VR-Defender: Self-Defense Against Vehicular Rogue APs for Drive-Thru Internet

Hao Han, Fengyuan Xu, *Member, IEEE*, Chiu C. Tan, *Member, IEEE*,
Yifan Zhang, and Qun Li, *Senior Member, IEEE*

***Abstract*—This paper considers the problem of vehicular rogue access points (APs) for drive-thru Internet. Vehicular rogue APs are set up in moving vehicles to mimic legitimate roadside APs so as to lure users into associating with them. Due to the mobility, a vehicular rogue AP is able to maintain a long connection with users, which gives the adversary more time to launch various attacks and steal users' sensitive data. We propose a practical user-side detection scheme to prevent users from connecting to vehicular rogue APs without the help of a network administrator. In our solution, each AP broadcasts its GPS location; thus, a vehicular rogue AP has to forge its location to evade detection. A lie detector algorithm based on information collected and exchanged by clients is then used to validate whether the reported location is fake, aiming to detect the rogue AP. We have implemented the prototype and evaluated it on commercial off-the-shelf devices. We observed that our scheme can achieve more than 90% accuracy in real-world experiments.**

***Index Terms*—Drive-thru Internet, IEEE 802.11 wireless networks, rogue access points (APs).**

I. Introduction

THE *drive-thru Internet* [1] has been of special interest in the wireless communication research community for several years. The goal of drive-thru Internet is to provide seamless Internet access to mobile users in moving vehicles by exploiting IEEE 802.11 technology [2], [3]. Fig. 1 shows a typical drive-thru Internet scenario, where IEEE 802.11-based access points (APs) are deployed along the road—within the city or on a freeway. Mobile users in vehicles (i.e., vehicular clients) connect to these APs (so called *roadside APs*) for Internet access. Recently, many citywide Wi-Fi infrastructures for drive-thru Internet have already been deployed in the real world. For example, Google provides a free wireless Internet service to Mountain View, CA, USA [4].

Manuscript received August 14, 2013; revised February 3, 2014; accepted February 8, 2014. Date of publication March 11, 2014; date of current version October 14, 2014. This work was supported in part by the U.S. National Science Foundation under Grant CNS-1320453, Grant CNS-1117412, and CAREER Award CNS-0747108. The review of this paper was coordinated by Prof. Y. Zhang.

H. Han is with the Networks and Security Group, Intelligent Automation, Inc., Rockville, MD 20855 USA (e-mail: hhan@i-a-i.com).

F. Xu is with the Storage System Group, NEC Laboratories America, Princeton, NJ 08540 USA (e-mail: fxu@nec-labs.com).

C. C. Tan is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: cct@temple.edu).

Y. Zhang and Q. Li are with the Department of Computer Science, College of William and Mary, Williamsburg, VA 23187-8795 USA (e-mail: yzhang@cs.wm.edu; liqun@cs.wm.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2014.2311015

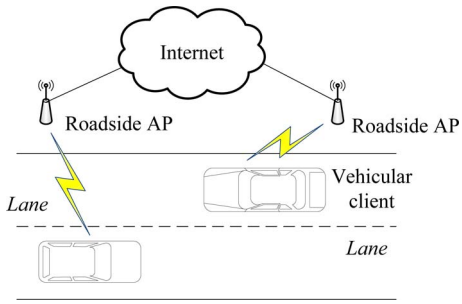


Fig. 1. Typical drive-thru Internet scenario.

Due to the ubiquitous deployment of APs, the problem of rogue APs has emerged as a well-recognized security threat. A rogue AP refers to a malicious AP that pretends to be a legitimate AP to induce users to connect. Once an innocent user has associated to a rogue AP, then the adversary can launch various attacks to steal the user's secrets. For example, the adversary can launch phishing attacks to redirect the user's web page requests to fake ones, seeking to steal bank account numbers and passwords.

Rogue APs in vehicular networks can be broadly classified into two categories: *stationary* and *mobile*. In the first category, a stationary rogue AP is set up at a fixed place, such as in a building facing a busy road. Due to the mobility of vehicular clients, this type of rogue AP is unlikely to keep a long connection with the clients, and the time window for adversaries to steal users' secrets is short. As a result, the damage of stationary rogue APs for drive-thru Internet is restricted. Furthermore, a stationary rogue AP usually keeps active for a long time in a place. It is relatively easy for authorities to detect such a rogue AP. Previous work [5]–[15] has already proposed several methods for detecting stationary rogue APs. However, there is little work on how to defend against a mobile rogue AP, where the malicious AP is set up in a moving vehicle. Since a vehicular rogue AP can follow traffic on a road, such an AP is able to maintain a long connection, which gives the adversary more time to launch various attacks. Therefore, this type of rogue APs is more dangerous.

In this paper, we consider the problem of detecting vehicular rogue APs from the user's perspective. This is challenging because the duration for a vehicular client connected to an AP is short, so that the time left for detection is restricted. According to IEEE 802.11 standards, when the signal strength of a connected AP is less than a threshold, the client will perform a handoff [16] and reassociate to another AP with the strongest signal strength. It is meaningless to determine whether an AP

is rogue while such an AP is out of the client's reach. Another challenge is that the information obtained by clients has to go through the associated AP. To evade detection, a rogue AP can impersonate a legitimate AP by providing any fake information such as medium access control (MAC) address, basic service set identification (SSID), and configurations. Clients cannot rely on the information obtained from the untrusted AP to detect a rogue AP.

To address these challenges, we propose a novel solution that prevents users from connecting to vehicular rogue APs. In our solution, the GPS location of each AP is added to its beacon frames, thus forcing a vehicular rogue AP to report a fake location. We then exploit the received signal strength (RSS) and test messages with changing transmission (TX) power and TX rate to detect the rogue AP. To the best of our knowledge, we are the first to consider the vehicular rogue AP problem and propose a pure user-side detection scheme. Our main contributions are summarized as follows.

1) We are the first to study the vehicular rogue AP problem. This paper lays down the foundation for future research in this area to improve the vehicular network security.
2) We creatively propose using the geographic location of each AP to detect rogue APs and design a client-side scheme to perform the location validation through collecting RSS and tweaking TX power and TX rate.
3) We implement our scheme on commercial off-the-shelf devices. The real-world experiments show the efficacy of our approach on realistic road conditions.

The rest of this paper is organized as follows. Section II discusses the related work. Section III describes the adversary model. Our detection algorithms are detailed in Section IV. The implementation and evaluation are presented in Section V. Finally, we discuss the limitation of our solution in Section VI and conclude in Section VII.

II. Related Work

The threat of rogue APs has attracted the attention of both industrial and academic researchers. Previous research has been mainly focused on detecting static rogue APs in enterprise or hotspot scenarios. Existing schemes can be broadly classified into three categories.

The first category relies on sniffers to monitor wireless traffic. These sniffers usually scan spectrum to examine the 2.4- and 5-GHz spectra. Once traffic is detected from unauthorized APs, they will alert the administrator. This approach usually demands a well-controlled infrastructure such as enterprise networks, where the administrator can easily deploy sniffers and cut off the access of rogue APs to Internet. Some commercial products such as that in [17] have been developed using this technique. Previous studies [11], [14] have proposed using desktop machines to perform radio-frequency monitoring. Different from this type of solution, our scheme does not reply on sniffers. This is because a small amount of sniffers may not catch vehicular rogue APs well and extensive deployment of sniffers on the road is impractical.

The technique used in the second category leverages fingerprints to identify rogue APs. Since an advanced adversary can easily spoof a rogue AP's MAC address, SSID, vendor name, and configuration to escape from the detection, the previous work often adopted fingerprints that cannot be easily forged. For example, the work in [18] uses every AP's clock skew calculated by beacon frames and probe responses to identify rogue APs. In addition to clock skew, RSS values [19] and radio frequency variations [20] are also used. However, a major drawback of this type of a scheme is that the AP validation requires access to the database containing the fingerprints of all legitimate APs. This database may not be available for end users before they connect to the AP. Our solution differs from such schemes in that we do not assume that the clients know the fingerprints of legitimate APs in advance. Therefore, our detection scheme can apply to not only network administrators but also to end users who use the wireless network for the first time.

The last category exploits the features of wireless traffic to detect the presence of rogue APs [5], [7], [8], [10], [21]. In [5], a practical timing-based scheme that measures the round trip time between end users and a local domain name system server is proposed. The work in [12] utilizes the immediate switch connecting rogue APs to measure the round-trip time of transmission control protocol traffic. Other studies [13], [15] use the spacing between packets to distinguish wireless networks from wired networks. In [8], interarrival time of an ACK pair is used to detect rogue APs. In [10], a wired verifier and wireless sniffers are deployed at the same time to detect rogue APs. All these approaches are used to detect static rogue APs in traditional wireless local area networks. It is unclear whether they work in vehicular networks. Additionally, analyzing the network traffic is time-consuming, whereas our solution is more efficient.

III. Adversary Model

The goal of vehicular rogue APs is to induce clients to connect. If any client associates to a vehicular rogue AP, the adversary succeeds. The following assumptions are considered in this paper.

First, we assume that the vehicular rogue APs and vehicular clients move toward the same direction on the road. The vehicular rogue APs on the opposite lane are not considered because, although some clients may associate to them by accident, the time window for them to launch further attacks is extremely limited.

Second, previous work [5] assumes that rogue APs have to connect to existing legitimate APs to access Internet; therefore, they suffer from the delay of multihop wireless transmission and are thus slower than legitimate APs. Here, we do not hold this assumption, as well as any other assumptions about the back-end infrastructure for vehicular rogue APs to access Internet.

Third, we assume that rogue APs can transmit any packet with arbitrary TX power and any content. For example, the adversary can generate fake reassociate frames to force clients to reselect APs immediately. However, rogue APs cannot modify the firmware of wireless cards. In other words, the adversary cannot control acknowledgment (ACK) frames. Although ACK frames can be generated by software, software ACK cannot be sent back to clients within the timeout [22].

Finally, we do not consider that the drive-thru Internet is capable of using RADIUS-based 802.1X authentication of users since it requires each vehicle to have the correct key to access the network. While this may be possible, for example, every car when registered with the Department of Motor Vehicles is issued the appropriate credentials, it is unclear how well this will work in practice. Thus, we only consider the open 802.11 network where any car can connect.

Based on these assumptions, the adversary can launch two types of attacks.

Basic Attack: In the basic attack, the vehicular rogue broadcasts beacons with the maximum TX power. This is because the maximum power will lead to the strongest signal strength with which the vehicular rogue has most probability to attract users to connect. The advantage of the basic attack is its easy setup. Without any complicated configuration, the basic attack can be launched anytime and anywhere.

Advanced Attack: The major difference between the advanced attack and the basic attack is that the former attack needs background preparation before creating a vehicular rogue AP. For example, the adversary first needs to drive along the road as a client to profile RSS values from existing roadside APs, and then tune TX power to make the signal strength received by nearby clients similar to the profiled values. By doing so, RSS-based solution cannot work because the RSS values measured by the clients appear to be "real." Compared with the basic attack, the advanced attack is more time-consuming but is more difficult to detect.

IV. Our Solution

The algorithms for detecting basic attacks and advanced attacks are presented separately. In both algorithms, each AP needs to broadcast its physical location such as GPS coordinates in beacon frames. Since the IEEE 802.11 standards allow adding arbitrary information elements with variable length in beacons, the GPS information can be easily included in beacons, which are broadcast periodically. To determine the AP's location, the network administrator can resort to an external GPS module and measure it offline. Upon receiving a beacon, clients know the AP's location. If any AP refuses to expose its GPS location, the client will never connect to that AP for security consideration. For a legitimate AP, it always reports its actual GPS coordinates, which indicate a fixed location alongside the road. However, a vehicular rogue AP is afraid of reporting its true GPS location because such a location indicates the AP is moving on the road. A user can easily detect it. Hence, a vehicular rogue AP has to forge its location to evade detection. Thus, detecting rogue APs can be converted to verifying whether an AP reports a fake GPS location.

A. Defending Against Basic Attacks

First, we present the algorithm to defend against the basic attacks. In this algorithm, clients measure the RSS of each beacon to verify whether the rogue AP lies on its location. Since the rogue AP is not physically at the reported location, there should be inconsistency in RSS. As shown in Fig. 2, a vehicular client moves on a road and receives m beacons. The RSS value

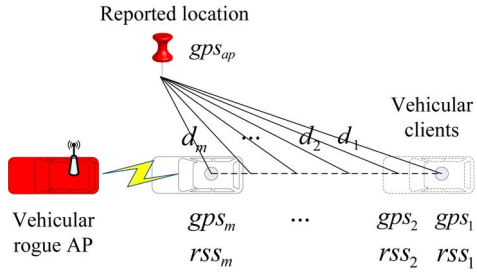


Fig. 2. Intuition of defending against basic attacks, where a vehicular client needs to measure RSS and its own GPS location at multiple places.

of each beacon is denoted rss_i, where $i \in [1, \ldots, m]$, and gps_i indicates the exact location where the beacon is received. Thus, based on AP's reported location $\mathrm{gps}_{\mathrm{ap}}$ and its own gps_i, the client can continuously compute the distance between the AP and itself (i.e., $d_1, d_2, \ldots, d_m$). On the other hand, the distance can be also inferred from RSS values. If the distances computed from GPS coordinates significantly differ from those inferred from RSS values, that AP is likely to be a vehicular rogue AP.

Inferring the Distance From RSS: We adopt a widely used log-distance propagation model to characterize the relationship between distance and RSS. In this model, the received signal power decreases logarithmically with distance, and it is expressed as

$$P_r(d) = \frac{c \cdot P_t}{d^{\gamma}} \tag{1}$$

where P_t refers to the transmit power of the sender, $P_r(d)$ is the RSS at a distance of d, and c is a correction constant that captures the effects of transmit frequency, antenna gains of both the sender and the receiver, and other factors in the environment. The *path-loss exponent* γ determines the rate of attenuation when the signal propagating through the space, which is dependent on the propagation environment. A larger γ means the environment is lossy and will cause the fall of RSS faster with distance.

In the scale of decibels (dB), (1) can be rewritten as

$$P_r(d) = P_t + c - 10\gamma \log_{10}(d) + X_{\delta} \tag{2}$$

which depicts a linear relationship between the RSS and the logarithmic distance. The newly introduced X_{δ} is a random variable with zero mean, which reflects the attenuation caused by flat fading [23]. Such a log-distance model has been verified to predict RSS well in various locations, including an outdoor scenario by previous research [24] and our empirical experiments. The model parameters (i.e., c and γ) are adjusted according to specific outdoor space. As in [24] and [25], γ normally ranges from 2 to 6, where 2 is for propagation in free space and 6 is for heavily lossy environment.

Overview of Algorithm: The overview of the algorithm is described in Algorithm 1. For each AP that a client tries to connect, this algorithm is used to test whether such an AP is a rogue AP. The inputs of the algorithm are $\mathrm{gps}_{\mathrm{ap}}$ and time-series data rss_i and gps_i for $i \in [1, 2, \ldots, m]$. Among these inputs, $\mathrm{gps}_{\mathrm{ap}}$ denotes the reported location of the tested AP, rss_i is the RSS of the ith beacon, and gps_i is the location of the client where the ith beacon is received. Parameter m determines the number of samples. Let us first assume that all these inputs are available. Later, we describe how to obtain them. The output of

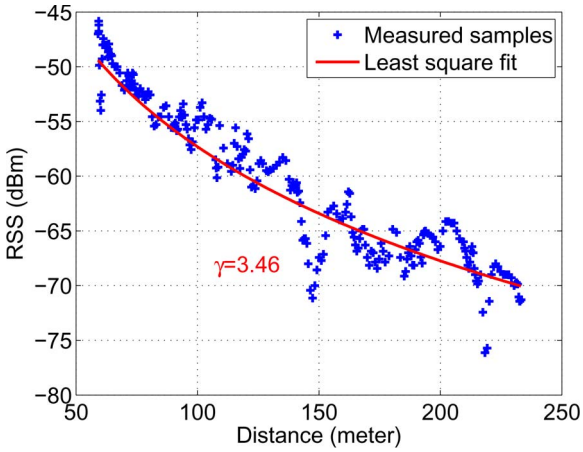


Fig. 3. Least squares fitting of the measured samples collected from a real roadside AP.

the algorithm is either *true* or *false*, which means the tested AP is either a rogue AP or not. At the beginning of the algorithm, the client computes distance d_i between AP's reported location and its own. Next, rss_i and d_i are fit into (2) to estimate the parameter γ using the least squares method. If the value of γ is within the normal range of 2–6, the algorithm terminates and returns false; otherwise, it returns true.

Algorithm 1 Detecting Basic Rogue APs

Input: $\mathrm{gps}_{\mathrm{ap}}$, rss_i, and gps_i where $i \in [1, 2, \ldots, m]$
Output: *true* or *false*

1: **for** $k = 1$ to m **do**
2: $\quad d_i = \|\mathrm{gps}_{\mathrm{ap}} - \mathrm{gps}_i\|$
3: **end for**
4: Use least squares fitting of $\langle \mathrm{rss}_i, d_i \rangle$ to estimate γ
5: **if** γ falls into the range from 2 to 6 **then**
6: $\quad$ Return false
7: **else**
8: $\quad$ Return true
9: **end if**

Fig. 3 shows an example of the least squares fitting of the samples collected from a real roadside AP. As shown, the estimated parameter γ indeed falls into the normal range from 2 to 6. We also tested many other APs using different devices and in varying environment. They all have similar results, but vehicular rogue APs yield out-of-range results. The results are presented in Section V. It is worth noting that the inaccurate GPS and fluctuating RSS values will not affect our algorithm in practice because the least squares method can minimize the noise of reading over multiple samples.

Matching GPS and RSS: Here, we present how to obtain rss_i and the corresponding gps_i. In practice, the frequency of GPS update is much slower than that of receiving a beacon. For example, most commercial GPS modules update GPS coordinates by less than 1 Hz. However, a client may receive about ten beacons within 1 s. Thus, many beacons may have obsolete GPS values. To address this problem, we assume the client did not change the speed during the interval of GPS update and apply interpolation on GPS data. This is true because the interval is very small so that the speed diversity can be ignored. Suppose two GPS updates occur at timestamps t_i and t_j, and the coordinates change from $\mathrm{gps}(t_i)$ to $\mathrm{gps}(t_j)$. Between them, the client receives several beacons with rss_k for $i \le k \le j$. To estimate $\mathrm{gps}(t_k)$, we have

$$\mathrm{gps}(t_k) = \mathrm{gps}(t_i) + (\mathrm{gps}(t_j) - \mathrm{gps}(t_i)) \cdot \frac{t_k - t_i}{t_j - t_i}.$$

Dealing With RSS Noise: Due to many reasons such as dynamic environment interference, measured RSS may suffer from some extreme values. With these abnormal values, the model parameters may be estimated inaccurately. Hence, we filter them out before applying the least squares method. The filtering is performed as follows. The client checks consecutive three RSS values. If the difference between the median value and the average of its prior and following values exceeds threshold τ, the median value is replaced by the average value. In this paper, τ is heuristically set to 5 dBm.

Determining the Number of Samples: The number of samples m affects the time duration and the accuracy of the detection. A larger m typically leads to a more accurate result but costs more time to finish. To determine m, we propose a method that can adaptively involve more samples until the estimated parameter γ becomes stable. In the approach, once an AP is first discovered, the client keeps measuring RSS in background. For every time duration Δt (in seconds), all the accumulated RSS values for that AP are fitted into (2) to estimate γ. If γ does not change within threshold θ for continuous two Δt, γ is deemed to be stable, and we stop to involve more samples. This method works well in practice by setting $\Delta t = 1$ and $\theta = 0.5$. The experiment results are presented in Section V.

B. Defending Against Advanced Attacks

As in Section III, once profiling in advance, a sophisticated rogue AP is able to tune TX power to mimic the RSS trend of a roadside AP. In this case, the clients cannot merely rely on RSS to detect rogue APs because the RSS values appear to be "real." Hence, the aforementioned algorithm for basic attacks cannot work. It is more challenging to defend against advanced attacks than basic attacks because the adversary is well prepared. Here, we present our novel solution. In our approach clients send multiple probe requests to the AP with different TX power levels and TX rates and then perform "lie-detection" tests. Note that this approach is complementary to the previous algorithm. They can be combined together to make the detection more accurate.

Background: Before presenting the detail, let us first introduce some background knowledge. In wireless communication, the transmission distance that can be achieved between two wireless devices is influenced by the TX power of the sender and the receiving (RX) sensitivity of the receiver. As a sender, there are two ways to adjust the TX distance in purpose. Tweaking TX power is one of them. Fig. 4 shows RSS values versus various TX power levels at a distance of 1 m between a sender and a receiver in our experiments. As shown, a large TX power level can increase the long communication distance.

The RX sensitivity is another factor that may affect the TX distance. With greater RX sensitivity, the device is able to receive weaker signals, leading to a longer TX distance.

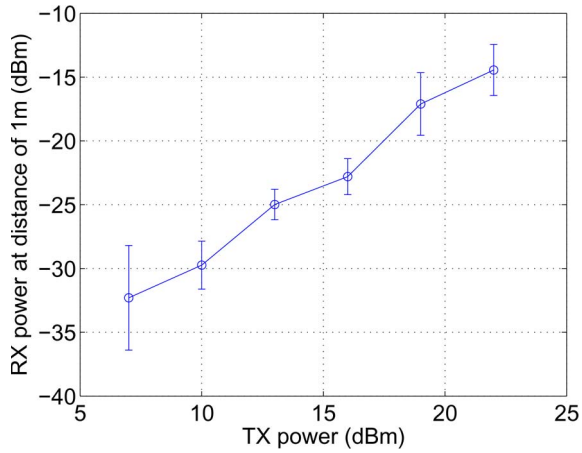


Fig. 4. Average RSS values versus varying TX power levels.

TABLE I
MODULATIONS FOR 802.11a/g

Rates (Mbps)	Standard	Modulation	Rx sensitivity (dBm)
6	802.11a/g	BPSK/OFDM	-82
9	802.11a/g	BPSK/OFDM	-81
12	802.11a/g	QPSK/OFDM	-79
18	802.11a/g	QPSK/OFDM	-77
24	802.11a/g	QAM-16/OFDM	-74
36	802.11a/g	QAM-16/OFDM	-70
48	802.11a/g	QAM-64/OFDM	-66
54	802.11a/g	QAM-64/OFDM	-65

However, if RX sensitivity is a metric to the receiver, how can a sender leverage RX sensitivity to change the communication distance? As shown in Table I, different TX rates require different RX sensitivity levels. A packet with a large TX rate typically demands high RX sensitivity for a successful demodulation. Thus, the sender can achieve a different communication range by controlling its TX rate.

Intuition: As previously mentioned, given certain TX power and TX rate, the communication range between the client and the AP is determined. If the reported location yields a significant departure from that range, there should be something wrong. For example, an AP claims far away from the client, but it can receive packets with very low TX power and high TX rates. Such an AP is likely to be a rogue AP. The client herself could perform this test according to the reported location of the AP. However, the problem is that different APs may have different RX sensitivity for each data rate, which may affect the correctness of the test. For instance, the success in receiving a packet with low TX power and a high data rate may be due to a powerful antenna rather than a short distance. To cope with this problem, we choose a different strategy that the client will test each AP with a vector of power-and-rate combinations from the low-power/high-rate end to the high-power/low-rate end. In this vector, the AP could receive the packets at the beginning but could fail after a certain combination. Such a combination is labeled as the boundary of the vector, which reflects the communication distance. Based on the AP's reported location, if the client finds that the reported distance changes a lot but the boundary barely changes, then such an AP is deemed a rogue AP. The rationale behind this idea is that the relative distance between a legitimate roadside AP and a client changes when the client is moving on the road, but the actual distance between a mobile rogue AP and the client never changes so much.

Overview of the Algorithm: Algorithm 2 describes the detail. The procedure *send_probe_requests* is used to send probe requests with varying TX power levels and TX rates to test an AP. Instead of exhausting all the combinations, we only pick a subset to reduce the test time. Suppose the maximum TX power of a client is $\mathrm{pwr}_{\max} = 27$ dBm, then the power set is {7, 17, 27 dBm}. The selected rate set is always limited to {54, 48, 36, 24 Mb/s} since it can already separate the 10-dBm difference. The vector of combinations is as follows

$$\{7\ \mathrm{dBm}/54\ \mathrm{Mb/s}, 7\ \mathrm{dBm}/48\ \mathrm{Mb/s}, 7\ \mathrm{dBm}/24\ \mathrm{Mb/s}, \ldots, 27\ \mathrm{dBm}/48\ \mathrm{Mb/s}, 27\ \mathrm{dBm}/24\ \mathrm{Mb/s}\}.$$

It should be noticed that, according to our experiments, the current ordering of the vector strictly follows the communication distance from short to long. For each combination, the client sends several probe requests to the AP. If more than half of packets got ACK back, such a combination is deemed as *receivable* or, else, *nonreceivable*, which is denoted 1 or 0, respectively. Based on this vector, the boundary is computed such that the ratio of 1 to the number of elements before the boundary and the ratio of 0 to the number of elements after the boundary are maximized. In our algorithm, the client will call this procedure twice: once at the time when first discovering the AP (weak RSS), and the other time when trying to associate to that AP (strong RSS). By comparing the boundaries of two vectors, if the difference does not exceed a threshold, the client will never connect to that AP because the physical distance has changed a lot but the boundary difference never reflects this change. If not, such an AP is deemed a rogue AP. In this paper, we heuristically set this threshold to 4.

Algorithm 2 Detecting Advanced Rogue APs

procedure: send_probe_requests
$\mathrm{pwrset} \leftarrow \{\mathrm{pwr}_{\max}\%10, \mathrm{pwr}_{\max}\%10+10, \ldots, \mathrm{pwr}_{\max}\}$
$\mathrm{rateset} \leftarrow \{54, 48, 36, 24\}$
for $i = 1$ to $\mathrm{size(pwrset)} * \mathrm{size(rateset)}$ **do**
  Set TX power to $\mathrm{pwrset}[\mathrm{ceil}(i/\mathrm{size(rateset)})]$
  Set data rate to $\mathrm{rateset}[i\%\mathrm{size(rateset)}]$
  Send n probe requests to AP with TX power and data rate above. If more than half receive ACK back, then $V[i] = 1$; otherwise, $V[i] = 0$
end for
if AP is first discovered (weak RSS) **then**
  Call **send_probe** to obtain vector V
end if
if client intends to associate to that AP (strong RSS) **then**
  Call **send_probe** to obtain another vector V'
end if
Find the boundary in both V and V' such that the ratio of 1 to the number of elements before the boundary and the ratio of 0 to the number of elements after the boundary are maximized.
if $\mathrm{boundary}(V) - \mathrm{boundary}(V') < \mathrm{threshold}$ **then**
  Such an AP is a rogue AP
end if

Fig. 5. Experimental setups of (left) a roadside AP and (right) a vehicular rogue AP.

V. Evaluation

Here, we present our experimental setup, the methodology, and the experimental results, which attempt to answer the following questions: 1) What is the performance of both basic algorithm and advanced algorithm working in practice? 2) What is the time cost of determining whether an AP is a rogue AP? 3) How does the speed of a vehicle affect the performance?

A. Experimental Setup and Methodology

The devices used in our experiments are comprised of a roadside AP, a vehicular rogue AP, and a vehicular client.

Roadside AP: A commercial outdoor AP (Deliberant CPE 2-12) was configured as a roadside AP. The specification of this model can be found in [26]. The AP was mounted on top of a tripod that is 2 m high (see Fig. 5, left side). When deploying the AP alongside the road, we used a GPS receiver (GlobalSat BU-353) to measure its physical location. To enable broadcasting the GPS information via beacons, we loaded the AP with OpenWrt [27] firmware and a modified Wi-Fi driver. The extra content in each beacon has 18 B including 1-B element ID, 1-B length, 8-B latitude, and 8-B longitude.

Vehicular Rogue AP: A laptop connected with an external omniantenna and a GPS receiver (see Fig. 5, right side) mounted on the roof of a car was configured as a vehicular rogue AP. The laptop was running a 2.6.27-generic Linux kernel with madwifi driver (svn r4128). Similar to the roadside AP, we modified the madwifi driver to support GPS broadcast. We did not set up the Internet access for all APs since it does not affect the performance of our algorithms. In basic attacks, we fixed the TX power by executing command `iwconfigtxpower[value]` with the maximum power value. In advanced attacks, we tried to automatically adjust TX power to mimic the real trend of RSS values, but eventually, we found that it was really difficult to make it work in practice. Most of the time, the rogue AP could be detected by our basic algorithm. To ease evaluation, we optimistically assume that the rogue AP can bypass our basic algorithm, and we investigate the advanced algorithm without changing the TX power of the AP. This is correct since our algorithm does not rely on any configuration of APs.

TABLE II
Equipment Description

Name	Notation
AP	Deliberant CPE 2-12 based on WDB-500 platform
Laptop	Lenovo T61 with 2.0GHz processor and 1G RAM
GPS	GlobalSat BU-353 USB GPS receiver
Wireless card	CB9-GP Cardbus 802.11a/b/g based on Atheros chipset
Antenna	7 dBi MA24-7N magnetic-mount omnidirectional

Vehicular Client: The vehicular client used the same hardware as the vehicular rogue AP. The Wi-Fi interface of the client was set to monitor mode, which could capture all the packets in air. Injecting and receiving packets were achieved by libpcap. The control of per-packet TX power and TX rate was done by a radiotap header. In Linux, the IEEE 802.11 MAC layer allows arbitrary injected packet composed in the following format:

$$[\text{radiotap header}] + [\text{ieee80211 header}] + [\text{payload}].$$

`IEEE80211_RADIOTAP_RATE` and `IEEE80211_RADIOTAP_DMB_TX_POWER` in the radiotap header are used to control the data rate and the TX power of injected packets. Given different values, a packet can be transmitted with the desired power and data rate. Note that to control per-packet TX power `hal_tpc` must be enabled while loading the madwifi module. Table II summarizes all the equipment used in our experiments.

The experiments were conducted in a suburb an area, where we could freely drive along the road and stop to collect measurements. In the experiments, the roadside AP was placed in a parking lot around 60 m away from the road. Two cars configured to be a vehicular rogue APs and a vehicular client were driven along the road passing through the roadside AP. The roadside AP broadcast its actual GPS location, and the rogue AP broadcast a location close to the roadside AP. We took two sets of experiments to evaluate our vehicular rogue AP detection schemes. The first set of experiments was used to evaluate the performance of our basic algorithm, where the client passively listened to the beacons. The second set was to evaluate the advanced algorithm, where the client actively sent probe requests to the AP.

B. Experimental Results

Basic Attack Evaluation: First, we tested whether legitimate roadside APs could pass our basic algorithm. As an example, the top of Fig. 6 shows the measured RSS values against the logarithmic distance in an experiment. The client started the algorithm when observing the first beacon from a roadside AP and terminated the algorithm when γ was stable. In total, the client collected 110 beacons within 11 s. The estimated γ was 3.31 eventually, which falls into the valid range from 2 to 6. Therefore, the AP is correctly labeled as a legitimate AP. Although the finish time seems a little long, it should be noticed that this cost only occurs once when the client initially turns on the Wi-Fi and tries to find an AP to connect. After that, the client will wait for a certain period until the signal strength of current APs becomes weak. Only at that time the client needs to find another AP for handoff. During the waiting period, the client should have collected enough RSS values from nearby APs to determine which APs are rogue APs. To investigate the robustness of our algorithm, we also conducted experiments

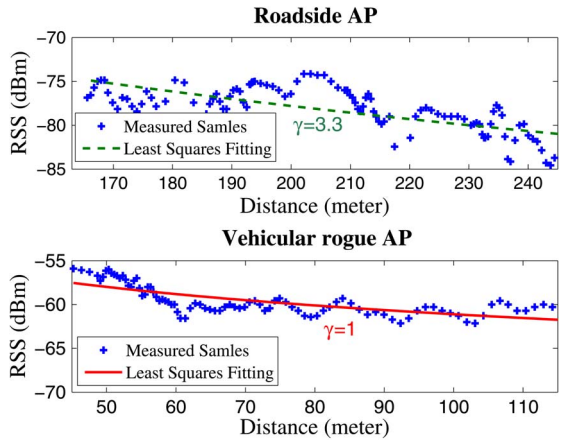


Fig. 6. Results of our basic algorithm.

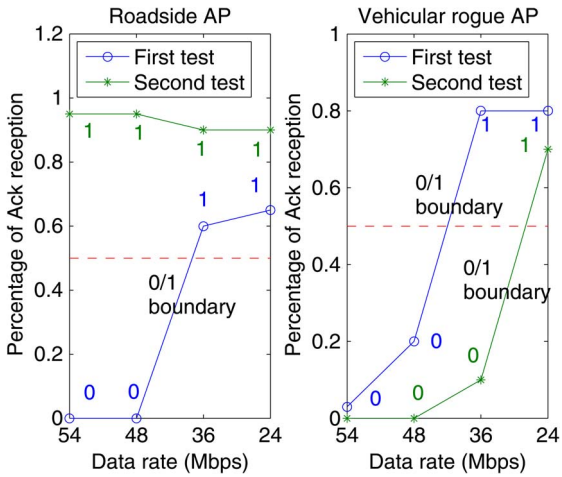


Fig. 7. Results of our advanced algorithm.

in different environments. We observed that γ varied across environments but they all fell into the normal range.

Next, we tested the performance of the algorithm when the tested AP is a rogue AP. The bottom of Fig. 6 shows the result. As shown, after collecting 90 beacons, γ was stable at 1. It is clear that the tested AP is a rogue AP.

Advanced Attack Evaluation: Fig. 7 shows the experimental results of our advanced algorithm with respect to a legitimate roadside AP and a vehicular rogue AP. The figure only shows part of the vector that contains the 0/1 boundary. The first test occurred when the first beacon was received, and the second test was performed when the client tried to associate to that AP. It is seen that the boundary for a roadside AP changed significantly due to the mobility of the vehicle. By contrast, the boundary changed little when the tested AP is a rogue AP. We also evaluated the performance at different locations. We observed that our algorithm could achieve more than 90% accuracy to correctly label an AP. Almost every incorrect detection was a false positive that the roadside AP was falsely detected as a rogue AP. After analyzing the captured traces, we found that it was caused by packet loss due to wireless interference. In our future work, we will investigate how to identify the interference to improve the accuracy.

Finish Time Versus Vehicle Speed: The finish time of our basic algorithm is determined by the duration when γ becomes

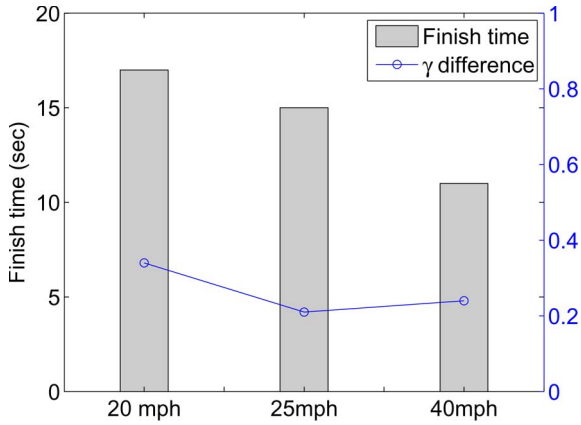


Fig. 8. Finish time and γ difference versus the vehicle speed.

stable. We have investigated the finish time and the difference of γ (the value when the algorithm is terminated to the ground truth) against different vehicle speeds. The finish time is measured in unit of seconds, and the ground truth of γ is derived when all beacons (from entering to leaving the communication range of the AP) are used for least squares fitting. Fig. 8 shows the results. As shown, the faster the speed of the vehicle, the quicker the algorithm can finish. Again, this cost is only incurred when the Wi-Fi interface is initially turned on. After that, the cost can be amortized by background scan. In addition, we find our algorithm can estimate γ accurately. The difference of γ is within 0.3.

VI. Discussion

Our solution makes use of physical characteristic such as path loss and the percentage of acknowledged TX packets to determine the discrepancy between the rogue AP's actual location and its reported location. This makes our solutions vulnerable to the following factors.

One factor is that outdoor wireless condition is unpredictable. Although our solution relies on well-known signal propagation models, it is inevitable that there will be instances where the actual conditions deviate from the models. When this happens, our solution is not able to detect the rogue AP successfully. We can mitigate this by adopting a more accurate model in our solution. In addition, it is unclear how well the adversary can take advantage of this limitation since the adversary is unable to predict the channel conditions as well. Finally, we have observed that our solutions may not work as well in some locations where there are several buildings. We will investigate more complex environments and refine our algorithms in our future work.

Another factor is that the wireless interference may have a negative effect on the packet reception. It is difficult for a sender to infer that an unsuccessful packet transmission is caused by either bad signal strength or interference. When this happens, our advanced algorithm may not detect the rogue AP well. However, the adversary cannot easily utilize this uncontrolled factor to increase the probability of escaping from the detection. In the future work, we will study how to reduce the detection errors caused by the interference.

VII. Conclusion

The ease of setting up a successful rogue AP in vehicular makes this form of wireless attack a particularly serious security problem in vehicular networks. In this paper, we are the first to demonstrate the feasibility of this type of rogue AP and present practical defending schemes to prevent the users from connecting to vehicular rogues. We implement our approach on commercially available hardware and perform extensive real-world experiments to evaluate our solutions.

References

[1] Drive-thru Internet. [Online]. Available: http://www.drive-thru-Internet.org/

[2] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proc. 12th ACM MOBICOM Conf.*, Los Angeles, CA, USA, Sep. 2006, pp. 50–61.

[3] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.11b for 'automobile' users," in *Proc. IEEE INFOCOM*, 2004, pp. 362–373.

[4] Google Mountain View. [Online]. Available: http://wifi.google.com

[5] H. Han, B. Sheng, C. Tan, Q. Li, and S. Lu, "A measurement based rogue AP detection scheme," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009, pp. 1593–1601.

[6] H. Han, B. Sheng, C. C. Tan, Q. Li, and S. Lu, "A timing-based scheme for rogue AP detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 11, pp. 1912–1925, Nov. 2011.

[7] L. Ma, A. Y. Teymorian, and X. Cheng, "A hybrid rogue access point protection framework for commodity Wi-Fi networks," in *Proc. INFOCOM*, 2008, pp. 1894–1902.

[8] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, and D. Towsley, "Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs," in *Proc. IMC*, 2007, pp. 365–378.

[9] A. Venkataraman and R. Beyah, "Rogue access point detection using innate characteristics of the 802.11 MAC," in *Proc. SecureComm*, 2009, pp. 394–416.

[10] H. Yin, G. Chen, and J. Wang, "Detecting protected layer-3 rogue APs," in *Proc. BROADNETS*, 2007, pp. 449–458.

[11] A. Adya, P. Bahl, R. Chandra, and L. Qiu, "Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks," in *Proc. MOBICOM*, 2004, pp. 30–44.

[12] L. Watkins, R. Beyah, and C. Corbett, "A passive approach to rogue access point detection," in *Proc. IEEE GLOBECOM*, 2007, pp. 355–360.

[13] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, "Rogue access point detection using temporal traffic characteristics," in *Proc. GLOBECOM*, 2004, pp. 2271–2275.

[14] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Enhancing the security of corporate Wi-Fi networks using DAIR," in *Proc. MobiSys*, 2006, pp. 1–14.

[15] S. Shetty, M. Song, and L. Ma, "Rogue access point detection by analyzing network traffic characteristics," in *Proc. IEEE MILCOM*, 2007, pp. 1–7.

[16] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting vehicular mobility in urban multi-hop wireless networks," in *Proc. MobiSys*, 2008, pp. 54–66.

[17] Air defence. [Online]. Available: http://www.airdefense.net

[18] S. Jana and S. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," in *Proc. MOBICOM*, 2008, pp. 104–115.

[19] Y. Sheng, K. Tan, G. Chen, and D. Kotz, "Detecting 802.11 MAC layer spoofing using received signal strength," in *Proc. IEEE INFOCOM*, 2008, pp. 2441–2449.

[20] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. MOBICOM*, 2008, pp. 116–127.

[21] Z. Ling, X. Fu, W. Jia, W. Yu, D. Xuan, and J. Luo, "Novel packet size-based covert channel attacks against anonymizer," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2411–2426, Dec. 2013.

[22] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "Softmac—Flexible wireless research platform," in *Proc. HotNets-IV*, 2005, pp. 1–5.

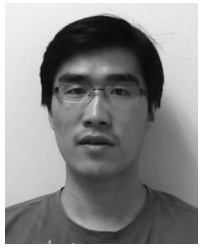
[23] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw. MobiCom*, 2010, pp. 173–184.

[24] U. Paul, R. Crepaldi, J. Lee, S.-J. Lee, and R. H. Etkin, "Characterizing WIFI link performance in open outdoor networks," in *Proc. SECON*, 2011, pp. 251–259.

[25] H. Han, F. Xu, C. C. Tan, Y. Zhang, and Q. Li, "Defending against vehicular rogue APs," in *Proc. IEEE INFOCOM*, 2011, pp. 1665–1673.

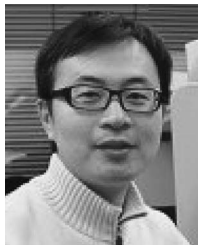
[26] Deliberant CPE 2-12. [Online]. Available: http://www.streakwave.com/mmSWAVE1/Video/CPE%202-12.pdf

[27] Openwrt. [Online]. Available: https://openwrt.org/



Hao Han received the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA, in 2013.

He is currently a Research Scientist with the Networks and Security Group, Intelligent Automation, Inc., Rockville, MD, USA. His research interests include wireless networks, mobile computing, cloud computing, and radio-frequency identification systems.



Fengyuan Xu (M'13) received the Ph.D. degree from the College of William and Mary, Williamsburg, VA, USA, in 2013.

He is currently a Researcher with the Storage System Group, NEC Laboratories America, Princeton, NJ, USA. His main research interests include mobile and distributed systems, including smartphone energy efficiency, secure wireless communications, and data-intensive computing. His current research interests include large-scale data analytic platforms and secure cloud computing.

Chiu C. Tan (M'13) received the Ph.D. degree in computer science from the College of William and Mary, Williamsburg, VA, USA.

He is currently an Assistant Professor in the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. His research interests include wireless security (802.11, vehicular, radio-frequency identification), cloud computing security, and security for mobile health systems.

Yifan Zhang received the B.S. degree in computer science from Beihang University, Beijing, China, in 2004. He is currently working toward the Ph.D. degree in computer science with the College of William and Mary, Williamsburg, VA, USA.

His research interests include wireless networks, mobile computing systems, and operating systems.

Qun Li (SM'12) received the Ph.D. degree in computer science from Dartmouth College, Hanover, NH, USA.

He is currently an Associate Professor with the Department of Computer Science, College of William and Mary, Williamsburg, VA, USA. His research interests include wireless networks and embedded systems, including pervasive computing, cognitive radio, wireless local area networks, mobile ad hoc networks, sensor networks, and radio-frequency identification systems.

Dr. Li received the U.S. National Science Foundation CAREER Award in 2008.