Received: 11 September 2018 | Revised: 16 March 2020 | Accepted: 18 April 2020

DOI: 10.1002/dac.4460

RESEARCH ARTICLE

WILEY

A passive user-side solution for evil twin access point detection at public hotspots

Fu-Hau Hsu[1] | **Chuan-Sheng Wang**[1,2] | **Chih-Wen Ou**[1] | **Yu-Liang Hsu**[1]

[1]Department of Computer Sciences and Information Engineering, National Central University, Taoyuan, Taiwan

[2]Information and Communication Security Lab, Chunghwa Telecom Co., Ltd, Taipei, Taiwan

Correspondence
Chuan-Sheng Wang, Department of Computer Sciences and Information Engineering, National Central University, Taoyuan, Taiwan.
Email: luckytft@gmail.com

Funding information
Ministry of Science and Technology, Taiwan, Grant/Award Number: MOST 105-2221-E-008-074-MY3

Summary

This paper proposes a passive user-side solution, called Wi-Fi legal access point (AP) finder (LAF), to the notorious evil twin access point problem, which in turn can result in diverse security problems, such as fraud, identity theft, and man-in-the-middle attacks. Due to the severe security threats created by evil twins, many promising solutions have been proposed. However, the majority of these solutions are designed for the administrators of wireless networks, not for Wi-Fi users. Hence, they are either too expensive or need some data that are usually not accessible to normal users. LAF utilizes the TCP three-way handshake-related packets and packet forwarding property created by evil twins to find legal APs, called good twins, at public hotspots or unencrypted WLANs; thus, it does not need any data or assistance from wireless network administrators. LAF does not send exploring packets actively; hence, evil twins cannot sense its existence. No matter when and where a user needs to utilize an AP to connect to the Internet at a hotspot, he can just use LAF to find out a legal AP to connect to. Experimental results show that LAF can quickly and accurately find legal APs after observing only a few packets.

KEYWORDS

evil twin, rogue access point, Wi-Fi, wireless security

1 | INTRODUCTION

The development of laptops and various mobile devices, such as smartphones, PDAs, and tablets, allows a user to carry the above devices around the world easily; meanwhile, the rich content and services provided by the Internet dramatically increase users' requirement to access the Internet at any place at any time. As a result, IEEE 802.11 wireless networks (WLAN) that allow a user to access the Internet become a popular facility for people's everyday life quickly. Through the access points (APs) of a hotspot, people can conveniently connect to the Internet in public spaces, such as airports, schools, bus/railroad stations, hotels, cafes, restaurants, and so on. Even though APs used at homes or companies usually utilize preshared keys and WEP/WPA/WPA2 to protect related WLANs, most hotspots only use web pages (captive portals) to authenticate users and no wireless encryption method is used when the wireless packets are transmitted between them and their users; hence, according to the US Department of Homeland Security,[1] "most Wi-Fi hotspots do not encrypt the information you send over the Internet." Since no encryption method is used to encrypt the wireless packets of most hotspots, the TCP/IP headers of these wireless packets can be directly observed at these hotspots. According to iPass,[2] there are about 103 million public hotspots in early 2016. The large user pool of public

hotspots makes them attractive targets of attackers. Various attacks aiming at WLANs and their users have been developed by attackers. Among these attacks, evil twins[3-6] are one of the most notorious attacks.

An evil twin is a kind of rogue APs. However, unlike traditional rogue APs which are usually installed by insiders for convenience sake and are directly connected to an internal wired network, an evil twin is usually installed by outsiders for vicious purpose and is connected to the Internet through an existing nearby legal AP, called good twin. Figure 1 shows a traditional setup of an evil twin. Through some off-the-shelf software,[7,8] an attacker can transform a laptop into an evil twin AP. Then, the attacker sets the service set identifier (SSID) of the evil twin as the same SSID of the legal AP (good twin) that he plans to fake. The evil twin can utilize high received signal strength indication (RSSI) to attract normal Wi-Fi users to connect to it, because according to 802.11 standard, no matter how many APs exist, a Wi-Fi user always chooses the AP with the strongest RSSI to associate with. Besides, the attacker can launch deauthentication attacks[9] to force normal Wi-Fi users to disassociate with the good twin AP and connect to the evil twin AP. Finally, the evil twin forwards Wi-Fi users data to the Internet through the good twin.

When user data are sent to the Internet through an evil twin, the evil twin can sniff the data to steal sensitive information, such as passwords, web sessions, credit card information, and so on. The evil twin also can launch man-in-the-middle attacks and phishing attacks. What is worse is that according to Yang et al.[10] an evil twin attack is easy to launch and an evil twin attack is also easy to be successful. Due to the severe threats that evil twins can impose upon WLAN users, many promising solutions have been proposed. However, the majority of these solutions were designed for system administrators, not for a normal Wi-Fi user who may need to use wireless networks to connect to the Internet at any time at any place. They may need (1) special devices, such as wireless sensors, (2) special data, such as lists of legal APs/IPs or training data, (3) network traffic trace, (4) dedicated network managers, and so on. The above requirements usually are not available to normal Wi-Fi users. Many AP owners even do not have enough budgets to support a dedicated network manager 24/7.

To improve the above problems, active user-side solutions were proposed. However, to examine an AP, these solutions must connect to the Internet first. Nowadays, many applications on computers or smartphones continue probing whether their devices connect to the Internet. Once they find they can connect to the Internet, they will automatically log in remote servers. Hence, before confirming whether an AP is an evil twin, some sensitive information, such as passwords of related servers, may have already been eavesdropped. Besides, the exploring packets sent by active user-side solutions usually have specific patterns which can be detected by evil twins. An evil twin can ignore the exploring packet and let the good twin to handle them. After these solutions finish their analyses and connect to the good twin, the evil twin can launch deauthentication attacks to force a Wi-Fi user to connect to it. Furthermore, existing active user-side solutions usually make their detection based on some time metrics. However, these time metrics may be influenced by prefetching, network topologies, traffic volume, or network types. To solve the above problems, this paper proposes a passive client-side solution, called Wi-Fi legal AP finder (LAF), to find legal APs/good twins and evil twins at hotspots or unencrypted WLANs. LAF makes its detection based on the TCP three-way handshake-related packets and the packet forwarding behavior of evil twins. Hence, LAF only needs to know the TCP/IP headers of wireless packets. During LAF's observation phase, if the three-way handshake of a TCP connection is proceeding, LAF can accurately detect whether an AP is a legal one through only few IP packets. Because various TCP connections are created and deleted frequently in a network, LAF can finish its detection quickly.

LAF has the following properties which make it an ideal tool for Wi-Fi users.

1) As a user-side solution, LAF does not require any information, lists of legal APs/IPs, training data, or assistance from WLAN administrators. LAF can protect a Wi-Fi user any time at any place independently.

FIGURE 1 Traditional setup of an evil twin

2) As a solution based on the well-designed TCP/IP protocol, it is difficult for an evil twin to evade LAF' s detection, because every TCP connection must go through the three-way handshake steps and the IP addresses, sequence numbers, and acknowledgement numbers of the IP packets of an ongoing TCP connection cannot be modified arbitrarily.
3) As a passive solution, an evil twin is not able to find that it is monitored by LAF, let alone adopt any counter-detection action immediately
4) As a passive solution, LAF does not need to associate with an AP to make its detection. In other words, LAF does not send any exploring packets; hence, while making detection, auto-login software in the LAF host cannot automatically send sensitive data.
5) As a solution that does not depend on any time metric, the detection accuracy of LAF is not influenced by network topologies, traffic volume, network types, and prefetching mechanism.

The rest of the paper is organized as follows. Section 2 discusses rogue AP and evil twin-related work. Section 3 describes the principle, algorithm, and proof of algorithm of LAF. Section 4 discusses various experimental results to evaluate the effectiveness and efficiency of LAF and some security issues about LAF. Section 5 gives the conclusion.

2 | RELATED WORK

Due to the severe threat created by rogue APs, many solutions have been proposed by industrial and academic researchers. We can classify these solutions into three categories: radio frequency sniffing, gateway-side detection, and active client-side detection.

2.1 | Radio frequency sniffing

This category of solutions utilizes various devices, such as sensor APs,[11,12] sensors,[13,14] or mobile devices with a wireless interface,[15,16] to scan the spectrum between 2.4 and 5 GHz to collect various data from detected APs. Then, diverse data about detected APs, such as SSID, MAC addresses, RSS values,[17] clock skews,[18] and radio frequency variations,[19] are extracted from the collected data as fingerprints of related APs. Finally, the fingerprints are compared with an authorized list to filter out rogue APs. However, the time and labor required to operate additional equipment, and additional equipment increases the maintenance cost of a WLAN. Besides, without continuously monitoring a WLAN, the above solutions may not be able to provide complete protection to a WLAN or its users; after all, an attacker can install an evil twin at any time and remove an evil twin quickly. Furthermore, these solutions may misjudge a legal neighbor AP as a rogue AP. To overcome part of the above problems, some hybrid solutions[20-22] were proposed. For example, Bahl et al.[20] turn existing desktops into wireless sniffers to decrease deployment cost and increase efficiency instead of sniffing wireless traffic, and Beyah et al.[21] and Yin et al.[22] also observe wired traffic to avoid misjudging a legal neighbor AP as a rogue AP. Hence, after sending packets to the Internet through a suspect AP, if an internal sensor observes the same packets in the wired traffic, then the systems can be sure that the suspect AP is not an evil twin.

2.2 | Gateway-side detection

Wired communication and wireless communication use different protocols to transmit packets between hosts. However, different protocols result in different packet transmission patterns and different packet transmission times. Based on the above properties, several studies[21-33] utilize various time metrics which are calculated from packets passing through a gateway to infer whether the related client host comes from a wireless network. If the host is not in the authorized list, then the AP transmitting the client packets is deemed as a rogue AP. Based on the intuition that inter-packet arrival times (IATs) of wireless traffic are more random than those of wired traffic, Beyah et al.[21] were among the first authors to use the above time metric to passively detect rogue APs. Using the median of the same time metric, Shetty et al.[28] proposed an automated classifier to detect rogue APs. Wei et al.[31,34] have shown that the inter-arrival time of a TCP ACK-pair is a good statistic to detect wireless hosts. Based on the above finding, Wei et al.[32] utilized the inter-arrival times of TCP ACK-pairs, combined them with the properties of 802.11 protocol and wireless channels, and

then used sequential hypothesis to differentiate wired and wireless traffic, which in turn was used to detected rogue APs. Watkins et al.[30] observed that the round trip time (RTT) for the traffic traversing the wired link is obviously less than that of the wireless links because of the instability and lower link capacity of wireless channels compared with wired links. Hence, they adopted RTT as the time metric to find wireless traffic and unauthorized APs. Mano et al.[27] used local RTT to differentiate wired traffic and wireless traffic. To increase the accuracy of the detections, their solution, RIPPS, utilizes packet slicing to eliminate large size packets. Venkataraman and Beyah[29] utilized the jumps in the IAT to filter out wired traffic and detect rogue APs. They found that the jump signature was created by the CSMA/CA of the DCF and the rate adaptation mechanism in the 802.11 MAC. Ma et al.[26] proposed a hybrid framework to detect rogue APs. Their work combined both wireless surveillance and gate-side traffic analyzer. The time metric utilized by them is the interpacket spacing. Unlike most of the above work which utilized various time metrics to distinguish wired traffic and wireless traffic, Kao et al.[35] utilized client-side bottleneck bandwidth to detect rogue APs, because they detected that client-side bottleneck bandwidth is related to the client connection device bandwidth and is able to differentiate wired traffic and wireless traffic. Most of this category of solutions needs to analyze the packets passing through the gateway of an organization. These solutions are mainly provided for system administrators, not for travelers. After all, the packet trace of an organization is usually unavailable to a traveler due to security concern. In fact, it is very dangerous to provide the network trace of an organization to an unknown third party because the trace may contain private, confidential, or important information. Besides, most of the solutions need an authorized list of APs/IPs which is usually not available to a traveler too. Furthermore, most of the above solutions detect rogue APs by distinguishing wired traffic and wireless traffic first. However, an evil twin usually hides behind a legal AP; hence, current versions of most of these solutions may not be used directly to detect an evil twin. Finally, most of this class of solutions distinguishes wired traffic and wireless traffic through various time metrics collected from IP packets passing through a gateway. However, these time metrics may be influenced by various factors such as traffic volume, network topology, network type, and network speed, which in turn may influence the detection accuracy.

2.3 | Active client-side detection

Active client-centric evil twin AP detection[10,25,26,36-40] is an emerging promising solution category for evil twin AP detection. By connecting to an AP, actively sending exploring packets, measuring various time metrics, and using different algorithms, this type of solution examines whether the exploring packets are transmitted through one or two APs to detect evil twins. This kind of solutions does not need any authorized AP/IP list, assistance from WLAN operators, or network IP trace that passes through a gateway; hence, they are very suitable for travelers who need to connect to an AP quickly at any place that provided Wi-Fi access at any time. However, this kind of solutions also faces some common challenges which need to be overcome to make them widely adopted. First, exploring packets usually have special forms which can be easily detected by evil twins. Thus, they can use various steps to bypass being detected. Second, active detection usually utilizes various time metrics related to inspecting packets, such as RTT or IAT, to discriminate evil twins from legitimate APs. However, when network traffic volume, topology, or device capability change, the same time metrics may not represent that the related WLAN uses the same number of wireless channels to transmit exploring packets. Third, it is possible for an experienced attacker to send forgery legitimate packets to bypass these solutions. For example, a previous solution[37] exploited the characteristics of TCP sequence and acknowledgment numbers to detect evil twins. If attackers could modify the TCP sequence or acknowledgment numbers when they raise an evil twin attack, it is possible to bypass the detection.

Nicholson et al.[41] proposed Virgil to automatically discover and select APs. Virgil associates to each AP found during a scan and chooses suitable APs based on the bandwidth estimation and RTT to a set of reference servers. Han et al.[25] proposed a client-centric evil twin detection approach which utilizes the RTT between a user and a DNS server to independently decide whether an AP is an evil twin without any assistance from WLAN operators. When using their approach to detect rogue APs, related AP(s) will receive successive DNS queries from the same client to nearby DNS servers. The above phenomenon may be used by a rogue AP to examine whether someone is inspecting it. The threshold utilized by their system to distinguish legitimate APs from rogue APs was decided by the WLAN used in the authors' research. However, different WLANs have different properties, such as network topology, transmission time, traffic volume, and so on. Even the packet transmission time of the same WLAN may change, when the traffic volume changes. Therefore, for a traveler who needs to visit different places at different time, this solution may not always function as expected. Yang et al.[10] also proposed a user-side evil twin AP detection system, called ETSniffer, which uses the

IAT as the detection statistic to detect evil twins. ETSniffer provides two algorithms, TMM and HDT, to distinguish one-hop wireless channels from two-hop wireless channels for different environments. ETSniffer does not need any support from WLAN operators or any authorized AP/host list. However, in order to get accurate IAT data, ETSniffer needs to send IP packets with special form based on immediate-ACK policy. Hence, an evil twin can detect the inspecting packets based on the form and adopt various approaches, such as prefetching web content, to bypass detection. Lu et al.[38] proposed a client-side solution which detects evil twins based on SYN reflection mechanism, called BiRe. Unlike LAF which only needs to send one SYN packet to complete its detection, BiRe needs to install two wireless network interface cards (WNICs) in a computer. Then, the computer makes two different TCP/IP connections separately through these two distinct WNICs to make its detection. Besides, both Hsu et al.[36] and BiRe proposed their own approaches to detect a new type of wireless security attacks. Under this type of attacks, a malicious AP does not forward victims' packets to an original legal AP in a wireless network, but sends victims' packets to the Internet through a 3/4/5 G network directly.

3 | PRINCIPLE AND DETECTION ALGORITHM

This section describes the monitor mode of a WNIC, fundamental phenomenon of an evil twin attack, design principle, and detection algorithm of LAF.

3.1 | Monitor mode

The monitor mode of a WNIC allows it to monitor all nearby wireless traffic. Unlike promiscuous mode, the monitor mode enables a WNIC to capture wireless packets without the need to associate with an AP. Most of the WNICs and modern OSes support monitor mode. But each has its own way to switch to this mode. In Windows, monitor mode is supported after Windows Vista and is controlled by Microsoft Windows Network Driver Interface Specification (NDIS) API. Microsoft provides a network monitor program called Microsoft Network Monitor[42] to allow users to operate WNICs in a convenient way. Activating monitor mode in Unix OS family is simpler. There are some built-in network-related commands for this purpose.

Because an evil twin needs to use a good twin to connect to the Internet, an evil twin has two wireless adaptors. One adaptor is imitated as a legal AP to induce users. This adaptor has an SSID and a WNIC will identify it as an AP. The other one is used to connect to the good twin; thus, it does not have an SSID and behaves like a laptop. Even though both an evil twin and its corresponding good twin have the same SSID, they have different basic SSIDs (BSSIDs), that is, MAC addresses. As a result, as shown in Figure 2, when an evil twin is forwarding packets, from the information contained in the wireless packets that a WNIC collects, an evil twin will be deemed as an AP and a laptop. In the following subsections, we will use an AP and a laptop to represent an evil twin.

3.2 | Packet forwarding

For a TCP connection between a victim and a server, the evil twin needs to forward corresponding packets between the victim and the good twin, because the evil twin uses the good twin to connect to the server. Therefore, packet forwarding is a fundamental phenomenon of an evil twin attack. LAF was designed based on this unchangeable

FIGURE 2 Wireless network interface card's view of an evil twin. AP, access point

property to make it difficult for evil twins to evade LAF's detection. Moreover, TCP three-way handshake is an essential procedure to establish a TCP connection. Three-way handshake also provides a good authentication mechanism between two machines. Even though when an evil twin forwards wireless packets, it can change many fields in the TCP and IP headers, an evil twin cannot change packets used in three-way handshake, such as SYN packets, SYN/ACK packets, and ACK packets, because changing these packets will influence the establishment of TCP connections. However, by simply adding or deducting some offsets, the evil twin even can change the sequence numbers and acknowledgment numbers of wireless packets, because it can restore these numbers later when receiving related response packets without influencing TCP connections. Furthermore, several encryption methods have been developed to protect the content of wireless packets from being eavesdropped. Thus, the TCP/IP headers of wireless packets might be encrypted and LAF may need to decrypt wireless packets to get information from the TCP/IP headers. However, in public areas, Wi-Fi service providers usually either do not encrypt wireless packets or they will provide encryption keys to their users. With the encryption key, a user can decrypt the wireless packets and get their TCP/IP headers to examine whether an AP forwards three-way handshake-related packets to other AP. Therefore, before using an encrypted Wi-Fi, a user can still use LAF to find a legal AP.

3.3 | Design principle and LAF algorithm

LAF finds legal APs/good twins and evil twins based on the principle that a good twin does not forward wireless packets; however, on the contrary, an evil twin forwards three-way handshake-related packets to remote servers, and when doing so, the evil twin cannot change the IP addresses of the remote servers. Figure 3 shows the transmission orders and paths of the three-way handshake packets of a TCP connection. The connection is established between a user and a remote server through a legal AP. Figure 4 shows the transmission orders and paths of the three-way handshake packets of a TCP connection. The connection is established between User 1 and a remote server through an evil twin. User 3 could be any notebook that has installed LAF.

Based on the transmission orders and paths of the three-way handshake packets of a TCP connection, this paper proposes the following algorithm, called LAF algorithm, to find legal APs/good twins and evil twins.

Assume the time that an AP accepts a wireless packet from a WNIC and forwards the packet through another WNIC is $T_{forward}$. The RTT between a WLAN user and a remote server is T_{RTT}. Besides, for a wireless packet, we use an n-tuple (channel, flag, sequence number, acknowledgment number, IP_{src}, MAC_{src}, IP_{dst}, MAC_{dst}) to record various information about the packet. Channel represents the wireless channel used to transmit the packet. The flag field could be SYN, SYN/ACK, or ACK. IP_{src} means source IP address. IP_{dst} means destination IP address. MAC_{src} represents source

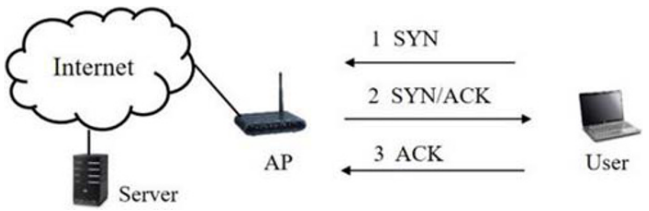

FIGURE 3 A user connects to the Internet through a legal AP. The number before each packet represents the transmission order of the packet. AP, access point

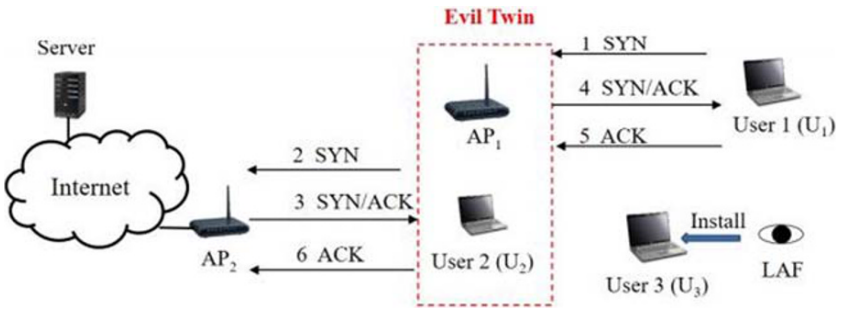

FIGURE 4 User 1 connects to the Internet through an evil twin. The number before each packet represents the transmission order of the packet. AP, access point; LAF, legal AP finder

TABLE 1 List of important symbols

Symbol	Description
AP_1	The evil twin
AP_2	The good twin
CH_1	The wireless channel which the evil twin use to transmit the packet
CH_2	The wireless channel which the good twin use to transmit the packet
IP_{src}	Source IP address
IP_{dst}	Destination IP address
IP_{server}	Remote server's IP address
MAC_{src}	Source MAC address
MAC_{dst}	Destination MAC address
S	The remote server
$T_{forward}$	The time that an AP accepts a wireless packet from a WNIC and forwards the packet through another WNIC
T_{RTT}	The round trip time between a WLAN user and a remote server
IP_{U_1}	The user

[a]Abbreviation: AP, access point; WNIC, wireless network interface card.

MAC address, and MAC_{dst} represents destination MAC address. In an n-tuple, if a field is recorded as *, it means the field will be ignored by LAF. Table 1 lists the important symbols and their meanings to be used in our analysis.

1) LAF checks whether there exist two APs with the same SSID. If there are no two APs with the same SSID, there is no evil twin. The detection finishes. Otherwise, The WLAN may contain an evil twin. Hence, go to Step 2 and assume that AP_1 with MAC address MAC_{AP_1} and AP_2 with MAC address MAC_{AP_2} have the same SSID. AP_1 uses channel CH_1 to access wireless packets, and AP_2 uses channel CH_2 to access wireless packets. However, both AP_1 and AP_2 could be on the same channel.
2) Monitor the WLAN until a SYN packet sent by a local host U_1 with IP address IP_{U_1} and MAC address MAC_{U_1} to a remote server with IP address IP_{server} is found. Assume that the SYN packet contains the following features (CH_i: i=1 or 2, SYN, SEQ_{U_1}, *, IP_{U_1}, MAC_{U_1}, IP_{server}, MAC_{AP_i}: i is the same i in the first field). Then, switch to the other channel, CH_j: j=1 or 2. If i=1, then j=2, else j=1.
3) Keep monitoring channel CH_j to detect an SYN packet with the following features (CH_j, SYN, SEQ_{U_2}, *, IP_{U_2}, MAC_{U_2}, IP_{server}, $MAC_{AP\ j}$: j is the same j in the first entry). If after $T_{forward}$, no such SYN packet appears, go to Step 4. Otherwise, go to Step 5.
4) Switch to channel CH_i and keep waiting for a packet with following features (CH_i, SYN/ACK, SEQ_{AP_i}, $SEQ_{U_1}+1$, IP_{server}, MAC_{AP_i}, IP_{U_1}, MAC_{U_1}). If after T_{RTT}, no such packet appears, go back to Step 2. Otherwise, AP_i is a good twin and this round of detection finishes.
5) Continue monitoring channel CH_j until a SYN/ACK packet with the following features (CH_j, SYN/ACK, SEQ_{server}, $SEQ_{U_2}+1$, IP_{server}, $MAC_{AP\ j}$, IP_{U_2}, MAC_{U_2}) is seen. If after T_{RTT}, no such packet appears, go back to Step 2. Otherwise, go to Step 6.
6) Switch to channel CH_i. Wait for another SYN/ACK packet with the following features (CH_i, SYN/ACK, SEQ_{AP_i}, $SEQ_{U_1}+1$, IP_{server}, MAC_{AP_i}, IP_{U_1}, MAC_{U_1}). If after T_{RTT}, no such SYN/ACK packet appears, go back to Step 2. Otherwise, go to Step7.
7) Continue monitoring CH_i and wait for an ACK packet with the following features (CH_i, ACK, $SEQ_{U_1}+1$, $SEQ_{AP_i}+1$, IP_{U_1}, MAC_{U_1}, IP_{server}, MAC_{AP_i}). If after T_{RTT}, no such ACK packet appears, go back to Step 2. Otherwise, switch to channel CH_j and go to Step 8.
8) Continue monitoring channel CH_j until an ACK packet with the following features (CH_j, ACK, $SEQ_{U_2}+1$, SEQ_{server}+1, IP_{U_2}, MAC_{U_2}, IP_{server}, $MAC_{AP\ j}$) is seen. If after $T_{forward}$, no such packet appears, go back to Step 2. Otherwise, APi is an evil twin, AP_j is a good twin, and this round of detection finishes.

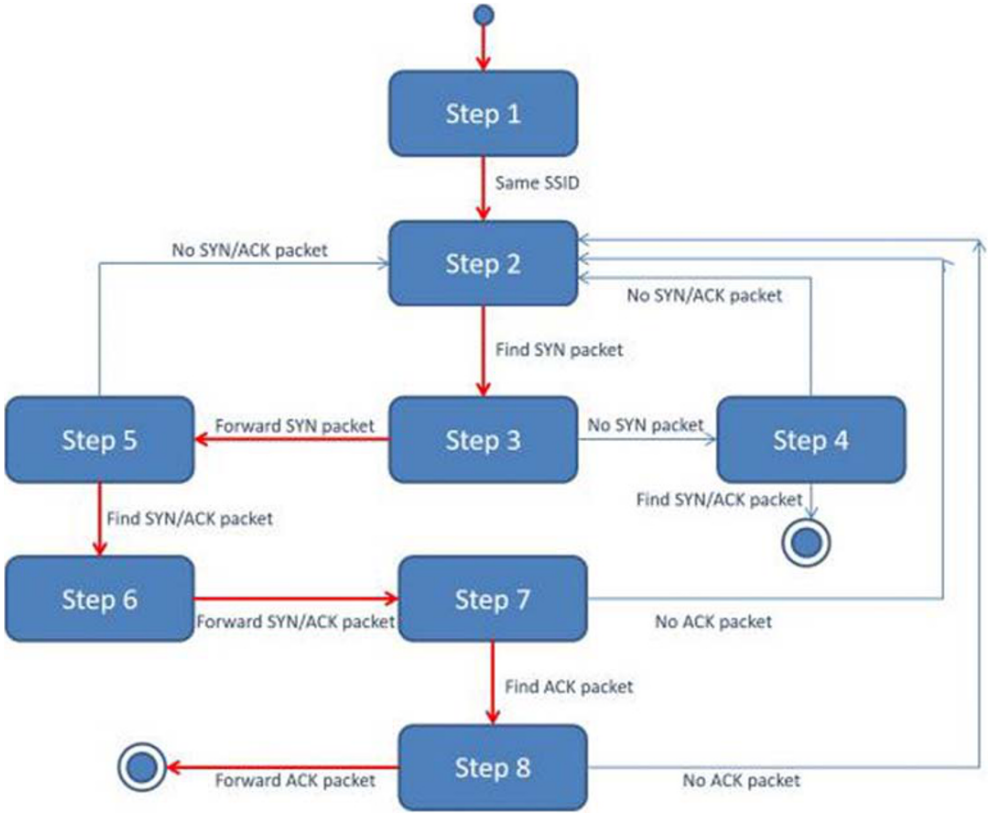

FIGURE 5 The detection round of legal access point finder

LAF makes R rounds of the above detection for an AP as Figure 5 shows. R is called *repeat number* hereafter. Among the R rounds of detection for an AP, if more than half of detection results deem the AP as a good twin, LAF records that AP as a good twin; otherwise, LAF records AP as an evil twin. In Section 4.3, we will discuss how we choose an appropriate R value so that LAF can accurately find a good twin and an evil twin.

3.4 | Correctness analysis of LAF algorithm

The three-way handshake algorithm of the TCP/IP protocols makes sure that each set of SYN, SYN/ACK, and ACK packets belongs to the same TCP connection. Hence, for each SYN packet, we can correctly find its corresponding SYN/ACK packet and ACK packet. To examine the correctness of LAF algorithm, two things must be confirmed. First, will an evil twin be classified as a good twin? Second, will a good twin be deemed as an evil twin?

For the first question, an evil twin will be classified as a good twin, if the forwarded SYN packet or the corresponding SYN/ACK packet of the forwarded SYN packet is lost. However, in this case, there will be no SYN/ACK packet of the SYN packet sent by an end user. Hence, Step 4 of LAF algorithm can avoid this misjudgment. However, for an end user U_1 who is connecting to a remote server S through an evil twin AP_i, there is one rare case that may influence the LAF algorithm, if the case is not handled appropriately. What follows is the scenario of the rare case. Before the second wireless adaptor of the evil twin AP_i forwards user U_1's SYN packet, a different user U_3 may utilize the good twin to connect to the same remote server S too by sending the good twin a SYN packet. In other words, LAF will see user U_3's SYN packet before it sees the forwarded SYN packet of user U_1. As a result, LAF will go to Step 6 earlier. If after going to Step 6, LAF only waits for $T_{forward}$, it may lost U_1's ACK packet; hence, it will not be able to make any decision and will go back to the very beginning to analyze the network traffic again. Hence, in Step 6, we enlarge the waiting time to T_{RTT} to solve this problem. However, after an evil twin receives an SYN packet, if a delay happens at the evil twin and the evil twin forwards an SYN packet after $T_{forward}$, and the SYN/ACK packet of the SYN packet returns to the client host in T_{RTT}, a false negative happens. If an evil twin owner does not make the above delay on purpose, the above situation rarely happens and can be solved using the methods described in Section 4.3. But if an evil twin owner makes the delay on purpose, a false negative occurs. In Section 4.5, we will discuss this problem and a solution.

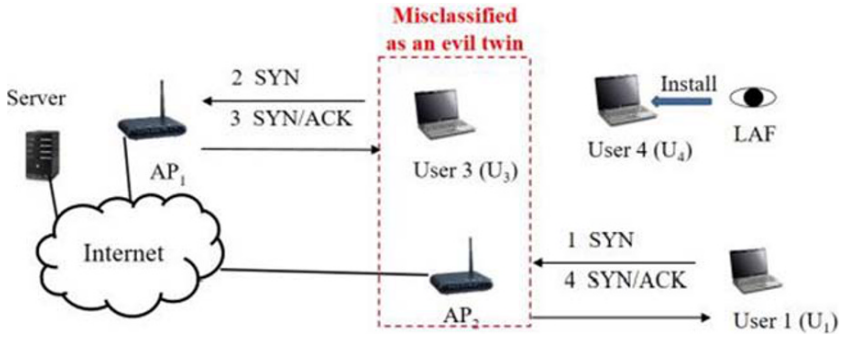

FIGURE 6 Analysis of the possibility that LAF misclassifies a good twin as an evil twin. The number before each packet represents the transmission order of the packet. User 4 could be any notebook that has installed LAF. AP, access point; LAF, legal AP finder

For the second question, the problem is whether it is possible that a good twin be deemed as an evil twin. We use Figure 6 to assist our analysis. Assume a wireless user, U_1, connects to a remote server S through a good twin, AP_2. At Step 2, LAF detects the SYN packet, SYN_{U_1}, sent by U_1. Then, it proceeds to Step 3. When LAF is at Step 3, a normal end user, U_3, begins to establish a TCP/IP connection with the same remote server S by sending a *SYN* packet, SYN_{U_3}, to the remote server. Hence, this second SYN packet becomes a forwarded SYN suspect of LAF. LAF keeps monitoring the same channel and will see the related SYN/ACK packet later on. Hence, LAF will come to Step 6. But at Step 6, LAF is not supposed to see the SYN/ACK packet of the first SYN packet because when LAF is at one channel checking the SYN/ACK packet of the second SYN packet, the SYN/ACK packet of the first SYN packet has already been transmitted on the other channel. After all, both SYN_{U_1} and SYN_{U_3} are sent to the same remote sever through the same good twin, AP_2, and SYN_{U_1} are sent before SYN_{U_3}; hence, the SYN/ACK packet of SYN_{U_1} is supposed to come back to the WLAN before the SYN/ACK packet of SYN_{U_3} in most cases. An exception happens when SYN_{U_1} and SYN_{U_3} are delivered through different paths and SYN_{U_3} arrives at the remote server earlier than SYN_{U_1}. Besides, if SYN_{U_1} and SYN_{U_3} are handled by different remote server threads and the thread handling SYN_{U_3} sends related ACK packet earlier than the thread handling SYN_{U_1}. But the above two situations rarely happen and can be solved by using the methods described in Section 4.3. Therefore, the above analysis shows that LAF will not misjudge a good twin as an evil twin.

4 | EVALUATION

In this section, we utilize various experiments to evaluate the effectiveness and efficiency of LAF. Besides, we also discuss the weaknesses of LAF and possible solutions to these weaknesses. We set up our experiments at a university campus with more than 10 000 students. We used a laptop equipped with two 802.11 network adapters to create an evil twin. The evil twin had the same SSID with its good twin which was a normal campus AP. The evil twin laptop had a 2.4-GHz Intel Core 2 Duo CPU and 4-GB memory. One of the laptop wireless cards was configured as the AP part of the evil twin. The evil twin is connected to the Internet through the good twin. The evil twin was deployed right beside a user laptop; hence, it produced almost 100% RSSI. The network sharing option of the evil twin laptop was switched on in order to redirect packets. To capture and analyze network traffic, LAF uses Microsoft Network Monitor 3.4 which is mentioned in Section 3.1. LAF was installed on another laptop runnning Microsoft Windows 7 64-bit operating system with a wireless network card, a 2.4-GHz Intel Core 2 Duo CPU, and 4-GB memory.

4.1 | TCP/IP connection establishment pattern

To evaluate the effectiveness of LAF, we used two dedicated hosts to simulate normal users' client hosts and used these hosts to create TCP/IP connections to remote servers. In order to make the TCP/IP connection establishment pattern of a client host similar to the pattern created by a normal user, we first used a sniffer to observe the traffic created by a host when a user surfs Gmail, Facebook, Google, Twitter, and other popular services. Then, we used the observed pattern to make TCP/IP connections to the following three websites: www.google.com, tw.yahoo.com, and adl.tw. The good twin and adl.tw are within the same subnet. The results of monitoring Facebook, Gmail, and Twitter show that there is at least one TCP/IP connection established in 30 s and at least five connections established in 1 min when surfing these websites. In a later subsection, we use the measured TCP/IP connection rate as a starting point for our experiments.

4.2 | Discussion of T_{RTT} and $T_{forward}$

T_{RTT} and $T_{forward}$ are two important factors of LAF. In this subsection, we discuss these values that we used in our experiments. The analysis result over 22 million TCP/IP connections made by Jay et al.[43] shows that the *RTT* ranges between 1 ms and 200 s. The majority of the *RTT* are smaller than 1 s, and only 5% of *RTT* are between 1 and 10 s. Phillipa et al.[44] show that the maximum *RTT* is about 629.0 ms. The average *RTT* is 145.7 ms. Our experiments were performed in a wireless network, and the *RTT* value in a wireless network should be higher than the *RTT* value in a wired network because the wireless network protocols are more complicated than the wired network protocols and a wireless network usually is not as stable as a wired network. According to Rafael et al.[45] the increment of the *RTT* of a wireless network is less than $20\mu s$. Xian et al.[46] shows that the *RTT* of 90% connections to Google is less than 100 ms. Given the above results, we chose 600 ms as the value of T_{RTT}.

$T_{forward}$ may be different in different practical environments with various infrastructures and hardware equipments; however, we discovered that $T_{forward}$ is usually less than 10 ms under various hardware environments. As a result, we chose 10 ms as the value of $T_{forward}$. It is possible that an attacker extends the forwarding time to conceal his forwarding behavior. However, LAF could still detect this behavior by a two-phase mechanism. We will describe it in Section 4.5.

4.3 | Accuracy of LAF under various situations

According to the reasons explained in the previous subsection, we chose 600 ms as the value of T_{RTT} and 10 ms as the value of $T_{forward}$ in our LAF accuracy experiments.

We set up our experiments in a university campus with more than 10 000 students. According to the observations discussed in Section 4.1, we wrote a python program in our client hosts. The program made five TCP/IP connections to a remote website server specified by us per minute. When an attacker sets up an evil twin, he can use the same channel that the related good twin uses or a different channel to launch an evil twin attack. Our LAF algorithm works on both cases. To test the robustness of LAF, in all subsequent experiments, the evil twin uses the same channel that the good twin uses. We examined the false positives, false negatives, true positives, and true negatives of LAF to evaluate its detection accuracy.

In our evaluation, various experiments were made in different scenarios. Two notebooks were used to simulate two normal users. One notebook chose the good twin to connect to website www.google.com, and the other chose the evil twin to connect to website tw.yahoo.com. Figure 7 shows the above setup which imitated the common situations in which different users surf different websites at the same period of time. We made two sets of experiments. Each set of experiments lasted for 30 min. In the first set of experiments, LAF was used to examine how many times the good twin (AP_2) will be deemed as an evil twin (false positive) and how many times the good twin (AP_2) will be deemed as a good twin (true negative). In the second set of experiments, LAF was used to examine how many times the evil twin (AP_1) will be deemed as an evil twin (true positive) and how many times the evil twin (AP_1) will be deemed as a good twin (false negative). At each set of experiments, these two client hosts continued connecting to the chosen remote servers five times per minute.

In the beginning, we find an appropriate repeat number for LAF. Besides, we also examine the robustness of LAF under various volumes of traffic and different packet loss rates. The first issue we discuss is repeat numbers. A larger

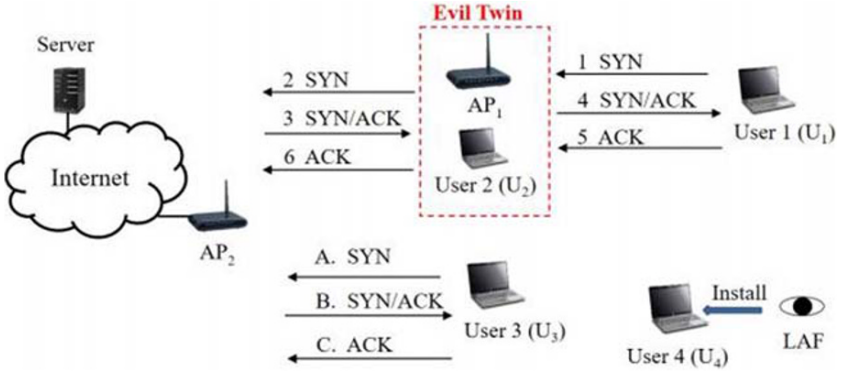

FIGURE 7 Experimental setup. User 4 could be any notebook that has installed LAF. AP, access point; LAF, legal AP finder

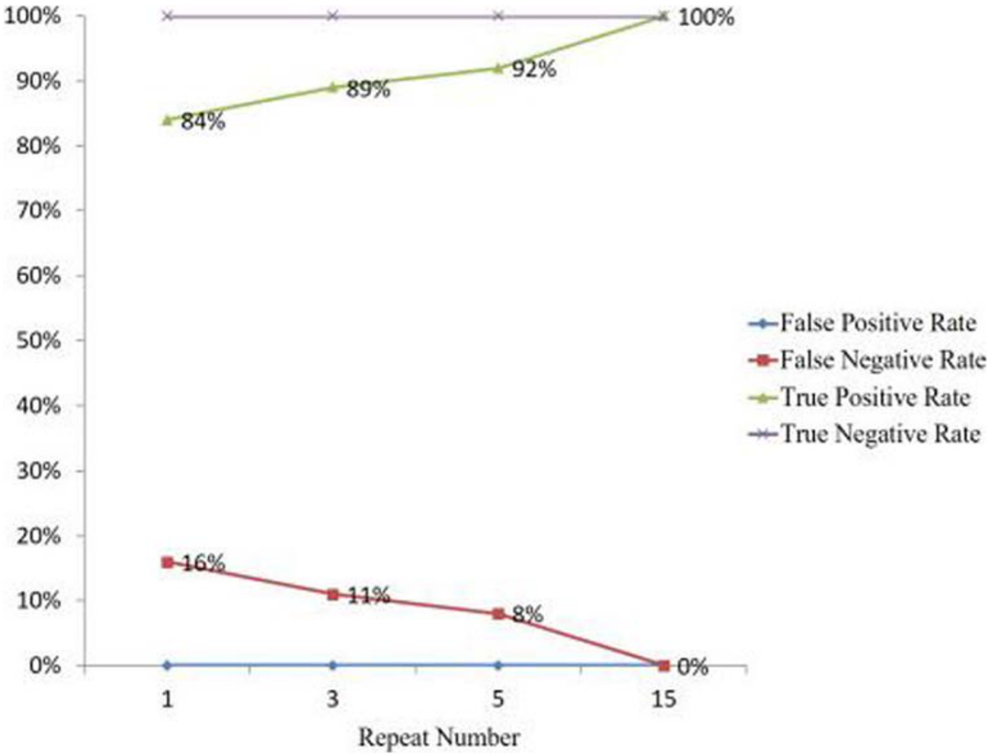

FIGURE 8 Accuracy of legal access point finder with various repeat numbers. False positive rate means the rate that a good twin is deemed as an evil twin. False negative rate means the rate that an evil twin is reported as a good twin. True positive rate represents the rate that an evil twin is correctly detected as an evil twin. True negative rate means the rate that a good twin is correctly detected as a good twin

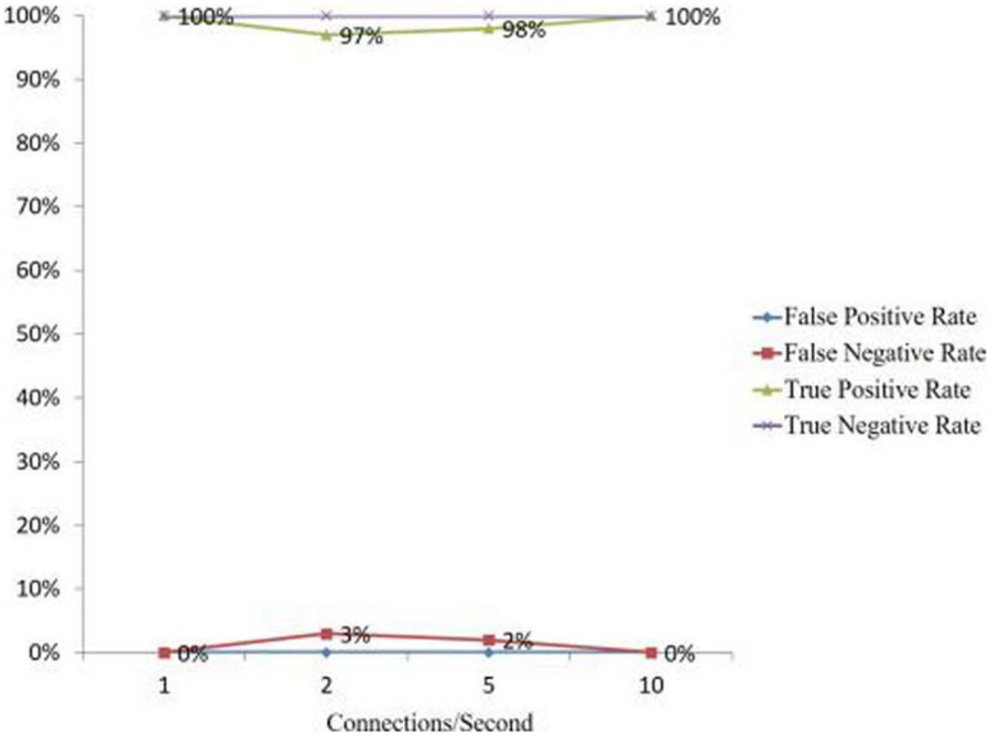

FIGURE 9 Accuracy of legal access point finder under various volumes of network traffic

repeat number should increase the accuracy of LAF. To examine the impact of larger repeat numbers, we chose repeat numbers 1, 3, 5, and 15 to evaluate the accuracy of LAF. For each repeat number, we made our experiments 100 times. Figure 8 displays the results which show that the higher the repeat number is, the more accurate LAF is. When the repeat number is 15, both the false positive rate and the false negative rate are 0%. As a result, we chose 15 as the value of the repeat number of LAF.

To evaluate the accuracy of LAF under different volumes of traffic, we created network traffic using two clients which opened 1, 2, 5, or 10 TCP/IP connections per second. Figure 9 displays the results which show that traffic volumes only have small influence on the accuracy of LAF.

To evaluate the accuracy of LAF under different packet loss rates due to degraded signal strength, channel switches, or noise, we examined the false positive rate, false negative rate, true positive rate, and true negative rate of LAF when the packet loss rate is 0%, 5%, 10%, 15%, 20%, or 50%. In this experiment, two client hosts were used. Each client opened a TCP/IP connection per second. Figure 10displays the result which shows that packet loss rates have great influence on the accuracy of LAF. However, according to Awoniyi and Tobagi,[47] a network with packet loss rate greater than 20% is unusable. When the packet loss rate is 15%, the false positive rate of LAF is 0% and the false negative rate of LAF is 13%. Hence, LAF still functions well in a hostile environment.

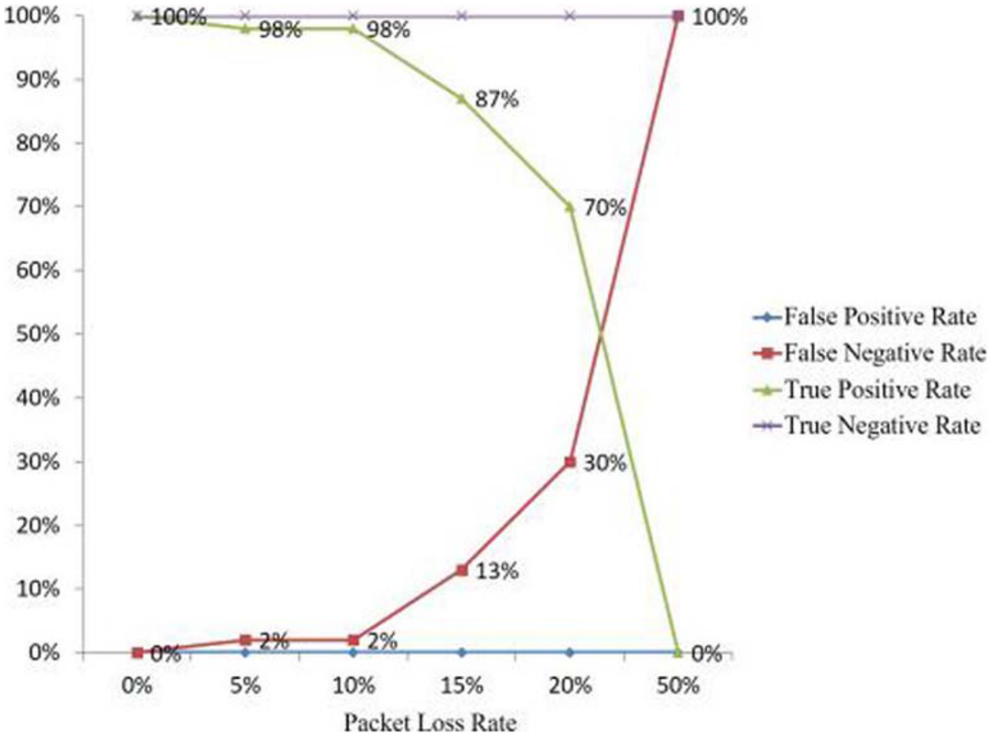

FIGURE 10 Accuracy of legal access point finder under various packet loss rates

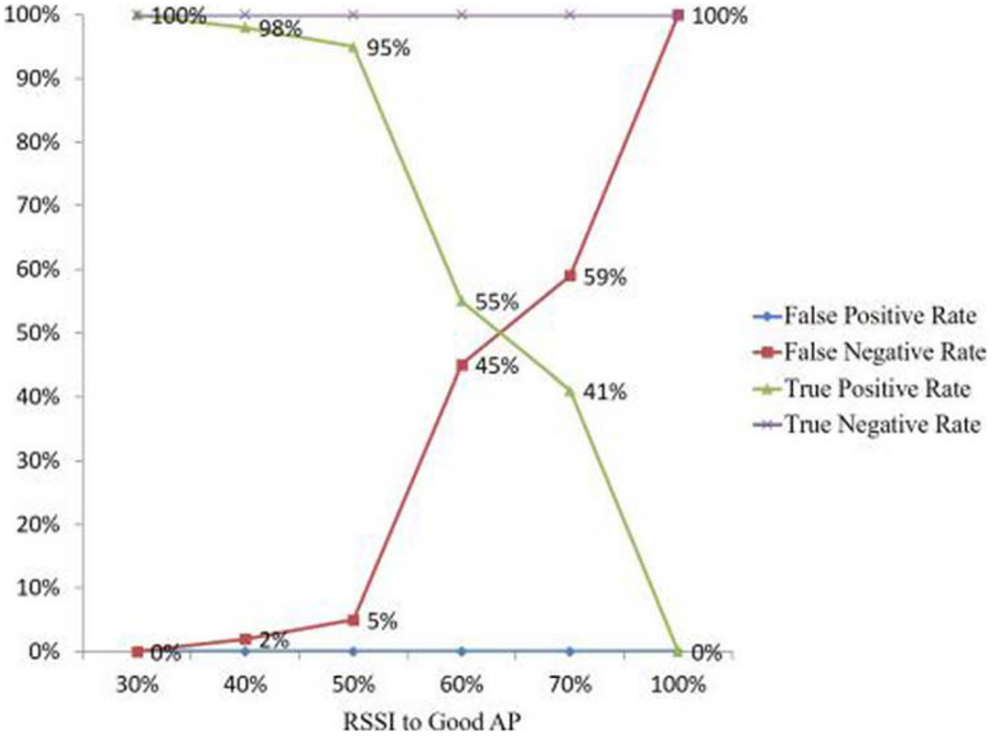

FIGURE 11 Accuracy of legal AP finder with different RSSI values. AP, access point; RSSI, received signal strength indication

We also evaluate the detection accuracy of LAF with different RSSI values.Figure 11 shows the results which show that LAF is accurate in detecting evil twins when the RSSI values are higher than 50%. However, its detection accuracy decreases dramatically if the RSSI value is lower than or equal to 50%. Hence, when an LAF user finds that the RSSI value of one of a pair of his hotspot APs which have the same SSID is lower than 50%, he can temporarily carry his portable device to a location that have a RSSI value higher than 50% to solve the accuracy problem. Besides, as discussed in Section 4.5, a user can also use a secondary device to solve the accuracy problem.

4.4 | Comparisons with other work

In this subsection, we compare LAF with our previous work[37] and a traditional time metrics-based solution.[48] Our previous solution exploited the characteristics that a traditional evil twin forwards a received IP packet directly. Hence, the TCP sequence number and acknowledgment numbers of an IP packet received by an evil will be the same as the TCP sequence number and acknowledgment numbers of the forwarded version of the received IP packet handled by the evil twin. However, if attackers could modify the TCP sequence numbers or acknowledgment numbers of forwarded IP packets when they launch an evil twin attack, it is possible to bypass the detection. In contrast to these kind of solutions, LAF does not need to associate with an AP to make its detection. An attacker is not able to find that it is monitored by LAF, let alone adopt any counter-detection action immediately. Han et al. used time metrics algorithm to detect rouge APs. They found that the detection rate could reduced from 100% to 60% under heavy traffic conditions. Kuo et al.[48] also show that time-based methods could be affected by several factors such as the RSSI and data traffic load. Since LAF does not depend on any time metric, these factors have little effect on the detection rate of LAF. Table 2 shows comparisons among these methods.

4.5 | Discussion

In this subsection, we discuss various problems that LAF may encounter. First, LAF detects an evil twin based on the forwarding phenomenon of the three-way handshake-related packets. However, a WLAN administrator may use a wireless distribution system to extend the coverage area of a WLAN. Hence, two legal APs with the same SSID do exist in some WLANs. In such a WLAN, if LAF is in the overlapped area of the signal of these two APs, then LAF will deem one legal AP as a good twin and the other legal AP as an evil twin. Moreover, even if this situation happens, this will just cause a user to choose the AP deemed as a good twin. Hence, the user still can connect to the Internet through the WLAN safely.

Second, if there is no network traffic in a WLAN, any passive solution, such as LAF, will not be able to function as expected. However, if a user has two wireless devices, such as a laptop and a smartphone, he can use the smartphone to generate some traffic for LAF to analyze. Using a secondary device can also solve the accuracy problem when the RSSI value is lower than 50%. When the RSSI value of an AP is less than 50%, a user can use a secondary device to connect to a nonpopular web page to check whether an AP is an evil twin by observing the destination IP fields of outgoing packets to see if they are forwarded by an AP.

Third, Microsoft Network Monitor is no longer maintained. Hence, in our future work, we plan to write device drivers to implement the channel switching APIs on Windows/Linux so that we can implement LAF on modern Linux and Windows System. Of course, we could add a Wi-Fi USB adapter to monitor the different channel regardless if they are using Windows or Linux. This kind of external USB Wi-Fi adapter is very cheap and can be find in a wide range of shops, including electronics stores and office supply stores. Fourth, even though current evil twin setup utilizes good

TABLE 2 Comparisons between LAF and previous works

	LAF	Hsu et al.[37]	Kuo et al.[48]
Detection time (s)	5–60	5–60	60
Affected by network traffic	Little	Little	Great
Bypass detection	Hard	Easy	Easy

[a]Abbreviation: LAF, legal access point finder.

twins to connect to the Internet, we cannot rule out the possibility that future evil twins may use 3/4/5 G mobile network or other approaches to connect to the Internet. Hence, new approaches must be developed to handle these new setups.

Fifth, for the case of multiple APs with an identical SSID, LAF still works in this case by repeating the detection for each AP. The only issue in this case is the increasing of detecting time. This case happens when enterprises or organizations provide WLAN network for public users. According to the consideration of cost, the deployment strategy must be efficient. It is not reasonable to make those APs installed closely in order to reach the maximum wireless signal coverage. Therefore, detection among these APs with a same SSID is not remarkable.

As discussed in Section 3.4, if an evil twin owner holds a received SYN packet for more than $T_{forward}$ before forwarding it to the good twin and the SYN/ACK packet of the SYN packet returns to the client host in T_{RTT}, the evil twin will be deemed as a good twin. However, LAF can adopt a two-phase mechanism to solve this problem. In the first phase, LAF operates as usual and recognizes some hosts as good twins. In the second phase, LAF only observes the recognized good twins but uses a high $T_{forward}$ that is greater than the time that the client host will abort a TCP/IP connection establishment request. If the evil twin holds the SYN for more than this new $T_{forward}$, the client host will not try to connect to the Internet through the evil twin. If the evil twin holds the SYN packet for less than $T_{forward}$, LAF can recognize this AP as an evil twin.

5 | CONCLUSION

In this paper, we propose a passive user-side legal AP finder, LAF, which can prevent a wireless user from using an evil twin to connect to the Internet, which in turn reduces a lot of security threats. LAF is a light weight solution; hence, a user can use it whenever he needs at any place without the assistance from the administrators of a WLAN, network trace of the related network, or a legal AP/IP list.

Compared with other active user-side solutions, LAF exploits the characteristics of TCP three-way handshake to detect evil twins. It is difficult for an evil twin to evade LAF's detection because every TCP connection must go through the three-way handshake steps. As a solution that does not depend on any time metric, the detection accuracy of LAF is not influenced by network topologies, traffic volume, network types, and prefetching mechanism. Besides, LAF is a passive solution; hence, an evil twin cannot detect its existence, let alone take any step to bypass its detection. Various proofs show that LAF can accurately find a legal AP through using only few wireless packets.

ACKNOWLEDGMENT

This work was partially supported by Ministry of Science and Technology, Taiwan, under Grant MOST 105-2221-E-008-074-MY3.

ORCID

Chuan-Sheng Wang https://orcid.org/0000-0002-0739-6891
Chih-Wen Ou https://orcid.org/0000-0002-8310-5283

REFERENCES

1. U. S. D. of Homeland Security. Tips for using public wi-fi networks; 2014.
2. iPass. Wi-fi growth map; 2015.
3. Chris Hails. Smartphones and public Wi-Fi evil twin attacks. Available from: https://blog.netsafe.org.nz/2011/04/28/smartphones-and-public-wi-fi-evil-twin-attacks/
4. CNN. Evil twin threat to Wi-Fi users. Available from: https://edition.cnn.com/2005/TECH/internet/01/20/evil.twins/
5. Erin Biba. Does your Wi-Fi hotspot have an evil twin. Available from: https://www.pcworld.com/article/120054/article.html
6. Scams Inc. Evil twin attacks: scamming wireless network users. Available from: http://scamsinc.com/2012/02/13/evil-twin-attacks-scamming-wireless-network-users/
7. Hack WiFi. Rogue AP dangers wireless evil twin attack techniques. Available from: https://www.freehowtohackwifi.com/advanced-wifi-hacks/rogue-ap/
8. Shmoo. Airsnarf - A rogue AP setup utility. Available from: https://airsnarf.shmoo.com/
9. Bellardo J, Savage S. 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In: Proceedings of the 12th conference on usenix security symposium, SSYM'03, vol. 12. USENIX Association; 2003; Berkeley, CA, USA.

10. Yang C, Song Y, Gu G. Active user-side evil twin access point detection using statistical techniques. *IEEE Trans Infor Forensics Sec*. 2012;7(5):1638-1651.
11. Airwave. The airwave project. Available from: https://www.airwave.com
12. Motorola Solutions. TIRED OF ROGUES? Solutions for detecting and eliminating rogue wireless networks white paper. Available from: https://www.motorolasolutions.com/web/Business/Products/Software0Applications/Networkments/Static_files/Tired_of_Rogues.pdf
13. Cisco. Cisco wireless lan solution engine (wlse) white paper. Available from: https://www.cisco.com/c/en/us/products/cloud-systems-management/ciscoworks-wireless-lan-solution-engine-wlse/index.html
14. Proxim. Rogue access point detection: Automatically detect and manage wireless threats to your network white paper. Available from: https://www.proxim.com
15. AirMagnet. The AirMagnet project. Available from: https://www.airmagnet.com/
16. Netstumbler. The Netstumbler project. Available from: https://www.netstumbler.com
17. Sheng Y, Tan K, Chen G, Kotz D, Campbell A. Detecting 802.11 mac layer spoofing using received signal strength. In: Infocom. IEEE; 2009:1768-1776.
18. Jana S, Kasera SK. On fast and accurate detection of unauthorized wireless access points using clock skews. *IEEE Trans Mob Comput*. 2010;9(3):449-462.
19. Brik V, Banerjee S, Gruteser M, Oh S. Wireless device identification with radiometric signatures.In: Mobicom Garcia-Luna-Aceves JJ, Sivakumar R, Steenkiste P, eds. ACM; September 22, 2008:116-127.
20. Bahl P, Chandra R, Padhye J, et al. Enhancing the security of corporate wi-fi networks using dair. In: Proceedings of the 4th international conference on mobile systems, applications and services, MobiSys '06. ACM; 2006; New York, NY, USA:1-14. https://doi.org/10.1145/1134680.1134682
21. Beyah RA, Kangude S, Yu G, Strickland B, Copeland JA. Rogue access point detection using temporal traffic characteristics. In: Globecom. IEEE; 2004:2271-2275.
22. Yin H, Chen G, Wang J. Detecting protected layer-3 rogue aps. In: Broadnets. IEEE; November 12, 2008:449-458.
23. Baiamonte V, Papagiannaki K, Iannaccone G. Detecting 802.11 wireless hosts from remote passive observations. In: Networking Akyildiz IF, Sivakumar R, Ekici E, de Oliveira JC, McNair J, eds., Lecture Notes in Computer Science, vol. 4479. Springer; January 21, 2008:356-367.
24. Corbett C, Beyah R, Copeland J. A passive approach to wireless nic identification. In: Icc. IEEE; 2006:2329-2334.
25. Han H, Sheng B, Tan CC, Li Q, Lu S. A timing-based scheme for rogue ap detection. *IEEE Trans Parallel Distrib Syst*. 2011;22(11):1912-1925.
26. Ma L, Teymorian AY, Cheng X. A hybrid rogue access point protection framework for commodity wi-fi networks. In: Proc. ieee infocom; 2008.
27. Mano CD, Blaich A, Liao Q, et al. Ripps: rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning. *ACM Trans Inf Syst Secur*. February 29, 2008;11(2).
28. Shetty S, Song M, Ma L. Rogue access point detection by analyzing network traffic characteristics. In: Military communications conference, 2007. milcom 2007. ieee; 2007:1-7.
29. Venkataraman A, Beyah R. Rogue access point detection using innate characteristics of the 802.11 mac. In: Securecomm Chen Y, Dimitriou T, Zhou J, eds., Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 19. Springer; 2009:394-416.
30. Watkins L, Beyah RA, Corbett CL. A passive approach to rogue access point detection. In: Globecom. IEEE; March 21, 2009:355-360.
31. Wei W, Jaiswal S, Kurose J, Towsley D. Identifying 802.11 traffic from passive measurements using iterative bayesian inference. In: In proc. ieee infocom; 2006.
32. Wei W, Suh K, Wang B, Gu Y, Kurose J, Towsley D. 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, IMC '07. ACM; 2007; New York, NY, USA:365-378. https://doi.org/10.1145/1298306.1298357
33. Wei W, Wang B, Zhang C, Kurose J, Towsley D. Classification of access network types: ethernet, wireless lan, adsl, cable modem or dialup. *Comput Netw*. 2008:3205-3217.
34. Wei W, Jaiswal S, Kurose J, Towsley D, Suh K, Wang B. Identifying 802.11 traffic from passive measurements using iterative bayesian inference. *IEEE/ACM Trans Netw*. 2012;20(2):325-338.
35. Kao K-F, Liao I-E, Li Y-C. Detecting rogue access points using client-side bottleneck bandwidth analysis. *Comput Sec*. 2010;28(3-4):144-152.
36. Hsu F-H, Hsu Y-L, Wang C-S. A solution to detect the existence of a malicious rogue ap. *Comput Commun*. 2019;142-143:62-68.
37. Hsu F-H, Wang C-S, Hsu Y-L, Cheng Y-P, Hsneh Y-H. A client-side detection mechanism for evil twins. *Comput Electrical Eng*. 2015:–.
38. Lu Q, Jiang R, Ouyang Y, Qu H, Zhang J. Bire: a client-side bi-directional syn reflection mechanism against multi-model evil twin attacks. *Comput Sec*. 2020;88:101618.
39. Roth V, Polak W, Rieffel E, Turner T. Simple and effective defense against evil twin access points. In: Proceedings of the first acm conference on wireless network security, WiSec '08. ACM; 2008; New York, NY, USA:220-235. https://doi.org/10.1145/1352533.1352569
40. Song Y, Yang C, Gu G. Who is peeping at your passwords at starbucks? to catch an evil twin access point. In: Dependable systems and networks (dsn), 2010 ieee/ifip international conference on; 2010:323-332.

41. Nicholson AJ, Chawathe Y, Chen MY, Noble BD, Wetherall D. Improved access point selection. In: Proceedings of the 4th international conference on mobile systems, applications and services, MobiSys '06. ACM; 2006; New York, NY, USA:233-245. https://doi.org/10.1145/1134680.1134705
42. Microsoft. Microsoft Network Monitor. Available from: https://www.microsoft.com/en-us/download/details.aspx?id=4865
43. Aikat J, Kaur J, Smith FD, Jeffay K. Variability in tcp round-trip times. In: Proceedings of the 3rd acm sigcomm conference on internet measurement, IMC '03. ACM; 2003; New York, NY, USA:279-284. https://doi.org/10.1145/948205.948241
44. Sessini P, Mahanti A. Observations on round-trip times of tcp connections. *SIMULATION SERIES*. 2006;38(3):347.
45. Gámez RCL, Marti P, Velasco M, Fuertes J. Wireless network delay estimation for time-sensitive applications. *Autom. Control Dept., Technical Univ. Catalonia, Catalonia, Spain, Tech. Rep. ESAII RR-06-12*. 2006.
46. Chen X, Jin R, Suh K, Wang B, Wei W. Network performance of smart mobile handhelds in a university campus wifi network. In: Proceedings of the 2012 acm conference on internet measurement conference, IMC '12. ACM; 2012; New York, NY, USA:315-328. https://doi.org/10.1145/2398776.2398809
47. Awoniyi O, Tobagi FA. Packet error rate in ofdm-based wireless lans operating in frequency selective channels. In: Infocom 2006. 25th ieee international conference on computer communications. proceedings; 2006April:1-13.
48. Kuo E, Chang M, Kao D. User-side evil twin attack detection using time-delay statistics of tcp connection termination. In: 2018 20th international conference on advanced communication technology (icact); 2018:1-1.

How to cite this article: Hsu F-H, Wang C-S, Ou C-W, Hsu Y-L. A passive user-side solution for evil twin access point detection at public hotspots. *Int J Commun Syst*. 2020;33:e4460. https://doi.org/10.1002/dac.4460