Online Detection of Fake Access Points using Received Signal Strengths

Taebeom Kim, Haemin Park, Hyunchul Jung, and Heejo Lee
Div. of Computer and Communication Engineering
Korea University
Seoul, Korea
Email: {ktb88, gaiger, change, heejo}@korea.ac.kr

***Abstract*—Wireless access points (APs) are widely used for the convenience and productivity of smartphone users. The growing popularity of wireless local area networks (WLANs) increases the risk of wireless security attacks. A fake AP can be set in any public spaces in order to impersonate legitimate APs for monetization. Existing fake AP detection methods analyze wireless traffic by using extra devices, and the traffic is collected by servers. However, using these server-side methods is costly and only provide secure communication, in limited places of clients' devices. Recently, several fake AP detection methods have been designed in order to overcome the server-side problems in a client-side. However, there are two limitations to the client-side methods: cumbersome processes and limited resources. When the methods attempt to collect data, calculating interval time incurs time-consuming processes to detect fake characteristics in the client-side. Moreover, the operating systems in smartphones provide limited resources that can hardly be adopted in the client-side. In this paper, we propose a novel fake AP detection method to solve the aforementioned problems in the client-side. The method leverages received signal strengths (RSSs) and online detection algorithm. Our method collects RSSs from nearby APs and normalizes them for accurate measurement. We measure the similarity of normalized RSSs. If the similarity between normalized RSSs is less than the fixed threshold value, we determine that the RSSs are generated from a fake device. We can measure the optimal threshold value derived from the sequential hypothesis testing. In our experiment, when the fixed threshold value was 2, the *true positive* was over than 99% and the *false positive* was less than 0.1% in three observations.**

I. Introduction

IEEE 802.11, a set of standards for wireless local area networks, has rapidly increased in popularity among home and enterprise clients: this is due to the standards' convenience and productivity. Recently, the increase in the number of smartphones has contributed to creating the synergy effect on WLANs because a smartphone provides a variety of services through wireless APs which are widely available in public spaces such as hotels, airports, schools, etc. In the past, laptops could be used for wireless access in the same way as the smartphone now can, but clients had difficulty walking around with their laptops, and using the wireless services on laptops cannot create the synergy effect on WLANs in the same way as is possible with smartphones. Mobility is an important consideration in WLANs.

Unfortunately, wireless security threats against WLANs are evolving as rapidly as the underlying wireless technologies and include a variety of attacks [1]. When clients use services through an AP and a smartphone, security-critical data for the services, such as e-bank account, e-mail account, and credit card number can be exposed by an attacker. The fake AP attack model, a type of Man-in-the-Middle (MITM) attack [1] [2] [3] [4], is the most feasible attack model. The attacker can easily set up a fake device in order to disrupt the network service or steal the sensitive data from nearby benign clients. For example, the attacker falsifies the communication messages in order to induce the user to go to phishing web-site: the attacker then injects a malicious code into benign clients' smartphones [5]. Therefore, it is important to determine whether an AP is fake, when a benign client wants to connect the AP for wireless services.

Previous research about fake AP problems and how to combat them points out the importance of guaranteeing the security in WLANs. Existing fake AP detection methods have been twofold: *Server-side* and *Client-side*. The server-side detection methods utilize extra devices, also known as Wireless Intrusion Detection System (WIDS) nodes. The WIDS nodes listen to the wireless traffic and send the gathered traffic to their servers. The servers learn the wireless environment to detect fake APs using wireless traffic from WIDS nodes. The client-side detection methods detect fake APs on client devices, without extra devices. The client device sends test packets to nearby APs in order to measure characteristics such as time intervals.

However, the previous works contained several limitations. In the server-side methods, if a client moves to other places where there are no WIDS nodes, the methods cannot guarantee secure communication in WLANs. Although there are WIDS nodes for secure communication, the detection methods hardly detect the fake APs when the servers have not learned wireless environments yet. Therefore, the server-side detection method for fake APs cannot guarantee the secure communication for mobility. In the client-side methods, it can provide the mobility for secure communication but existing methods based on the client-side have a cumbersome process in detecting fake APs in practice. Furthermore, the features of existing methods can hardly be applied in to smartphone.

Considering the problems mentioned above, we propose a novel fake AP detection method in the client-side. To overcome the limitations of the client-side, our method is designed to be a lightweight solution. The key process of our method is to find highly correlated RSS sequences that

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

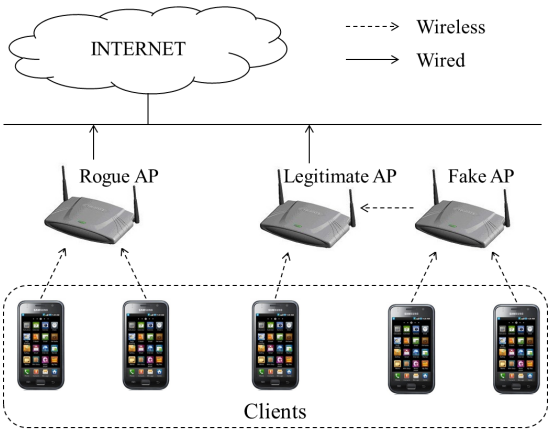

Fig. 1. There are two types of unauthorized APs in the network. The main difference between them is that the rogue AP is directly connected with a wired network, and the fake AP is indirectly connected to a wireless network.

can be collected in any wireless devices. If a similarity of RSS sequences is less than a threshold value, we define the RSS sequences as fake signals. In our experimental results, when the fixed threshold value was 2, the *true positive* was greater than 99% and the *false positive* was less than 0.1% after three observations. With the observations, our method detects the fake APs in an online manner. To the best of our knowledge, this work is the first to consider the practical fake AP detection method for a smartphone. Our main contributions are as follows:

- To guarantee the mobility of a client, we considered developing the fake AP detection method on a limited platform such as a smartphone.
- To guarantee availability to the client, our method discovers fake APs without extra monitoring devices or a network manager privilege in WLANs. Furthermore, the method does not require modification of the AP device, and it can detect the fake APs even if their traffic is encrypted.
- For the lightweight method, we provide fixed threshold values for detecting the fake APs using sequential hypothesis testing to enable us to detect malicious APs without learning tasks.

The rest of this paper is organized as follows: in Section II, we provide the general description of unauthorized AP and its surveys. Secion III defines the requirements and problems of our research. Section IV illustrates the proposed detection method of fake APs and describes sequential hypothesis testing for the threshold value. In Section V, we present the results of our experiments, and section VI concludes this paper.

II. Background and Related Work

Before we discuss existing works regarding detecting a fake AP, we clearly describe two different types of unauthorized APs in order to provide a better understanding of our work.

Fig. 1 shows an unauthorized AP type that is separated into two categories: 1) *rogue AP* and 2) *fake AP*.

1) The rogue AP is not only set up by an attacker but also by legitimate users, for convenience and productivity [6] [7] [8]. As it opens the inside network to unknown users, the rogue AP has minimal or absence of security.

2) The fake AP has some different characteristics from the rogue AP [9] [10]. The fake AP is only set up by a malicious user for malicious behaviors, such as eavesdropping, falsification, and others. If legitimate clients communicate with a fake AP, the attacker can eavesdrop and falsify entire messages.

In this paper, we focus on the fake AP scenario because there are some reasons that the rogue AP scenario is unsuitable in the real-world [9]. The first reason is of the limited number of ethernet ports. Also, when the rogue AP is successfully connected to the ethernet network, it cannot move closer to clients for inducement. Finally, network managers can disable unknown devices connected to the ethernet network.

We introduce several related works, broken down into two categories: *Server-Side* and *Client-Side*.

In the server-side methods, deployed extra devices collect the wireless signals and send them to the server for analysis. Jana and Kasera [10] have proposed fake AP detection using clock skews. Using extra devices, they collect beacon frames for calculating the clock skews of APs in the WLANs. After, if a clock skew is different from predefined legitimate clock skews, the AP is classified as a fake AP. Kao *et al.* [11] have proposed an unauthorized AP detection system based on sensing the existence of AP packets. If the MAC address of the AP packets is not on a white list, the system classifies it into be a fake or neighbor AP.

The client-side methods measure interval times between clients and APs. Han *et al.* [9] have proposed fake AP detection using round trip time (RTT) on a client device. They measure the RTTs, between a probe request and response. After that, they calculate differences between RTTs with a threshold to determine whether it is a fake AP. Recently, they have proposed another detection method of the vehicular fake AP, using received signal strength [12]. The vehicular fake AP detection method is similar to our work in terms of guaranteeing clients' mobility.

III. Problem Definition

In this section, we describe fake AP attack models. Then, we describe three requirements for detecting fake APs. The last subsection describes the problem statement that we aim to solve throughout this paper.

A. Fake AP attack models

The fake AP attack models aim to damage benign clients in WLANs. For the effective attack model, attackers try to generate multiple signals in their devices. For example, in Aircrack-suite, there is a module called *Airbase-ng*, which creates multiple SSIDs. If the attacker creates multiple SSIDs identical to legitimate ones, clients are unsure whether the AP is legitimate.

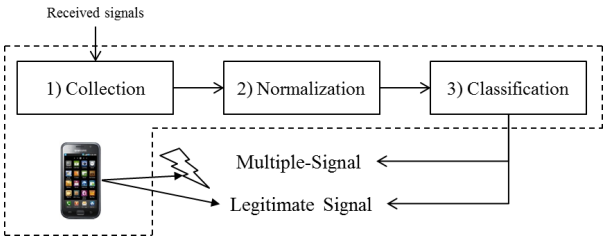

Fig. 2. The figure represents the overview of our detection method. The detection method collects received signal strengths and normalizes received signals for accurate measurement. Finally, the last step classifies whether the received signals are generated from a fake AP.

B. Requirements

Here, we describe three requirements for detecting fake APs and guaranteeing client-side.

1) **Portability**: the detection method should be able to run on multiple operating systems in client devices.
2) **Scalability**: the detection method should be applied to APs without modification.
3) **Safety**: the detection method should determine the fake AP before clients connect to it.

C. Problem statements

The problems we resolve to detect with regard to fake APs are twofold:

- To detect the fake AP without assistance from a network manager and without extra devices.
- To detect the fake AP without a learning phase for guaranteeing public places.

We design fake AP detection which satisfies the three requirements and resolves the problem statements above.

IV. Fake AP Detection using received signal strengths

In this section, we describe the fake AP detection method and sequential hypothesis testing in order to measure the optimal threshold value.

A. Fake AP detection method

Fig. 2 shows our detection method, which consists of three phases. 1) Collection of RSSs: the first phase measures the RSSs from nearby APs. 2) Normalization of collected RSSs: for accurate measurement, the second phase estimates some missed RSSs, caused by air conditions, such as interference, home appliance, noise or, etc., and normalizes the estimated RSSs for generalization of a variety of wireless environments. 3) Classification of RSSs: Finally, our method determines which RSSs are highly correlated to others based on our empirical threshold value Δ. We define that the highly correlated RSS sequences as fake signals from a single device.

1) Collection of received signal strengths: In the first phase, our method collects the RSSs from nearby APs. In IEEE 802.11 infrastructure, the received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal using beacons from nearby APs. There are two techniques to read beacons in WLANs. The first is an active scanning that sends a probe request message from a client to nearby APs. The AP's response is a probe response message to the client. The second technique is passive scanning, which involves listening for beacons from nearby APs. The APs typically send $10 \sim 100$ times per second [10]. A sequence of received signal strengths is represented as

$$s_i = [s_i^1, \cdots, s_i^t]^T,$$

where s_i^t is the received signal strength from ith AP at time t. Then, we represent k sequences as

$$S = [s_1, s_2, ..., s_k]^T.$$

2) Normalization of signal strengths: After the collection of received signal strengths, we estimate and normalize the vectors S for suitability to detecting fake APs. When we collect signals, some low signals or intermittent appearing signals can be collected from nearby APs due to some characteristics of WLANs, such as distance of APs, reflection, etc. The phenomenon of wrongly collected signals causes the received signal strength to become zero; this is termed missing data. Missing data can be a consequence of a failure result. There are some techniques to cope with the missing data problem [13].

For all the pairs of sequences s_i^t and s_j^t define b_t as

$$b_t = \begin{cases} 0, & \text{if both } s_i^t \text{ and } s_j^t \text{ are available} \\ 1, & \text{otherwise} \end{cases}$$

Then, the proximity $\wp$ between s_i and s_j is defined as

$$\wp(s_i^t, s_j^t) = \frac{t}{t - \sum_{k=1}^{t} b_k} \sum_{\forall t: b_t = 0} \phi(s_i, s_j), \sum_{k=1}^{t} b_k \neq t$$

,where $\phi(s_i, s_j)$ denotes the proximity between the two sequences s_i and s_j. When a dissimilarity measure is involved, the ϕ is the *Manhattan Norm* and is represented as $\phi(s_i, s_j) = |s_i - s_j|$.

For the normalization process, the following equation transforms received signal strengths into a range of [0, 1]. If $d_1(s_i, s_j)$ is equal to $\min_{1 \leq i \leq t} d_1(s_i, s_j)$, then $Normal(S)$ is equal to 0. If $d_1(s_i, s_j)$ is equal to $\max_{1 \leq i \leq t} d_1(s_i, s_j)$, then $Normal(S)$ is equal to 1.

$$Normal(S) = \frac{\max_{1 \leq i \leq t} d_1(s_i, s_j) - d_1(s_i, s_j)}{\max_{1 \leq i \leq t} d_1(s_i, s_j) - \min_{1 \leq i \leq t} d_1(s_i, s_j)},$$

where $Normal$ metric is a 3-dimensional space that has a distance matrix of each AP's unit of time. If the distance between s_i and s_j is less than a threshold, we determine that the signals are strongly indicative of a fake AP.

3) Classification of RSSs: The last step classifies whether a RSS is multiple-signal or not. The classification process measures a distance of two randomly selected signal sequences. If the distance is above the Δ, we cluster the two signal sequences that are highly correlated with each other. The highly correlated signals are classified into multiple signals generated from a fake AP. We discuss the threshold value Δ in the next subsection, and the algorithm 1 is the classification process.

Algorithm 1: Classification of received signal strength

```
Normal ← A metric from the normalization phase;
N_ap ← The number of APs; t ← The observed times;
Δ ← The threshold value; C ← φ;
for i ← 1 to N_ap − 1 do
    C_i ← {i};
    for j ← i + 1 to N_ap do
        sum ← 0;
        for k ← 1 to t do
            sum ← sum + |Normal_i^k − Normal_j^k|;
        end
        if sum ≤ Δ × t then
            C_i ← C_i ⋃ {j};
        end
    end
end
for i ← 1 to N_ap do
    if |C_i| ≥ 2 then
        C_i ← A Multiple-Signal AP;
    end
    else
        C_i ← Legitimate APs;
    end
end
```

The algorithm 1 describes how to cluster the highly related signals that we call multiple signals. There are four input parameter, $Normal$ is a 3-dimensional vector which has sequences of received signal strengths from nearby APs, N_{ap} denotes the number of nearby APs, t represents the number of iterations for receiving beacons and Δ is a threshold value for determining whether the two sequences are related. The output C is a set of related RSS sequences. For example, if $C_1 = \{1, 3\}$ then the RSS sequences of s_1 and s_3 are the multiple signals from a single device.

B. Sequential hypothesis testing

The multiple signals from a single device have similar power sequences. Based on this observation, we determine how similar sequences are defined as multiple signals and what is the optimal threshold value Δ should be in order to differentiate between single and multiple signals. We also estimate the optimal number of observations N_{ob} to determine whether the AP is fake. To solve this problem, we use the sequential hypothesis testing theory developed by Wald in his work [14].

Given two randomly selected RSS sequences, S_1 and S_2, let Y_t be a random variable that represents whether the difference of the sequences is equal to or less than Δ, where

$$Y_t = \begin{cases} 0, & \text{if } \sum_{i=1}^{t} |S_1^i - S_2^i| \leq \Delta \times t \\ 1, & \text{if } \sum_{i=1}^{t} |S_1^i - S_2^i| > \Delta \times t \end{cases}$$

We define two hypotheses H_0 and H_1 as follows: H_0 is the null hypothesis that the randomly selected two RSS sequences are generated from a single device, and H_1 is the alternative hypothesis that the two randomly selected RSS sequences are generated from multiple devices. Then, we express the distribution of Y_i as below:

$$Pr[Y_i = 0|H_0] = \theta_0, Pr[Y_i = 1|H_0] = 1 - \theta_0$$

$$Pr[Y_i = 0|H_1] = \theta_1, Pr[Y_i = 1|H_1] = 1 - \theta_1$$

We now have four decisions from two hypotheses. If the algorithm selects H_1 when the H_1 is true, the decision is called a *true positive*. On the other hand, if the algorithm selects H_0 when the H_1 is true, this is called a *false positive*. Likewise, if the algorithm selects H_1 when the H_0 is true, it is called a *false negative*. Finally, choosing H_0 when it is true is termed a *true negative*. We use the two detection probabilities of P_D, true positive, and P_F, false positive, for specifying detection performance. The desired values for α and β are as follows:

$$P_F \leq \alpha \qquad and \qquad P_D \geq \beta \tag{1}$$

where typical values might be α = 0.01 and β = 0.99.

Following [14], as each RSSs is observed, we calculate the likelihood ratio of determining the hypothesis to be true while satisfying the performance condition (1) as follows:

$$\Lambda(Y) \equiv \frac{Pr[Y|H_1]}{Pr[Y|H_0]} = \prod_{i=1}^{n} \frac{Pr[Y_i|H_1]}{Pr[Y_i|H_0]}$$

where Y is the vector of events (i.e., the difference of two randomly selected sequences is not more than the threshold value Δ). We can compute the expected number of observations to detect fake APs online [15]:

$$E[N_{ob}|H_1] = \frac{\beta \ln \frac{\beta}{\alpha} + (1-\beta) \ln \frac{1-\beta}{1-\alpha}}{\theta_1 \ln \frac{\theta_1}{\theta_0} + (1-\theta_1) \ln \frac{1-\theta_1}{1-\theta_0}} \tag{2}$$

In experiment section, we determine the optimal threshold value from experimental results. With the optimal threshold value, we can also estimate the expected number of observations in order to detect fake APs from Eq. (2).

V. Experiment and Results

In this section, we describe the setup of our experiments and evaluate the threshold value Δ, then estimate the expected number of observation N_{ob}.

A. Experiment setup

Our experiment is divided into three components: legitimate APs, a fake AP, and a client. We use a laptop which performs as a fake AP. We add a wireless card, ipTIME G054UA 802.11b/g/n using RT chipset, on the laptop for monitoring mode. The aircrack-suite, a widely used for auditing wireless network tool, is used to perform the fake AP on the laptop. We use a smartphone, Samsung Galaxy Tab based on android 2.2, as a client.

We evaluate our method in public spaces such as coffee shops, stations, and universities which are already equipped with legitimate APs. We set up a fake AP near the legitimate APs and measured received signal strengths from both legitimate and fake AP on the smartphone. The following section describes the experimental results.

B. The threshold value Δ

Table I shows the optimal threshold value of detecting fake APs. There are three probabilities, which are accuracy, true positive rate, and false positive rate. When the Δ is 1, the performance is best. However, the small threshold Δ determines multiple fake signals as legitimate signals. When the Δ is more than 2, the true positive rate is close to 1. However, accuracy and false positive rate are decreased. As a result, when the Δ is 2, the detection performance is optimal and has high accuracy, true positive rate, and low false positive rate.

TABLE I
DETECTION PROBABILITIES BY THE Δ

	The Threshold Value Δ				
	1	2	3	4	5
Accuracy	0.971	0.965	0.91	0.877	0.834
True Positive Rate	0.301	1	1	1	1
False Positive Rate	0.009	0.036	0.093	0.126	0.17

C. The expected number of observations N_{ob}

After determining the Δ, we can estimate the expected number of observations N_{ob} from the Δ. Fig. 3 shows how $E[N_{ob}|H_1]$ is changed by the detection probabilities, true positive and false positive. When Δ is 2, the *true positive rate* is over 95% and the *false positive* is less than 0.1%. Fig. 3 shows that the performance satisfies Eq. (1) in three observations.

VI. CONCLUSION

In this paper, we present a fake AP detection method to protect against the stealing of sensitive data in a client-side. Our fake AP detection method measures correlated RSS sequences from nearby APs in order to determine whether the sequences are legitimate or fake. Using the sequential hypothesis testing theory, we predefine the appropriate threshold value Δ using the expected number of iterations. The predefined threshold value enables us to detect fake APs without supervised learning techniques. In practice, we developed the fake AP detection method in a client-side, such as an android platform. Our experimental results show that the detection method is accurate and lightweight when detecting fake APs in a client-side.

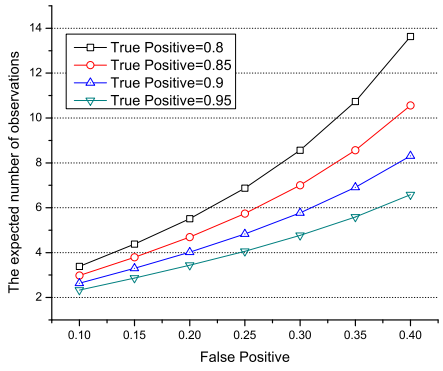

Fig. 3. The figure shows the expected number of iterations by the *true positive* and the *false positive*. If the *true positive* is 0.95 and the *false positive* is 0.1, the expected number of observations is 3.

ACKNOWLEDGEMENTS

This research was supported by the Seoul R&BD Program (WR080951).

REFERENCES

[1] Airdefense. [Online]. Available: http://airdefense.net
[2] Airmagnet. [Online]. Available: http://www.airmagnet.com
[3] Airwave. [Online]. Available: http://airwave.com
[4] T. R. Schmoyer, Y. X. Lim, and H. L. Owen, "Wireless intrusion detection and response: A case study using the classic man-in-the-middle attack," in *Proceedings of IEEE Wireless Communication and Networking Conference*, 2004.
[5] Immunity, the future of wireless security assessment, http://www.immunitysec.com.
[6] A. Godber and P. Dasgupta, "Countering Rogues in Wireless Networks," in *ICPP Workshops*, 2003, pp. 425–431.
[7] S. Srilasak, K. Wongthavarawat, and A. Phonphoem, "Integrated Wireless Rogue Access Point Detection and Counterattack System."
[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 *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 365–378.
[9] H. Han, B. Sheng, C. W. Tan, Q. Li, and S. Lu, "A Measurement Based Rogue access point Detection Scheme," in *INFOCOM*, 2009, pp. 1593–1601.
[10] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08. New York, NY, USA: ACM, 2008, pp. 104–115.
[11] K.-F. Kao, T.-H. Yeo, W.-S. Yong, and H.-H. Chen, "A location-aware rogue AP detection system based on wireless packet sniffing of sensor APs," in *ACM SAC*, 2011, pp. 32–36.
[12] H. Han, F. Xu, C. C. Tan, Y. Zhang, and Q. Li, "Defending against vehicular rogue APs," in *INFOCOM*, 2011, pp. 1665–1673.
[13] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.
[14] A. Wald, *Sequential Analysis*. J. Wiley & Sons, 1947.
[15] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing," in *IEEE Symposium on Security and Privacy*, 2004, pp. 211–225.