Journal of Quantum Computing
DOI:10.32604/jqc.2020.09433
Article

Tech Science Press

A Position Self-Adaptive Method to Detect Fake Access Points

Ping Lu[1,2,*]

[1]Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2]Guizhou University, Sate Key Laboratory of Public Big Data, Guiyang, 550025, China

[*]Corresponding Author: Ping Lu. Email: pinglu@bupt.edu.cn

Received: 02 May 2020; Accepted: 07 September 2020

Abstract: In recent years, with the maturity and popularity of Wi-Fi technology, wireless hotspots have been deployed on a large scale in public places. But at the same time, it brings many security issues that cannot be ignored. Among them, the fake access point attack is a very serious threat in wireless local area network. In this paper, we propose a method to detect fake access points in wireless local area network. First, our detection method is passive, which means there is almost no additional traffic will be generated during the program's operation. Second, different from many existing methods, our method allows the detection device to change position, the move will be perceived and the fingerprint will be updated automatically. Third, we use a variety of features as fingerprints to describe an access point better and improve efficiency. At last, the method we propose is more in line with the actual scene and has been proved effective by experiments.

Keywords: Fake AP; WLAN; beacon frame

1 Introduction

With the popularity of Wi-Fi and the establishment of more and more infrastructure, Wi-Fi can be seen everywhere in people's lives. At public places such as fast-food restaurants, café, shopping malls Wi-Fi service is provided [1]. At the same time, for ease of use, they usually do not set password for these access points (APs), which is very easy to be used by malicious attacker. Users can browse the web and use instant messaging tools after being connected to the Internet, and some users will perform financial operations such as online payment. Once the user is connected, they may be exposed to various attacks. The attacker can use it to obtain the account name and password of the mobile phone user when they browse the shopping site, or they can intercept the traffic for further analysis. This will lead to the disclosure of personal privacy information, such as leaking the account and password of the shopping website.

Fake AP uses a software-based AP which is installed in a portable device [2]. The access point has the same Service Set Identifier (SSID) with the legitimate AP. If the attacker wants to simulate it more realistically, the fake AP can has the same Media Access Control (MAC)Address, channel information with the legitimate AP and so on with the legitimate AP. Wi-Fi is a wireless local area network (WLAN)based on the IEEE 802.11 standard. According to the 802.11 protocol, in a wireless local area network, a user connects to WLAN need to provide SSID and password. For the convenience of the user, a password is required only when connecting to the AP for the first time. In the following connection, user's equipment automatically selects the network according to the connection history of the WLAN and the signal strength. When a malicious attacker collects enough information of a legitimate AP, he establishes a fake AP with the same SSID and MAC address as the legitimate AP, furthermore, the fake AP does not ask user for password and shows stronger signal than the legitimate AP [3]. It is very likely that the user's terminal will be connected to a fake AP automatically without the user's permission.

After the user connects to a fake AP, all traffic generated by the user will be forwarded by the fake AP. The username and password entered during the user's online operation, and the images and web pages user has scanned will be intercepted by malicious attackers. Unencrypted data can be obtained directly, furthermore, the attackers can tamper with the information and then send it out [4]. As can be seen from the above, the fake AP can eavesdrop on user privacy and even modify the user communication data, which is extremely harmful. Fig. 1 shows the WLAN topology in two cases.

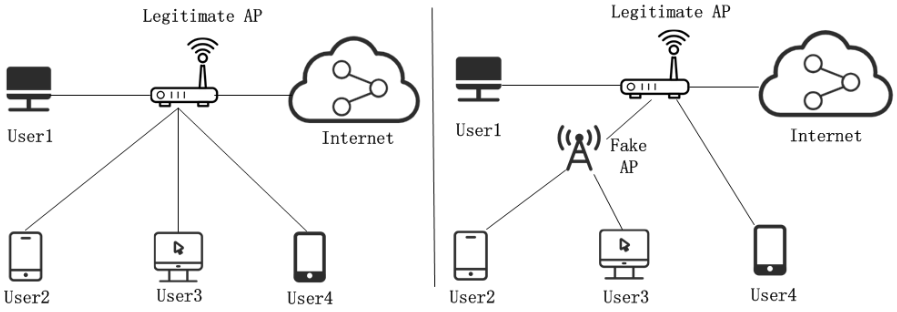

Figure 1: WLAN without fake AP and WLAN with fake AP

In this paper, we propose an adaptive fake AP detection method based on fingerprint fusion. The innovation of the method is that the fingerprint of legitimate AP can be dynamically updated. When the detection device moves, the detection device can perceive its position change, update fingerprint in the legitimate AP fingerprint list automatically. The rest of the paper is organized as follows. In the second section, we introduce the related work. In the third section, we present the framework of our algorithm, introduce the relevant knowledge and describe each part of the framework in detail. In the fourth section we verify and evaluate our method through experiments, and the fifth section summarizes our work.

2 Related Work

In order to detect fake AP, there have been many researches. We can divide the detection methods into two types, server-based and client-based.

2.1 Server-Based

The server-based detection methods usually need a detection device with a detection system installed. The device can monitor frames in the WLAN, and extract important fingerprint information, then save the legitimate AP's information in a white list. By comparing current information with that in the white list, we can detect the fake AP.

Apisak and Sakchai propose a mechanism for fake AP detection considering the sequence number of a beacon frame [5]. The three consecutive beacon frames received from the same AP are used as a detection window. The leftmost frame sequence number is *Fsn,* and the rightmost frame sequence number is *Lsn*. Since one AP broadcasts the beacon frame every 100ms by default, and the sequence number in the beacon frame is continuous. So calculate whether *Lsn-Fsn* is greater than 1. If it is greater than 1, it indicates that a fake AP found.

Bandar and Khaled propose a method for fake AP detection using an empirical fingerprint [6]. They believe that only the tool developers can increase the size of the beacon frame. All of the attackers cannot increase the size of the beacon frame. Therefore, the MAC, SSID, and beacon frame size of the AP can be used as fingerprint, and a legitimate threshold is determined for the frame size. Extract received beacon frame information then compare beacon frame size and other fingerprints to decide whether fake AP exists.

Yong et al. observe that Received Signal Strength (RSS) follows a mixture of multiple Gaussian distributions, so a method based on Gaussian mixture model is proposed [7]. They establish an RSS configuration file for fake AP detection. Because RSS is a measurement that is difficult to forge and highly correlated with the position of the transmitter, assuming a reasonable distance between the attacker and the victim, RSS can be used to distinguish them to detect MAC spoofing.

Suman et al. consider clock skew as a device fingerprint [8], they calculate the clock skew of the AP through the Time Synchronization Function (TSF) timestamp sent in the beacon/probe response frame. They find that the clock skew of the same AP remained the same, but the difference between the APs is large.

2.2 Client-Based

The client-based detection system is very suitable for installation on the user's device. Generally, the white list of the legitimate AP devices does not need to be generated in advance. Different from the passive detection method of the server, the client-based fake AP detection method usually sends out special detection packets. The client-based detection usually utilizes various time metrics (e.g., round trip time or inter-packet arrival time) to find fake AP [9].

Chao and Yimin et al. propose a method that does not rely on any training information or knowledge [3]. Generally, a fake AP still needs a normal AP to access the Internet. The fake AP transfers traffic in the middle. Therefore, the number of hops that the user accesses the Internet increases. As long as the attacker complies with the TCP protocol and the 802.11 standard, the increased delays introduced by one additional wireless hop can't be neglected. In this paper, the packet arrival time is used as the detection information to distinguish the one-hop and two-hop wireless channels on the client side.

Qian et al. use the special length effective data frames (SL-EDF) as an indicator to determine the malicious forwarding behavior [10]. They propose a fake AP detection method based on the arrival time of a special frame length. They monitor the frames emitted by target APs and filter out the SL-EDF shared by the two suspect APs, then extract their arrival time generated on the detection node. If fake AP exists, their arrival time should be approximately equal.

3 Proposed Method

Our proposed fake AP detection method can be divided into four separate stages, as shown in Fig. 2, which shows the four stages of processing. In this chapter we will describe our detection method in detail.

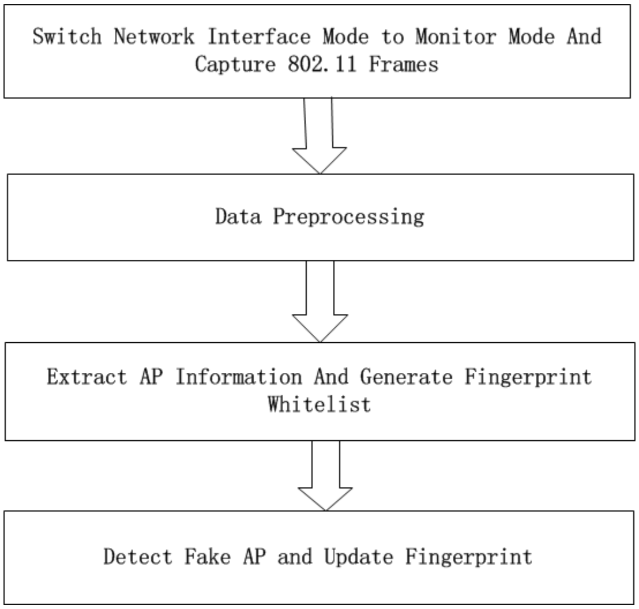
Switch Network Interface Mode to Monitor Mode And Capture 802.11 Frames
Data Preprocessing
Extract AP Information And Generate Fingerprint Whitelist
Detect Fake AP and Update Fingerprint

Figure 2: Fake AP detection method framework

3.1 Capture 802.11 Frames

In the default mode of the network card, we cannot capture the 802.11 frames, so we choose Linux operating system and switch the network card mode to monitor to capture frames. We use Scapy to capture packets. Scapy is a very powerful third-party library in Python that can be used for network sniffing, packet sniffing and packet forging.

3.2 Data Preprocessing

For the different functions of the frame, the frames in 802.11 can be divided into the following three categories.

(1) Control frame: Used for handshake communication during the competition and positive confirmation, ending non-competition period, etc.

(2) Management frame: Used for negotiation and relationship control between STA and AP, such as association, authentication, synchronization, etc.

(3) Data frame: Used to transfer data during competitive and non-competitive periods.

The beacon frame belongs to management frame, which is mainly used to announce the existence of a certain network. The beacon frame that the AP periodically broadcasts every 100 ms can let the mobile station know the existence of the network, thereby adjusting the parameters necessary for joining the network. Fig. 3 shows the structure of beacon frame.

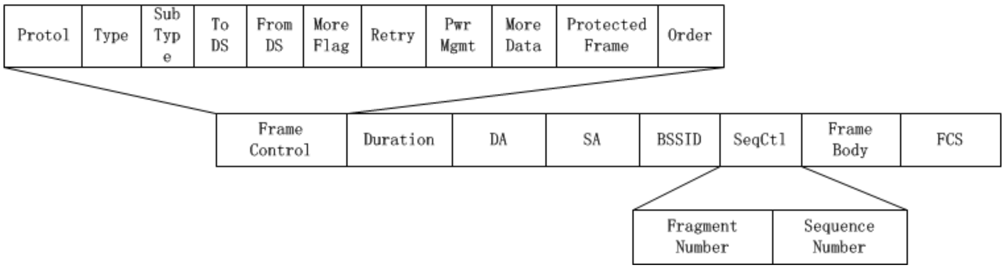

Figure 3: Beacon frame structure

The detection uses beacon frames, so we need filter out the beacon frames. First, the *type* field is used to classify frames, 00 means it is a management frame, and then the *subtype* field is used to distinguish the subtype of the frame, for beacon frame it is 1000. So we can filter out the beacon frames by these two conditions.

3.3 Fingerprint Extraction

Research-related fields in beacon frame:

- SSID: Service Set Identity, a string used to distinguish different networks, which can be considered as the name of the AP we saw before the Wi-Fi connection.
- SA: Source Address, can be regarded as the MAC address of the AP.
- SeqCtl: Sequence Control, which is used to reorder frame segments and filter repeat frames, can be divided into two parts: 12-bit sequence number and 4-bit fragment number [11].

The value of the sequence control cannot be modified by software. In order to announce the existence of the AP and let the user connect to him, the AP periodically sends beacon frame every 100ms, but there may be a delay when receiving. The sequence control number of the beacon sent by each AP is continuous with respect to itself.

The SSID can be considered as the name of wireless network found around our cell phone [12], and the SSID field can be directly extracted. We consider SA as the MAC address of the AP. The sequence number can be obtained from the SeqCtl field.

The RSS can be obtained from the radio tap. RSS is related to the distance [13]. The closer the distance is to the AP, the larger the RSS is. RSS value is negative. The closer the detection device is to the AP, the closer the value is to 0. At the same time, uncontrollable factors such as temperature and humidity will also affect RSS.

In a secure environment, for each legitimate AP, we collect the SSID and MAC of it. Ideally, the sequence number in beacon frame the same AP broadcasts is continuous, but in the actual scene, there may be too many APs or the signal strength is not strong, so we cannot capture every packet. The number of the sequence number may not be continuous, but the difference between the front and the back will not be large. Record the sequence numbers $sn_1, sn_2, sn_3...$ in the beacon frames the legitimate AP broadcasts, and calculate the difference $diff_1, diff_2, diff_3...diff_n$ between the adjacent two sequence numbers, make $diff$*2 as the sequence number difference threshold (*SN Threshold*) where $diff$ is the largest value among $diff_i - diff_{i-1}$.

In our proposed method, the fingerprint extraction stage is under a secure environment, the location of the detection device and the detected AP is fixed. After the fingerprint extraction stage finished, the detection device can move. When the detection device move, the fingerprint parameters related to the location will change, such as RSS, the threshold of the sequence number, so these two fingerprints need to be updated. In order to describe the relative position of the detection device, three anchor APs need to be established. After the anchor APs established, set the anchor APs to hide the SSID, so the SSID of the anchor APs will not be displayed in the scanned Wi-Fi list. The attackers are usually not interested in forging AP with hidden SSID. But we can still capture the beacon frames the anchor APs broadcast. Beacon frame has RSS field, and RSS is related to distance. According to this feature, we can use the anchor APs to help determine whether the position of the detection device has changed.

The detection device extracts RSS of the anchor APs and stored with a vector (RSS_a, RSS_b, RSS_c). We measure the vector multiple times and store these vectors in a matrix, then we use the Euclidean distance to calculate the pairwise distances between vectors in matrix, we make the largest value as the *RSS Threshold*.

$$\mathrm{d}(x,y) = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}(x_i - y_i)^2} \tag{1}$$

Finally, the information about one legitimate AP can be represented by a vector *(SSID, MAC, SN Threshold)*. We use a *buffer* to store the legitimate APs information, $buffer_i$ is a vector that represents necessary information of a legitimate $\mathrm{AP_i}$. At the same time, a vector $(RSS_A, RSS_B, RSS_C, RSS\ Threshold)$ is generated to represent the position information of the detection device itself, RSS_i is the average signal strength of the three anchor APs:

$$RSS_A = {\sum_{\mathrm{i}=1}^{\mathrm{n}} x_{ai}}/{\mathrm{n}} \tag{2}$$

3.4 Fake AP Detection and Fingerprint Update

We obtained a fingerprint list of the legitimate APs in secure environment, which contains sufficient information describing every legitimate AP. Then we start the fake AP detection and fingerprint update stage. With our method, the fake AP detection device can adaptively update the fingerprint according to the change of position, that is, after the data preprocessing stage, the detection device can move, and the legitimate AP fingerprint list can be updated in time. Fig. 4 shows the flow of our proposed fake AP detection method.

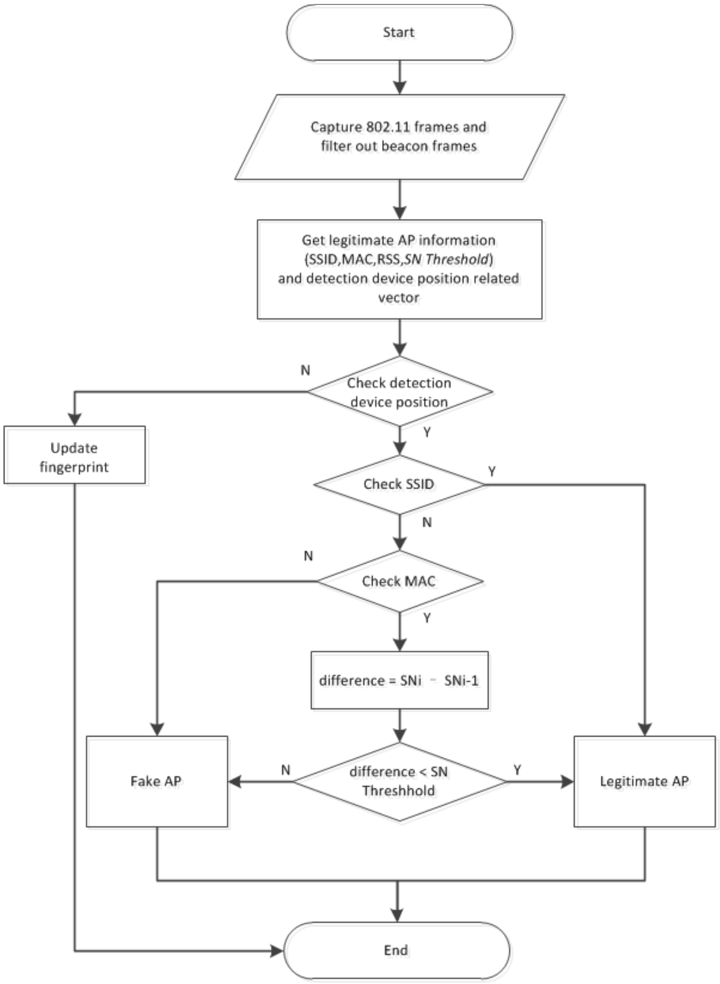
Start
Capture 802.11 frames and filter out beacon frames
Get legitimate AP information (SSID,MAC,RSS,SN Threshold) and detection device position related vector
N
Check detection device position
Y
Update fingerprint
Check SSID
Y
N
N
Check MAC
Y
difference = SNi - SNi-1
Fake AP
N
difference < SN Threshhold
Y
Legitimate AP
End

Figure 4: Algorithm of proposed method

Since the change of the position of the detection device affects the detection result, the fake AP detection device should first determines whether the position has changed. Get the *(RSS_A, RSS_B, RSS_C)* of the anchor APs, calculate the distance between this vector and the anchor APs average RSS vector. If the distance is greater than the *RSS Threshold*, the detection device moved. Then record the difference between two consecutive vectors until the difference tends to be stable. At this time, record the position information of the detection device including anchor AP's average RSS and RSS Threshold, and update the legitimate AP fingerprint list.

After the first three stages, we will filter out all beacon frames, and extract the fingerprint information. The fingerprints can be divided into two categories. One type is a fixed fingerprint feature, if it does not meet the conditions, it must be a fake AP, such as SSID, MAC. Another type of fingerprint is dynamic, it needs to be determined in a variety of situations, such as RSS threshold, sequence number threshold. For a static feature, if the SSID is not in the legitimate APs list, it must not be a fake AP. If the SSID is in the legitimate APs fingerprint list, continue to fetch the MAC and compare it with the MAC address that in the list, if the MAC is different, fake AP found. If the addresses are the same, compare sequence number. We get the difference between the sequence numbers of two adjacent beacons, if the difference is less than the sequence number threshold, it is legitimate, if not, fake AP found.

4 Experiment and Analysis

In our experimental scene, there is a legitimate AP, two laptop devices. One was installed with Kali operating system and necessary programs for emulating the fake AP, the other was installed with Ubuntu operating system and the detection program. We set network card on monitor mode to detect fake AP. There are also three anchor APs for providing location-related information, the anchor APs are set to turn off SSID broadcast. In addition, there are multiple wireless users, including mobile phone users and laptop users, as well as other unknown APs. Fig. 5 is our experimental model.

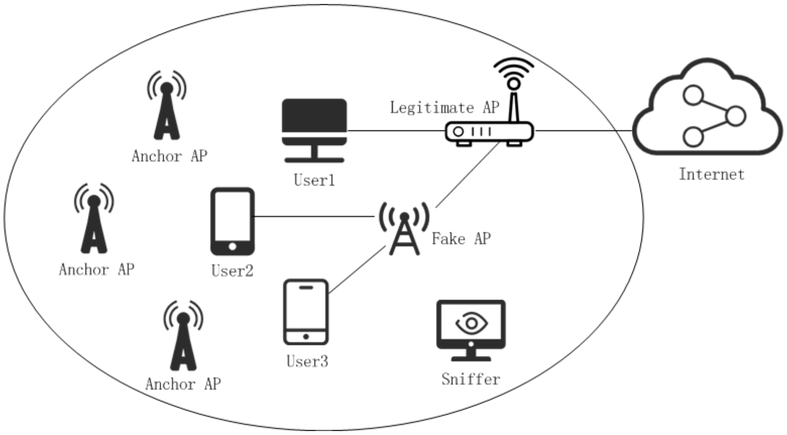

Figure 5: Experimental model

In the data preprocessing stage, there is no fake AP in WLAN. The sequence number of the legitimate AP is continuous, and since the location of the anchor APs are fixed, the signal strength of the anchor APs received by the detecting device is stable. Fig. 6 shows the signal strength of the three anchor APs received by the detection device over a period of time. According to the method we proposed in Section 3, we can calculate a vector (-30.8125, -54, -45) that can describe the relative position of the detection device, and we can also calculate the *RSS Threshold* is 4.123.

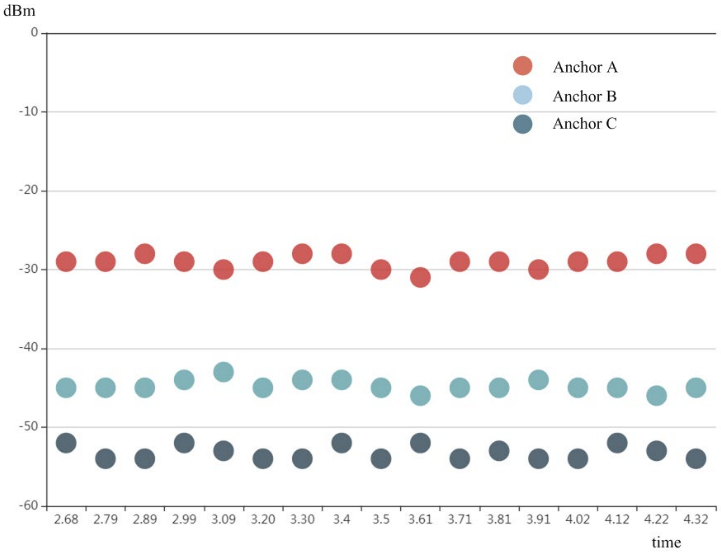

Figure 6: RSS of anchor APs

When a fake AP is simulated by kali, the sequence number in beacon frame legitimate AP broadcasts is no longer grows linearly, the difference between adjacent beacon frames sequence number is large and irregular.

The experiment can effectively detect the fake AP when the detection device is fixed. When there is no fake AP in the WLAN, if we move the detection device, the detection program can perceive that the position of the detection device has changed and will update the fingerprint automatically instead of alarming.

Compared with some client-based fake AP detection methods, our detection method is passive. Except for the beacon frames anchor APs broadcast, no additional traffic will be generated in the WLAN when the system is working. In terms of detection efficiency, when the detection device is fixed in position, our method has high detection efficiency. When the detection device moves, the system can perceive the change and update the fingerprint. Then the system starts the fingerprint extraction stage and during this period it cannot detect the fake AP. However, in the actual scene, the detection device generally does not move frequently, and the probability of occurrence of a fake AP during the detection device move is small, so our detection method still has high availability. The experiment proves that our proposed detection method can achieve 84.7% accuracy.

5 Conclusion

In this paper, we propose a position self-adaptive method to detect fake AP in WLAN. By combining multiple fingerprints, it is possible to update the fingerprints automatically when the position of the detection device has been changed. We have confirmed our method through experiments and it can detect fake AP accurately. In the future, we plan to improve the detection accuracy, to locate the position of the fake AP and try to introduce more fingerprints.

Acknowledgement: This work is supported by the NSFC (Grant Nos. 61671087, 61962009, 61003287), the Fok Ying Tong Education Foundation (Grant No. 131067), the Major Scientific and Technological Special Project of Guizhou Province (Grant No. 20183001), the Foundation of State Key Laboratory of Public Big Data (Grant No. 2018BDKFJJ018), the High-quality and Cutting-edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China), the Fundamental Research Funds for the Central Universities, and the Fundamental Research Funds for the Central Universities (No. 2019XD-A02).

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Modi and C. Parekh, “Detection & analysis of Evil Twin attack in wireless network,” *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 1, 2017.

[2] B. Alotaibi and K. Elleithy, “A passive fingerprint technique to detect fake access points,” *Wireless Telecommunications Symposium, IEEE*, pp. 1–8, 2015.

[3] C. Yang, Y. Song and G. Gu, “Active user-side evil twin access point detection using statistical techniques,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1638–1651, 2012.

[4] M. Samra, M. Mengi and S. Sharma, “Detection and mitigation of rogue access point,” *Journal of Scientific and Technical Advancements*, vol. 1, no. 3, pp. 195–198, 2015.

[5] K. Apisak and T. Sakchai, “Rouge access point detection mechanism considering sequence number of beacon frame for wireless local area networks,” in *2017 14th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 509–512, 2017.

[6] B. Alotaibi and K. Elleithy, “An empirical fingerprint framework to detect rogue access points,” *2015 Long Island Systems, Applications and Technology*, pp. 1–7, 2015.

[7] Y. Sheng, K. Tan and G. Chen, “Detecting 802.11 MAC layer spoofing using received signal strength,” in *IEEE INFOCOM 2008-The 27th Conf. on Computer Communications*, pp. 1768–1776, 2008.

[8] S. Jana and S. K. Kasera, “On fast and accurate detection of unauthorized wireless access points using clock skews,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 449–462, 2009.

[9] F. H. Hsu, Y. L. Hsu and C. S. Wang, “A solution to detect the existence of a malicious rogue AP,” *Computer Communications*, vol. 1, no. 142, pp. 62–68, 2019.

[10] Q. Lu, H. Qu and Y. Ouyang, “SLFAT: Client-side Evil Twin detection approach based on arrival time of special length frames,” *Security and Communication Networks*, vol. 2019, no. 1, pp. 1–10, 2019.

[11] B. Hu, P. Yi, “Rouge access point detection based on beacon sequence,” *Electronic Measurement Technology*, vol. 40, no. 4, pp. 123–126, 2017.

[12] X. Y. Li and X. Y. Li, “Rogue access points detection based on theory of semi-supervised learning,” in *Int. Conf. on Security, Privacy and Anonymity in Computation, Communication and Storage*, vol. 10658, no. 1, pp. 35–44, 2017.

[13] X. Feng and Y. Hu, “Research of rouge AP detection based on CSI in smart home,” *Information Security Research*, vol. 4, no. 2, pp. 163–169, 2018.