, 2008.

[35] R. Chandra and P. Bahl. MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card. In *Proceedings of INFOCOM*, 2004.

[36] O. Barak, R. Friedman, and G. Kliot. PeerBooster: Enhancing Throughput in Wi-Fi Networks Through Network Virtualization. www.cs.technion.ac.il/ gabik/publications/PeerBoost-submitted.pdf.

[37] P. Bahl, R. Chandra, P.P.C. Lee, V. Misra, J. Padhye, D. Rubenstein, and Y. Yu. Opportunistic use of client repeaters to improve performance of WLANs. *Networking, IEEE/ACM Transactions on*, 17(4):1160–1171, 2009.