International Journal of Wireless Information Networks (2018) 25:130–145
https://doi.org/10.1007/s10776-018-0396-1

# An Efficient Scheme to Detect Evil Twin Rogue Access Point Attack in 802.11 Wi-Fi Networks

Mayank Agarwal[1] · Santosh Biswas[2] · Sukumar Nandi[2]

Received: 7 October 2017 / Accepted: 26 March 2018 / Published online: 29 March 2018

## Abstract

The MAC layer of 802.11 protocol possess inherent weakness making it vulnerable to various security attacks like denial of service, deauthentication attack, flooding attacks, rogue access point (RAP) etc. In this manuscript we focus on evil twin attack. An evil twin is a RAP setup by cloning the MAC address and the Service Set IDentifier of an existing wireless access point (AP). An evil twin is setup so that the client(s) unknowingly connect to them under the pretext that they are connected to a genuine AP. Once a client is connected, an attacker eavesdrops on its communication to hijack client's communication, re-direct clients to malicious websites, steal credentials of the clients connecting to it. Existing methods to detect the evil twin include maintaining white lists, patching AP/client, timing based solutions, protocol modifications etc. These methods usually require extensive setup and maintenance, have scalability and compatibility issues, require changes in protocol stack making them expensive to deploy and manage. The network conditions under normal and evil twin attack are almost similar thereby crafting a signature or defining an anomaly pattern usually leads to large amount of false positives. In this manuscript, we propose an IDS for detecting the evil twin attack, which addresses most of these issues associated with the existing detection mechanisms. Further the scheme is also proved to detect a single evil twin, multiple evil twins for single AP and multiple evil twins for multiple APs. The proposed IDS has been deployed in a lab environment and its detection rate exceeds 92% mark and the accuracy is 100% in all the runs.

**Keywords** Intrusion Detection System · Evil Twin Rogue Access Point Attack · WiFi networks · False alarms

## 1 Introduction

Wi-Fi has seen tremendous growth in the recent years. Various wireless technologies include Wi-Fi, Bluetooth, Infrared, broadcast radio, Microwave radio, satellite communication, Zigbee, wireless sensor networks etc. [1–3] Users enjoy un-interrupted communication at their homes, offices, libraries, coffee-shops while on the move without the hassles of the wired connection [4]. However, all these benefits come at a price. As Wi-Fi communication happens over the air, it can be eavesdropped easily sitting meters away from the actual access point (AP). Wi-Fi networks are prone to a number of attacks like jamming [5], war driving, man-in-the-middle [6], pollution [7], authentication flood, de-authentication flood, MAC spoofing, and many more [8]. One of the principle reasons for the large number of attacks on Wi-Fi networks is due to inherent weakness of the 802.11 protocol. Considering the fact that Wi-Fi would be the primary driver for technologies like Internet of Things (IoT), mobile cloud computing etc. the security issues plaguing the Wi-Fi networks would make these technologies vulnerable to various security attacks [9].

One of the most challenging security breaches to a Wi-Fi network is the rogue access point (RAP). RAP(s) are installed without explicit authorization from a network administrator [10]. By setting up a RAP, an attacker can re-direct clients to fake login portal(s), steal passwords, access credit card information by eavesdropping on communication medium, launch man-in-the-middle attacks etc.

✉ Santosh Biswas
santoshbiswas402@yahoo.com

Mayank Agarwal
agarwalm@post.bgu.ac.il

Sukumar Nandi
sukumar@iitg.ernet.in

[1] Department of ISE, Ben-Gurion University of the Negev, Beersheba, Israel

[2] Department of Computer Science and Engineering, IIT Guwahati, Guwahati, India

**Table 1** Type of rogue access point

| Type of rogue access point (RAP) | How it is exploited |
|---|---|
| Improperly configured AP | Administrators setting default/weak passwords or unknowingly installing faulty/buggy device driver making AP vulnerable |
| Unauthorized AP | Setup without the prior permission of the network administrator |
| Compromised AP | Attacker cracks the encryption key using key cracking tools to evade security |
| Evil twin (Phishing) AP | Clone (mimic) a legitimate Wi-Fi AP |

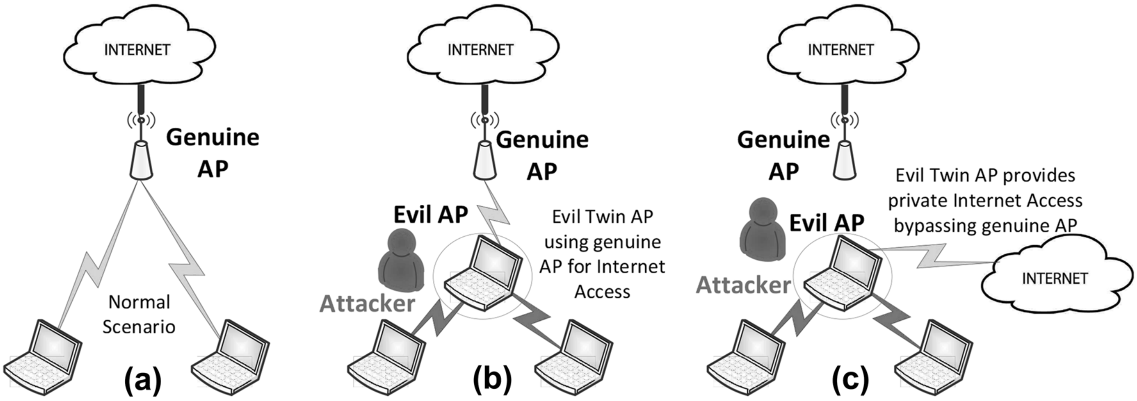


**Fig. 1** Normal and evil twin setup

Ma et al. [11] have given a comprehensive taxonomy of different classes of RAP as shown in Table 1.

In this manuscript, we concentrate on the evil twin AP. An evil twin AP is setup by an attacker to tempt clients into connecting to it and redirecting them to fraudulent websites to steal client information. The attacker spoofs the MAC address and the Service Set IDentifier (SSID) of the genuine AP[1] to setup evil twin. When a client sees the list of available Wi-Fi APs, it sees only one AP instead of two APs as the evil twin AP spoofs both the MAC address and the SSID of the genuine AP. Most of the modern operating systems (OS) are configured to connect to the AP providing higher signal strength in-case there are multiple APs associated with the same SSID. In presence of an evil twin AP, if the signal strength of the evil twin exceeds the signal strength of the genuine AP, the client(s) get associated with the evil twin AP. Higher signal strength leads to higher throughput and less frame loss. Hence a client always prefers to opt for APs offering higher signal strength. All modern operating systems display the list of available APs in order of their signal strengths offered. An attacker usually launches this attack near public Wi-Fi hotspots, libraries, coffee-shops, hotels and other popular Wi-Fi venues.

It must be noted that the evil twin must provide Internet connection to the clients connecting into it. Under normal circumstances, the AP provides the Internet connection as shown in Fig. 1a. If the evil twin does not provide Internet, the clients will dis-connect from the evil twin AP and switch to other APs. For providing Internet, the evil twin can bridge the connection using the genuine AP as shown in Fig. 1b (as genuine AP is already configured to provide Internet services) or the evil twin may provide a private Internet connection all by itself as shown in Fig. 1c. A private connection by the attacker helps it to overcome many existing detection methodologies as explained in the later section. In this work, we assume that the attacker setting up the evil twin provides a private connection. However, our detection methodology is capable to detect the presence of evil twin irrespective of the type of Internet access provided by the attacker.

Existing solutions to detect the presence of evil twin require installation of proprietary hardware [12, 13], protocol changes [13], measuring frame characteristics [14–19] etc. However these methods are either too expensive, require protocol modifications, suffer from high false positive rate and have scalability issues. In this work we have proposed a methodology to detect the evil twin AP with high detection rate and accuracy. The proposed philosophy also addresses the issues plaguing the existing approaches to detect evil twin AP.

The summary of our contributions are:

[1] Genuine AP implies an AP setup by a network administrator.

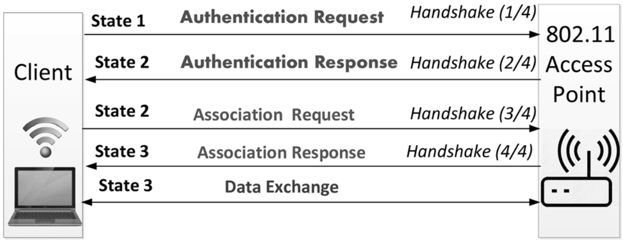

Client
802.11 Access Point
State 1 Authentication Request Handshake (1/4)
State 2 Authentication Response Handshake (2/4)
State 2 Association Request Handshake (3/4)
State 3 Association Response Handshake (4/4)
State 3 Data Exchange

**Fig. 2** Four way handshake between client and access point

1. An IDS to detect the evil twin AP is proposed that adheres to the 802.11 protocol standard. No protocol alternation is needed making the scheme applicable to legacy as well as modern Wi-Fi networks.
2. It is proved that the IDS detects all scenarios of the attack namely, a single evil twin, multiple evil twins for single AP and multiple evil twins for multiple APs.
3. Only a sniffer capable of sniffing Wi-Fi data is needed to detect the evil twin AP. This reduces deployment costs.
4. The overheads incurred due to deployment of the proposed IDS is low which makes the scheme network efficient.

Our manuscript organization is as follows. In Sect. 2 we discuss the basic four way handshake of the client and the AP along with evil twin attack. Following that, a comprehensive literature survey on the existing approaches to tackle the evil twin attack is presented. In Sect. 3 we illustrate the proposed IDS for the evil twin attack. Section 4 deals with the accuracy, detection rate, and the system performance of the IDS. We conclude in Sect. 5.

## 2 Background and Motivation

In this section, we look into the four way handshake that takes place between the client and the AP. We also look into the vulnerabilities that exist during the four way handshake. The evil twin attack is elaborated next. Following that the existing methods to detect the evil twin are described.

The four way handshake between the client and AP is shown in Fig. 2. In Step 1 the client sends an authentication request to the AP. In Step 2, the AP sends back an authentication response. An authentication is unsuccessful only if the AP uses MAC filtering and the requesting client's MAC address is in the AP's blacklist or the AP is overloaded with large number of clients. In Step 3, the client sends an association request to the AP. In Step 4, the AP sends an association response to the client. If all of the above four steps are successful, the client is said to be successfully authenticated as well as associated. Every client that connects to the AP needs to successfully complete this four way handshake.

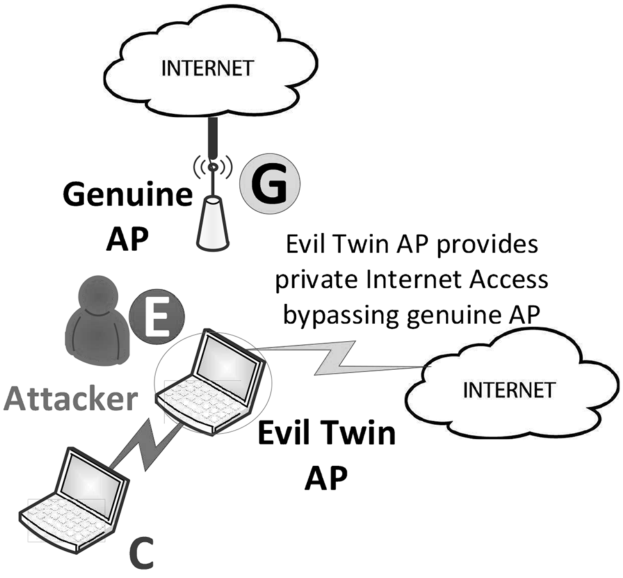

INTERNET
Genuine AP
G
Evil Twin AP provides private Internet Access bypassing genuine AP
E
Attacker
INTERNET
Evil Twin AP
C

**Fig. 3** Evil twin setup

### 2.1 802.11 Management and Control Frames Vulnerabilities

802.11 frames are divided into three categories: Management, Control and Data. The management and control frames are vital which facilitate client's connection establishment, sustainment and termination. The data frames contain the data payload to be exchanged with the AP. 802.11 provides wired equivalent privacy (WEP), Wi-Fi Protected Access (WPA) and WPA2 as encryption mechanisms. Each of the encryption mechanism encrypts only data frames. The management and control frames travel in clear text making them vulnerable to spoofing. All the frames exchanged in the four way handshake explained earlier are management frames making them vulnerable to spoofing. We now describe the evil twin attack in detail.

### 2.2 Evil Twin Attack

The evil twin attack is explained using the network setup shown in Fig. 3 and the timeline in Fig. 4. We assume an open Wi-Fi network. G is the **g**enuine AP, E is the **e**vil twin AP, C is a **c**lient that wishes to connect to G which is unaware of the existence of E. From Fig. 2, it can be seen that the attacker needs to do the following to launch the evil twin attack:

1. Spoof Authentication Response Frame [Handshake (2/4)].
2. Spoof Association Response Frame [Handshake (4/4)].

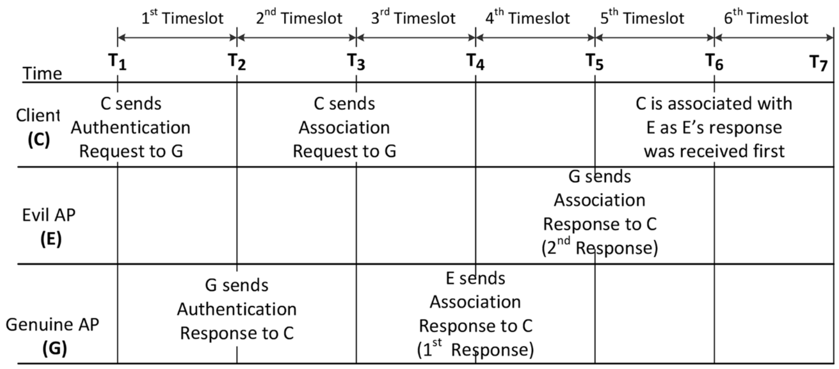

1st Timeslot
2nd Timeslot
3rd Timeslot
4th Timeslot
5th Timeslot
6th Timeslot
Time
T1
T2
T3
T4
T5
T6
T7
Client (C)
C sends Authentication Request to G
C sends Association Request to G
C is associated with E as E's response was received first
Evil AP (E)
G sends Association Response to C (2nd Response)
Genuine AP (G)
G sends Authentication Response to C
E sends Association Response to C (1st Response)

**Fig. 4** Timeline for launching evil twin attack

We now explain how the attacker achieves these goals using the timeline shown in Fig. 4. Each of the distinct time-slots shown in above timeline are explained as follows:

1. [1st *Timeslot* $(T_1)$ – *Handshake*$(1/4)$] C sends authentication request frame to G. E sniffs the authentication request from C.
2. [2*nd Timeslot* $(T_2)$ – *Handshake*$(2/4)$] (Spoof Authentication Response Frame) G replies to C with a successful authentication response[2]. C is now authenticated to G. Authentication process in an open Wi-Fi network does not involve any sort of key exchange. It involves a simple frame exchange between the client and the AP. Sending an authentication response by E might expose E to be detected by IDS. E does not explicitly authenticate C as G would anyways authenticate C. E maintains a list of clients authenticated to G. E also keeps a record of the parameters sent by G to C when G sends an authentication response to C. E then readies itself expecting a possible association request from C.
3. [$3^{rd}$ *Timeslot* $(T_3)$ – *Handshake*$(3/4)$] C sends an association request frame to G.
4. [4*th Timeslot* $(T_4)$ – *Handshake*$(4/4)$] (Spoof Association Response Frame) E sends a successful association response[3] to C.
5. [5*th Timeslot* $(T_5)$] G too sniffs an association request frame from C sent in time-slot $(T_3)$. So, G also sends an association response to C.

As seen above, just the injection of one association response is enough to launch the evil twin attack making it a stealthy attack. Here the important point to note is that the client C associates with the AP whose association response it receives first (E in this case). The first response could be either from the genuine AP (G) or evil twin AP (E) depending on their proximity to C.

Now we look into the existing schemes for detecting evil twin attack. These can be broadly classified into three categories:

## 2.3 Existing Schemes for Detecting Evil Twin Attack

### 2.3.1 Monitoring Wireless Traffic

Most of the wireless-side solutions try to deploy sniffers across the network periphery to gather vital information statistics like MAC address of AP, SSID, RSSI values, operating channel etc. The information collected using these sniffers can help the network administrators to detect the RAPs. Kao et al. [20] detect the presence of rogue AP by maintaining a white-list of the authorized MAC addresses of legitimate APs of the networks. The sniffer monitors the wireless traffic continuously and if an AP is found whose MAC address is not in white-list, it is marked as rogue AP. Sriram et al. [21] and Chirumamilla et al. [22] proposed an agent based IDS for finding out RAP(s). The task of the agent is to monitor the network and detect the presence of new APs. New APs not listed in authorized lists are marked as RAP(s). In evil twin AP, both the MAC address and the SSID are spoofed. As a result, both these techniques fail as evil twin's MAC address matches with the MAC address of the authorized AP.

### 2.3.2 Feature Extraction and Timing Based Solution

In the frame analysis technique, the system first aggregates/logs all the frames using the mirror port of a core switch or by analyzing the frames received from the wireless sensors deployed throughout the network segments. Using the information from the collected frames, various frame features are extracted to gain vital information regarding the presence or absence of evil twin APs.

Some authors assume that evil twin AP forms a bridge between the genuine AP and a client to provide Internet

[2] Henceforth in the manuscript an authentication response would mean a successful authentication response.

[3] Henceforth in the manuscript an association response would mean a successful association response.

services. Due to bridging an additional hop is introduced. Timing based solutions work upon the extra delay occuring due to the additional hop. This extra delay provides evidence for detection of evil twin AP. Han et al. [16] have proposed a method in which they measure the Round Trip Time (RTT) for a DNS query. Mano et al. [18] have also made use of a local RTT metric in addition to a frame payload slicing technique to detect evil twin APs. Yimin et al. [19] have proposed the use of inter arrival time (IAT) to capture the additional delay induced by evil twin. Again these scheme fail when evil twin AP provides its own private connection circumventing the delay effect induced due to the additional hop.

Kohno et al. [23] have shown that clock skews can be effectively used to generate a reliable fingerprint of a device. Jana et al. [17] have used the clock skew to detect unauthorized APs present in the network. They calculate the clock skews of various APs present using the IEEE 802.11 time synchronization function (TSF) timestamps sent out in the beacon frames. If the clock skew calculated for a device differs from the existing clock skews maintained in master database, it is termed as evil twin AP. Bratus et al. [24] showed that it is possible for an attacker to synchronize the TSF of an evil twin AP with that of the legitimate AP inorder to pass the clock skew test circumventing the detection procedure.

The feature extraction and timing based solutions listed above suffer from the following drawbacks. First, it results in an additional load on the core switch due to the additional burden of feature processing. Second, the approach may result in lot of false positives in case the frames are queued by a busy router. The queuing delay of the frames at the busy router results in the additional delay; this results in the techniques using delay based detection (RTT/IAT) to report falsely the presence of evil twin AP. Third, if the attacker provides its own private Internet connection, the traffic does not even reach the core switch, leaving the attack undetected. Fourth, an attacker can simply send a spoofed response to the client, so as to avoid the time delay that may incur due to the additional hop.

#### 2.3.3 Proprietary Hardware

Pradip et al. [13] propose the use of a special probing unit that sends a pre-detection message to all associated client(s) informing them not to respond to probe request. It then sends a probe request. All APs responding to it are marked as evil twin APs. This technique suffers from two drawbacks. Not responding to 802.11 probe requests results in violation of the 802.11 standards [25]. Secondly, an attacker may not respond to probe request to remain hidden which renders the scheme useless.

A signature based IDS refers to a database of known intrusion patterns typically known as signatures for detecting intrusions. A signature based IDS can detect only those attacks whose signature is present in the database. An anomaly based IDS builds a normal profile of a host or network based on statistical evaluation. Unlike signature based IDSs, anomaly based IDSs have the capability to detect both known and unknown attacks. A review of various anomaly based IDSs can be found in [26]. Evil twin attack does not differ in semantics or statistics under normal and attack circumstances thereby crafting a signature or defining an anomaly pattern usually leads to large number of false positives. In short, the existing schemes to detect the evil twin AP suffer from the following drawbacks: (1) Deployment is costly. (2) Protocol modifications are required. (3) Requires proprietary hardware. (4) Patching client software. (5) Result in large number of false positives.

In this manuscript, we propose an IDS for detecting evil twin AP that overcomes the problems associated with the current methods and detects the evil twin AP with high detection rate and accuracy.

## 3 Proposed Scheme: IDS for Evil Twin Attack

In this section, we look into the attacker and IDS assumptions followed by explanation in detail of the principle behind the proposed IDS using an example. We also look into a case-wise analysis of how the proposed IDS can detect all instances of evil twin attack.

### 3.1 Attacker and IDS Assumptions

**Attacker Assumptions**

- *Attacker (evil twin AP) does not send beacon frames neither replies to probe request.*

  Beacon frames are sent by an AP periodically to inform about its presence. The transmission of beacon by the attacker may lead to its detection since beacon frame contains vital characteristics about the network. Methods like [17, 23, 27] use the beacon frame information to detect evil twin AP. By not sending beacons, the attacker overcomes these methods.
- *Attacker does not depend on the genuine AP to provide Internet to the client and uses its own private connection.*

  Most of the rogue AP/evil twin AP setup studied in the literature assume that the attacker is connected to the genuine AP, to provide Internet services to the client. Many techniques [16, 18, 19] use the additional delay induced (due to additional hop traversed to connect to the genuine AP to provide Internet connection) for detecting the evil twin AP. We assume that the attacker provides

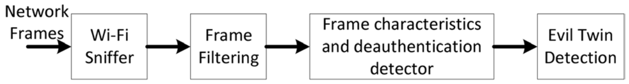


**Fig. 5** Architecture of the proposed IDS

private connection for each client connected to it thereby circumventing these approaches (Fig. 5).

### IDS Assumptions

- *Monitoring of authorized APs only.*
  The IDS monitors only those APs whose MAC addresses are provided by the administrator. Frames traveling to/from other APs are ignored.
- *Genuine AP always replies to client's request.*
  We assume that the genuine AP always replies to client's request and no DoS attack has been performed on it.
- *Proposed IDS has sniffing capabilities.*
  The proposed IDS can sniff Wi-Fi frames traveling in the air in promiscuous mode. This ensures all frames traveling to/from the monitored AP are captured.

## 3.2 Architecture of the Proposed IDS

The architecture of the proposed IDS consists of 4 main components.

- *Wi-Fi sniffer* The sniffer sniffs the Wi-Fi frames traveling in the network. This is a raw sniffer and it dumps all the frames on the Wi-Fi channel it is monitoring.
- *Frame filtering* The raw frames captured by the Wi-Fi sniffer are filtered by this module. It discards the frames that are directed to other APs that are not being monitored. It scans for the 4-way handshake between the monitored AP and the clients, deauthentication frames, association request and response frames. The frame filtering is required in order to make the post-processing efficient.
- *Frame characteristics and deauthentication detector* As seen in the working principle for the detection of the evil twin attack, the receipt of two association frames indicates towards a suspicious activity. On the receipt of two association responses, the order in which the responses were received is observed. Also the frame characteristics like retry bits, sequence number and AID of both responses are analyzed in order to determine the presence of evil twin attack. The deauthentication detector module is used to detect the presence of any deauthentication frame is received for the same client that received the two association responses. This is vital to check because if there is a deauthentication frame present in between the association responses received by the client, it means that the client connected to the AP, then disconnected from it and again reconnected. So,, in such cases, we consider that only 1 response is obtained by the client and not two. If no deauthentication frame is present in between two association responses for a client, then it is considered as receipt of two association responses[4]. Henceforth, we consider only those cases in which no deauthentication frame is present in between two association frames.
- *Evil twin detection* This module alerts to the administrator whether the evil twin attack has taken place or not.

## 3.3 Working Principle of the IDS

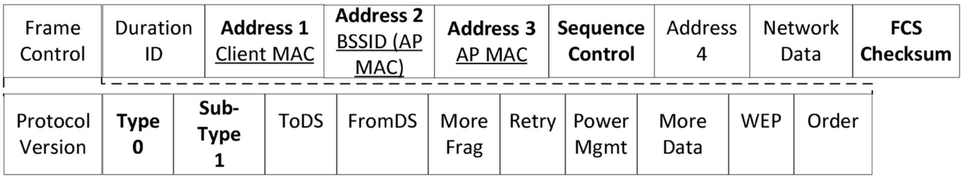
*Working principle* Initially the authentication request and authentication response frames are observed by the IDS for the client. After the IDS sniffs the association request frame it checks whether the client receives one or two association responses. If only one association response is obtained then the network is under normal circumstances. However, if two association responses are obtained then the activity is marked as suspicious and further analysis is done for determining the presence of evil twin. It must be noted that if two association responses are received, it cannot be directly marked as an attack activity as two association responses can also be obtained under normal circumstances (for example, in case the acknowledgment for the first response is not obtained by the AP, the AP retransmits the association response making it a genuine second response). The retry bits, sequence number and association ID (AID) of both responses are analyzed for concluding if evil twin AP is present. The combination of retry bits, sequence number and association ID of the two association responses are enough to detect the presence of evil twin in the network. The algorithm for evil twin detection is shown in Algo. 1. It contains only a brief description of why the particular case leads to detection of evil twin or not.

We will consider the retry bit, sequence number and AID of both the responses. Retry bit can take vales 0 or 1. We represent the retry bit of first and second response as $R_1$ and $R_2$, respectively. AID can take values from 1 to 2007. We consider all 8 cases and show that the evil twin does not escape detection in any of them.

## 3.4 Analysis of Association Response Frame of 802.11 Network

In this subsection, we explain briefly about the function of each field of the association response. The association

[4] Henceforth, we consider only those cases in which no deauthentication frame is present in between two association frames.



| Frame Control | Duration ID | **Address 1** Client MAC | **Address 2** BSSID (AP MAC) | **Address 3** AP MAC | **Sequence Control** | Address 4 | Network Data | **FCS Checksum** |
|---|---|---|---|---|---|---|---|---|

| Protocol Version | **Type 0** | **Sub-Type 1** | ToDS | FromDS | More Frag | Retry | Power Mgmt | More Data | WEP | Order |
|---|---|---|---|---|---|---|---|---|---|---|

**Fig. 6** Association response frame format

response frame format is shown in Fig. 6. As stated in the working principle to detect the evil twin attack, the receipt of two association responses may lead to a possible presence of evil twin attack in the Wi-Fi network. Now we analyze each field of the association response and justify why the sequence number, retry bit and association IDs are important field to detect the evil twin attack.

1. *Duration ID* This field is used to specify the duration in microseconds indicating the amount of airtime the sending radio is reserving for the pending acknowledgment frame. If a client is sending PS-Poll (Power Save Poll) frame this field contains the AID of the frame. So, this field does not provide any significant clue under presence of evil twin.
2. *Address 1, Address 2, Address 3 and Address 4 Fields* In an association response, the Address 1 contains the client's MAC address, Address 2 is Basic Service Set IDentifier (BSSID) address and Address 3 field contains the AP's MAC address. In our case, Address 4 is unused. In case of evil twin, the attacker clones the MAC address of the genuine AP so the Address 2 and Address 3 fields can be easily manipulated. The Address 1 field that contains the client's MAC address can be obtained by the attacker by observing the preceding association request frame. This field provides client information and is useful for the detection of evil twin attack.
3. *Sequence control field* This field is incremented by 1 for every frame that is transmitted. The field is 12 bit in length so the range of sequence numbers are $0-(2^{12}-1)$ i.e., $0-4095$. The start value of the sequence number between a client and the AP varies with the implementation. This field is useful for differentiating the two association responses received in presence of evil twin. The two association responses may have same or different sequence numbers and provides vital clue for the detection of evil twin detection.
4. *Network data* This contains the data payload of the frame. The data payload contains the association ID (AID) assigned by the AP to the client. The AID assigned by the genuine AP and evil twin AP may be same or different. The analysis of AID assigned by the genuine AP and evil twin AP is important for the evil twin detection.
5. *FCS—Frame check sequence* This constitutes the checksum of the frame to be transmitted and is used to check for the integrity of the frame. The checksum can be easily calculated over the frame. So just by verifying the checksum of the association response, the presence of evil twin cannot be determined. So, FCS cannot be used as a parameter for the detection of the evil twin.

   The frame control has various fields under it which are explained below:
6. *Type* and *Sub-type fields* The Type field has its bit set to 0 while the Subtype field has its bit set to 1 indicating that the frame is of association response type. Since the attacker spoofs the association it has to set the Type/Subtype field to 0/1 respectively. So by observing the Type/Subtype fields it is not possible to detect the presence of evil twin in the network.
7. *ToDS* and *FromDS* The ToDS and FromDS fields indicate the direction of frame traveling. If ToDS (FromDS) is set to 1, it indicates that the frame is traveling to(from) AP. For all management and control frames both ToDS and FromDS bits are set to 0. As association response is a management frame, both ToDS and FromDS bits are set to 0. So, observing the ToDS and FromDS bits for detection of evil twin is futile as the attacker sets both these bits to 0 in its association response.
8. *More Frag* A long frame is fragmented into parts. If More Frag is set to 1, it indicates the frame is a fragment of a long frame. The content of an association response frame fits into one frame so there is no fragmentation is required. So, this bit is always 0. Again this field does not help in detection of evil twin attack as the genuine and attacker set the More Frag as 0.
9. *Retry* Every frame that is transmitted for the first time has a Retry bit set to 0. If the frame gets lost (i.e. no acknowledgment is received), it is re-transmitted with Retry bit set to 1. The retry bit plays an important role in the detection of the evil twin attack. The retry bits of the association responses received from the genuine and attacker AP provides significant clue for the detection of evil twin attack.
10. *Power Mgmt* A client sets this bit to 1 and enters into sleep state in order to preserve its battery. A client that is awake has the Power Mgmt bit to 0. Since this frame

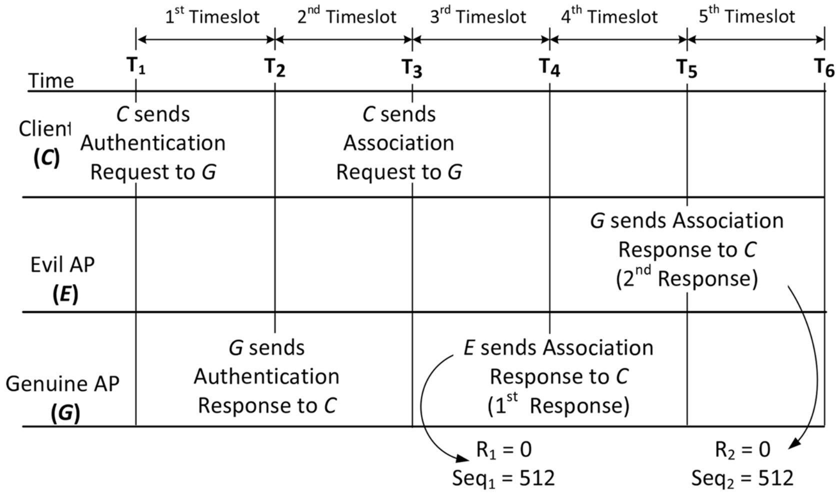

1st Timeslot
2nd Timeslot
3rd Timeslot
4th Timeslot
5th Timeslot
Time
T1
T2
T3
T4
T5
T6
Client (C)
C sends Authentication Request to G
C sends Association Request to G
Evil AP (E)
G sends Association Response to C (2nd Response)
Genuine AP (G)
G sends Authentication Response to C
E sends Association Response to C (1st Response)
R1 = 0
Seq1 = 512
R2 = 0
Seq2 = 512

**Fig. 7** Timeline of detection of evil twin attack

**Table 2** Database view for the *first* response received by the client

| Client MAC | $R_1$ | $R_2$ | $Seq_1$ | $Seq_2$ | $AID_1$ | $AID_2$ | Flag |
|---|---|---|---|---|---|---|---|
| MAC C | 0 | – | 512 | – | 4 | – | 1st Response received |

is sent by the AP its Power Mgmt is always set to 0 (as AP is always awake). So the Power Mgmt bit does not help in detection of evil twin attack as it is always set to 0.

11. *More Data* When set to 1, this indicates to a client in power save mode that it should come out of the power save mode as there are buffered frames at the AP. A client stays awake till it gets associates with the AP. So this bit is always set to 0 in the association response. So the More Data bit does not help in the detection of evil twin.
12. *WEP* If Wired Equivalent Privacy (WEP) encryption is used, then this bit is set to 1 else it is set to 0. As we have assumed open Wi-Fi network in place, this field is always set to 0. So the WEP bit does not help in the detection of evil twin.
13. *Order* If this bit is set to 1, it implies strict ordering is in use. In association response frame, this bit is always set to 0. So the Order bit does not help in the detection of evil twin.

Thus as seen the sequence number, retry bit and association IDs form a vital component for the detection of the evil twin attack. Now we look into the proposed algorithm in order to detect the evil twin AP. The algorithm is elaborated in Algorithm 1. The algorithm considers all possible cases for the detection of the evil twin attack.

*Case 1:* $R_1 = 0, R_2 = 0, Seq_1 = Seq_2$

→ We explain Case 1 shown for the Algo. 1 using the timeline shown in Fig. 7 and the experimental setup shown in Fig. 11. We denote *G* as the genuine AP, *E* as the evil twin AP, *C* as the client that wishes to connect to *G* who is unaware of the existence of *E*. The MAC address of *G*, *E*, and *C* are denoted as MAC *G*, MAC *E*, and MAC *C* respectively. Note that MAC *G* and MAC *E* are identical as *E* is the evil twin AP setup by the attacker. We now explain the database schema that the IDS uses. The following information with regards to the client is stored in the database: Client MAC, $R_1$, $R_2$, $Seq_1$, $Seq_2$, $AID_1$, $AID_2$, Flag. Client MAC is the MAC address of the client that attempts to connect to the AP *G* in presence of evil twin AP *E*. As the IDS only monitors traffic to AP *E* and ignores the traffic sent to other APs, we do not store the AP MAC address in the database (as it would always be MAC *E*). $R_1$, $R_2$, $Seq_1$, $Seq_2$, $AID_1$, $AID_2$ represents the value of the retry bit, sequence numbers and association IDs of the first and second response respectively. Flag stores whether an instance of evil twin attack has occurred or not.

The first four timeslots shown in Fig. 7 are same as those explained in Sect. 2.2. In brief, they indicate the

**Table 3** Database view for the *second* response received by the client

| Client MAC | $R_1$ | $R_2$ | $Seq_1$ | $Seq_2$ | $AID_1$ | $AID_2$ | Flag |
|---|---|---|---|---|---|---|---|
| MAC C | 0 | **0** | 512 | **248** | **4** | **5** | Case 1: Evil twin detected |

four-way handshake which includes the first association response (here we have assumed that the first response is from evil twin). We assume that the first response has retry bit set to 0 (so $R_1 = 0$, Line 2). As this is the first response, the clients information namely, client's MAC address, sequence number ($Seq_1$), retry bit ($R_1$) and association ID ($AID_1$) is stored into the database (Lines 3–4). We assume that the client's MAC address is MAC $C$, sequence number is 512 (so, $Seq_1 = 512$), and association ID is 4 (so, $AID_1 = 4$). The flag field is marked with '1*st* Response received'. Table 2 shows the database view at this point.

Note that $R_2$, $Seq_2$, $AID_2$ of the database are blank as the client just received its first response. At this point, if a deauthentication frame is received for the client, then the database entry for the client is purged and the process of evil twin detection is aborted (Lines 5–6). In this case we have assumed that no deauthentication frame is received. Now a second association response is received for the same client with retry bit 0 (so $R_2 = 0$, Line 7). As this is a second response, the attributes of the first association response stored in the database (i.e., retry bit ($R_1$), sequence number ($Seq_1$) and association ID ($AID_1$)) are retrieved (Lines 9). Since both the retry bits are set to 0 ($R_1 = R_2 = 0$), the sequence number of both responses are compared (Line 10). In this case the sequence number of both responses are identical ($Seq_1 = Seq_2 = 512$) which leads to the detection of evil twin attack (Line 11). Table 3 shows the database view at this point (i.e., after receiving the second association response is).

Reason for the presence of evil twin: As per the 802.11 standard, if a frame is lost then the frame is retransmitted with same sequence number but with retry bit set to 1 (i.e $R_2 = 1$). The sequence number is incremented by 1 in every case except for those cases where the frames are retransmitted. Since the retry bit in the second response is 0 ($R_2 = 0$), it is not a re-transmitted frame. So the sequence number of the second response must be different than the one received in the first response. As the second response has the same sequence number, such a case is possible under an evil twin attack only. The 5*th Timeslot* ($T_5$) in Fig. 7 represents the second association response with retry bit 0 and same sequence number. The 'Flag' column in the database is updated to 'Case 1: Evil Twin detected'. Note that in this case we did not compare the association IDs of both responses as the sequence number alone gives us conclusive evidence for the presence of evil twin attack. However, association IDs is used to detect the presence of evil twin in other cases. In a similar manner the other cases of the occurrence of evil twin attack are detected.

*Case 2:* $R_1 = 0, R_2 = 0, Seq_1 \neq Seq_2$

→ Here, the IDS checks for the de-authentication frame for the client in question between the two association response frames received. If no de-authentication frame was found in between two association response frames, this indicates the presence of evil-twin in the network, since if the genuine AP had lost the frame, it would have retransmitted the lost frame with the same sequence number and $R_{bit}$ set to 1. So, the presence of evil twin is detected here too.

*Case 3:* $R_1 = 0, R_2 = 1, Seq_1 = Seq_2$, $\text{AID}_1 \neq \text{AID}_2$

→ In this case the attacker sends all the frames with $R = 1$. The attacker also forges the sequence number of the genuine AP. As a result, the IDS receives two responses with same sequence number and $R_1 = 0$, $R_2 = 1$. So this represents a frame that has been re-transmitted by the AP to the client. However, when we see the AID of both the responses, they are different. In absence of evil twin, had the genuine AP lost the frame, it would had re-transmitted the same frame keeping each of the parameter values same (including the AID). The difference in AID bit is sufficient to conclude the presence of evil since a genuine AP cannot assign two different AIDs to same client in a re-transmitted frame.

**Algorithm 1:** Algorithm for detecting evil twin AP in the network.

```
Input: $R_1$: Retry Bit of first response, $Seq_1$: Sequence number of first response. $R_2$:
       Retry Bit of second response, $Seq_2$: Sequence number of second response. The
       same client receives two responses.
Output: Evil Twin Present or Absent
begin
    if $R_1$ == 0 then
        Create a database (DB) entry for the client.
        Store the client's MAC address, retry bit ($R_1$), sequence number ($Seq_1$),
        association ID ($AID_1$) in the DB.
        if Deauthentication frame is received for the same client then
            Purge DB entry for the client and exit.
        else if $R_2 == 0$ then
            Store retry bit ($R_2$), sequence number ($Seq_2$), association ID ($AID_2$) in
            the DB for the client.
            Fetch $R_1$, $Seq_1$, $AID_1$ values for the client from the DB.
            if $Seq_1 == Seq_2$ then
                Evil Twin Detected. Mark this as Case 1.
                Flag Entry for the client in the DB.    // Reason: Second Response
                Cannot have Retry Bit = 0 ($R_2$ = 0) if it has the same
                sequence number as the first response. (It has to be 1)
            else
                Evil Twin Detected. Mark this as Case 2.
                Flag Entry for the client in the DB.    // Reason: If, no
                deauthentication frame is present between two respones,
                second response cannot have retry bit = 0 (It has to be 1)
        else if $R_2 == 1$ then
            Store the client's MAC address, retry bit ($R_2$), sequence number ($Seq_2$),
            association ID ($AID_2$) in DB.
            Fetch $R_1$, $Seq_1$, $AID_1$ values for the client from the DB.
            if $Seq_1 == Seq_2$ then
                if $AID_1 \neq AID_2$ then
                    Evil Twin Detected. Mark this as Case 3.
                    Flag Entry for the client in the DB.  // Reason: Different AID
                    for the same client by the same AP is not possible.
                else
                    Evil Twin Detected. Mark this as Case 4.
                    Flag Entry for the client in the DB.  // Reason: Retransmitted
                    Frame (Frames with retry bit set to 1) cannot have
                    different sequence number than original frame.
        else
            If no second association is received,then network is assumed to be under
            normal conditions.
    else //First Response with $R_1 == 1$
        Create a DB entry for the client.
        Store the client's MAC address, retry bit ($R_1$), sequence number ($Seq_1$),
        association ID ($AID_1$) in the DB.
        if Deauthentication frame is received for the same client then
            Purge DB entry for the client and exit.
        else if $R_2 == 0$ then
            Store the client's MAC address, retry bit ($R_2$), sequence number ($Seq_2$),
            association ID ($AID_2$) in DB.
            Fetch $Seq_1$ value for the client from the DB.
            if $Seq_1 == Seq_2$ then
                Evil Twin Detected. Mark this as Case 5.
                Flag Entry for the client in the DB.    // Reason: Same as Case 1
                above.
            else
                Evil Twin Detected. Mark this as Case 6.
                Flag Entry for the client in the DB.  // Reason: Same as Case 2
                above.
        else if $R_2 == 1$ then
            Note the sequence number and association ID of this association response.
            Store it as $Seq_2$ and $AID_2$ respectively.
            Fetch $R_1$, $Seq_1$, $AID_1$ values for the client from the DB.
            if $Seq_1 == Seq_2$ then
                if $AID_1 \neq AID_2$ then
                    Evil Twin Detected. Mark this as Case 7.
                    Flag Entry for the client in the DB.  // Reason: Same as Case 3
                    above.
                else
                    Evil Twin Detected. Mark this as Case 8.
                    Flag Entry for the client in the DB.  // Reason: Same as Case 4
                    above.
        else
            If no second association is received,then network is assumed to be under
            normal conditions.
End
```

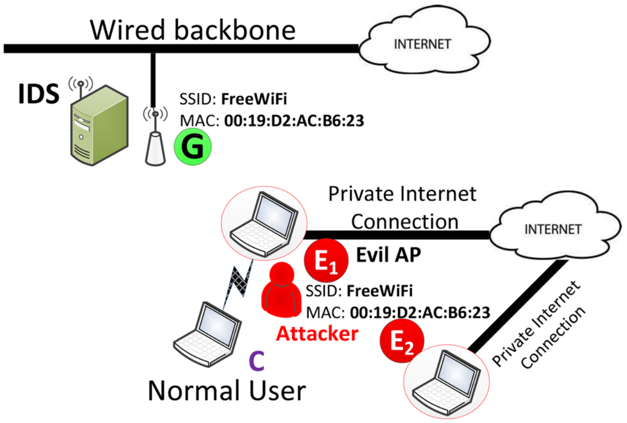


**Fig. 8** Case 1: One genuine AP in the BSS, two evil twin APs setup by an attacker

*Case 4:* $R_1 = 0$, $R_2 = 1$, $Seq_1 \neq Seq_2$

→ As the second response is a re-transmitted one (indicated by $R_2 = 1$), the sequence number must be same as that in the first response. As the sequence number differs in the re-transmitted frame, one of the response must have been sent by the evil twin AP.

*Case 5:* $R_1 = 1$, $R_2 = 0$, $Seq_1 = Seq_2$

→ We are considering the scenario in which the client connects to the AP in 1 hop. As a result there is no out-of-order delivery. Here the initial response the IDS received was with retry bit = 1, this implies that the first association response from the AP was lost. As a result the AP resent the frame with same sequence number. However the IDS receives a second response after this with retry bit = 0 and same sequence number. This is not possible under normal circumstances. Had the AP lost the frame with retry bit = 1, it would have sent another frame with same sequence number and retry bit = 1. Thus in this case too, the IDS is successful in detecting the presence of evil twin.

*Case 6:* $R_1 = 1$, $R_2 = 0$, $Seq_1 \neq Seq_2$

→ This case is similar to the Case 2 explained above. With the similar technique applied in Case 2 the IDS can detect the presence of evil twin. The difference here being is that the first association response frame sent by the genuine AP is lost and hence it resend the same frame with retry bit = 1. The IDS just needs to check the presence of de-authentication frame between the association responses other things are similar to Case 2.

*Case 7:* $R_1 = 1$, $R_2 = 1$, $Seq_1 = Seq_2$, $\mathrm{AID}_1 \neq \mathrm{AID}_2$

**Fig. 9** Case 2: Two genuine AP in the same BSS, one evil twin for each of genuine AP

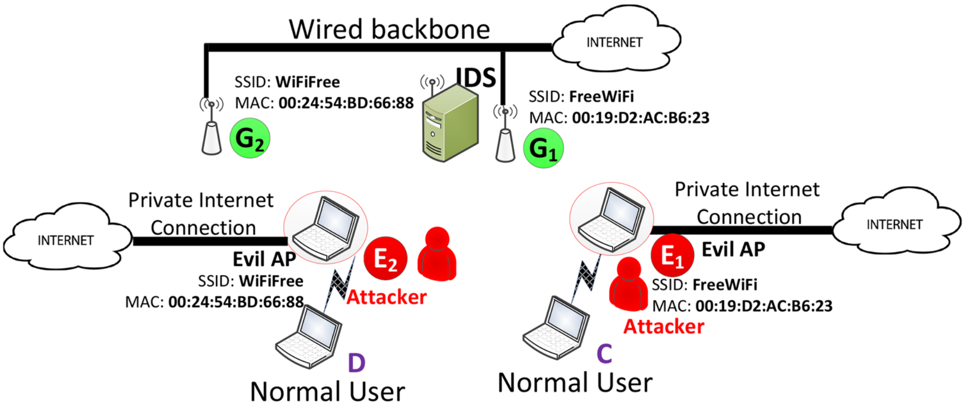
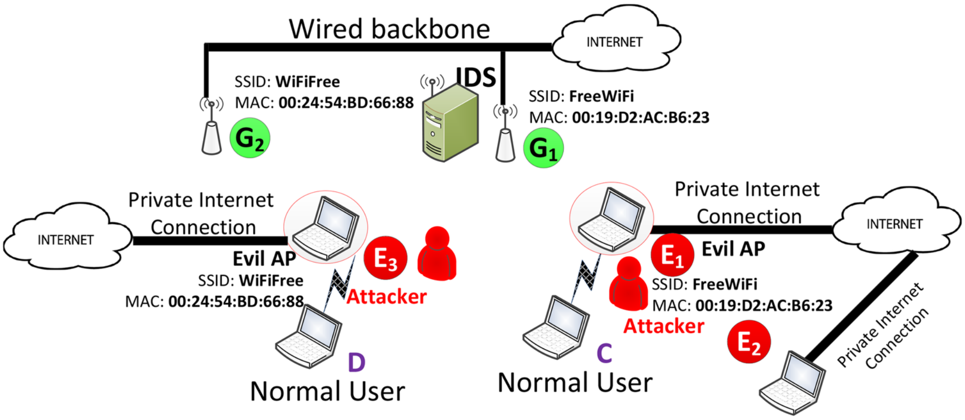


**Fig. 10** Case 4: Multiple genuine APs and multiple evil twin APs; a sub-set of evil twins target some genuine AP



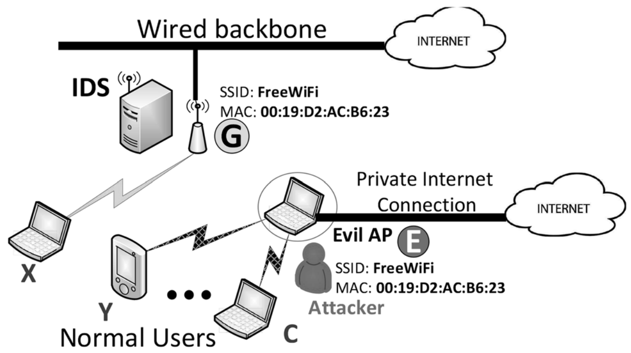


**Fig. 11** Experimental setup for evil twin attack

→ This case is on the similar lines as explained in the Case 3. Here too the IDS can detect the attacker directly based on the difference in AID(s) assigned.

*Case 8:* $R_1 = 1, R_2 = 1, Seq_1 \neq Seq_2$

→ This is just a combination of Case 2 and Case 6. On occurrence of this case, the IDS checks for the presence of de-authentication frames between the association responses. If no de-authentication frame is present in between, IDS raises an alarm, detecting the presence of evil twin.

As seen, only two cases require the checking of the AID (Case 3 and 7). So the evil twin AP escapes detection only when it does not send an association response. If the evil twin AP does not send an association response, the client connects to genuine AP making the evil twin dormant entity. Hence an evil twin AP tries to send a successful association response to every client requesting association with the AP. However sending of association response leads to the detection of evil twin AP by the proposed IDS. Thus, the IDS is successful in detecting all the 8 possible ways in which the attacker can launch evil twin attack.

### 3.5 Extending the Scheme to Detect Multiple Evil APs

In this subsection we show how our proposed methodology can be used to detect the presence of multiple evil APs in the network. The scenarios can be exhaustively enumerated as follows:

1. *Case 1 (One genuine AP and multiple evil twin APs targeting the genuine AP)* First we demonstrate the case when there is one genuine AP and two evil twin APs. Following that, we show the proposed algorithm can be used even if the number of evil twin APs is more than two.

   Let there be one genuine AP in the BSS, two evil twin APs setup by an attacker. This scenario is shown in Fig. 8. As we see that the attacker has setup two evil twins. This leads to increase in the chances of the client getting connected to either of them. Let us call the evil twin APs as $E_1$ and $E_2$, respectively while the genuine is denoted by $G$ as before. APs $G, E_1, E_2$ obviously have the same SSID and MAC address as $E_1, E_2$ are evil twins. The client $C$ when tries to send an association request in such a setup (as shown in Fig. 8), it would receive 3 association responses. One response would be from the genuine AP while the remaining two would be from the two evil twins setup by the attacker. We have already shown that the proposed algorithm (see Algorithm 1) can detect the presence of evil twin in the network once we receive two association responses. Now that we receive a third association response, the administrator just needs to be alerted that multiple evil twins are present in the network.

   Now let us assume that there are $n$ evil twin APs (as $E_1$, $E_2 \cdots$ and $E_n$) along with the genuine AP under the BSS under consideration. When the client when tries to send an association request in such a setup, it would receive $n + 1$ association responses, leading to detection of multiple evil twins in the network.
2. *Case 2 (Multiple genuine APs and multiple evil twin APs, however one evil twin targets one genuine AP)* First we demonstrate the case when there are two genuine APs and two evil twin APs (each targeting one genuine AP). Following that, we show the proposed algorithm can detect attacks, even if the number of genuine and evil twin APs pairs is more than two.

   Let us assume that there are two genuine APs in the same BSS and one evil twin for each genuine AP. This scenario is shown in Fig. 9. Here there are two genuine APs (denoted by $G_1$ and $G_2$, respectively) and an attacker

**Table 4** Detection rate statistics using the proposed IDS

| Run # | Attack instances launched | Instances detected using proposed IDS | Detection rate % | Run # | Attack instances launched | Instances detected using proposed IDS | Detection rate % |
|---|---|---|---|---|---|---|---|
| 1 | 120 | 111 | 92.50 | 6 | 120 | 119 | 99.17 |
| 2 | 120 | 112 | 93.33 | 7 | 120 | 114 | 95.00 |
| 3 | 120 | 115 | 95.83 | 8 | 120 | 120 | 100.00 |
| 4 | 120 | 118 | 98.33 | 9 | 120 | 117 | 97.50 |
| 5 | 120 | 114 | 95.00 | 10 | 120 | 114 | 95.00 |

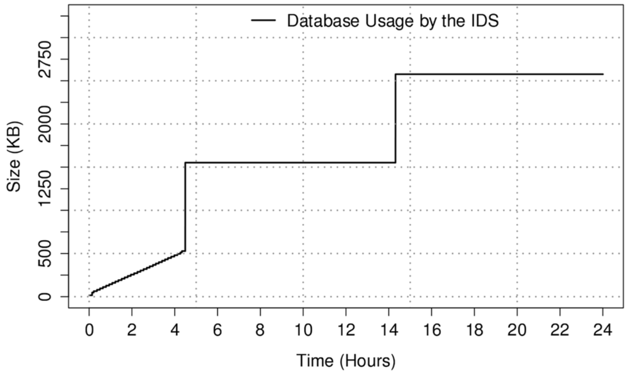


**Fig. 12** Size of database growth of the proposed IDS

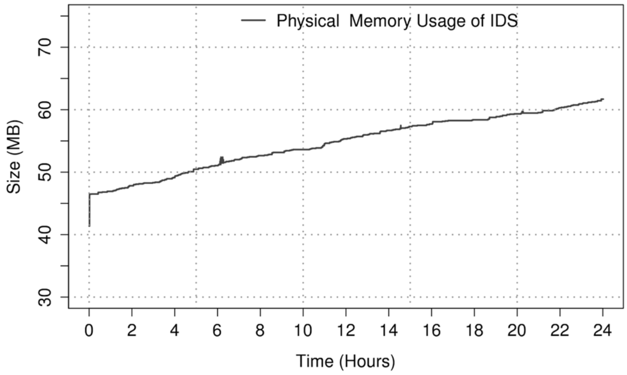


**Fig. 13** Memory usage of the proposed IDS

sets up evil twins for each of them. For the genuine AP $G_1$ let its corresponding evil twin be denoted by $E_1$ while for $G_2$ let its corresponding evil twin be denoted by $E_2$. In our IDS assumptions (see Sect. 3.1) we already stated that the proposed solution monitors only the traffic destined to and from authorized APs. In such a case, both $G_1$ and $G_2$ are added to the authorized lists of monitored MACs for detection of evil twin. When both the APs are monitored the proposed solution checks for the presence of evil twin for both the APs and alerts the administrator when it detects any evil twin.

So, the proposed algorithm can easily detect all ($n$, say) evil twins if each is associated with a genuine AP.

3. *Case 3 (Multiple genuine APs and one evil twin AP)* This is a sub-case of Case 2, where we consider only the pair “Evil Twin and its target genuine AP”.
4. *Case 4 (Multiple genuine APs and multiple evil twin APs; a sub-set of evil twins target some genuine AP)* Let there are two genuine APs (denoted by $G_1$ and $G_2$, respectively) and three evil twins (denoted by $E_1$, $E_2$ and $E_3$, respectively). $E_1$ and $E_2$ are the evil twins for the genuine AP $G_1$ while $E_3$ is an evil twin for $G_2$. The setup for this case is shown in Fig. 10. Hence, this case refers to a case of multiple genuine APs ($G_1$, $G_2$) and multiple evil twin APs ($E_1$, $E_2$ and $E_3$); and a sub-set of evil twins targeting some genuine AP ($E_1$, $E_2$ targeting $G_1$ and $E_3$ targeting $G_2$). Such a case can be detected using the proposed methodology. We can use the combination of

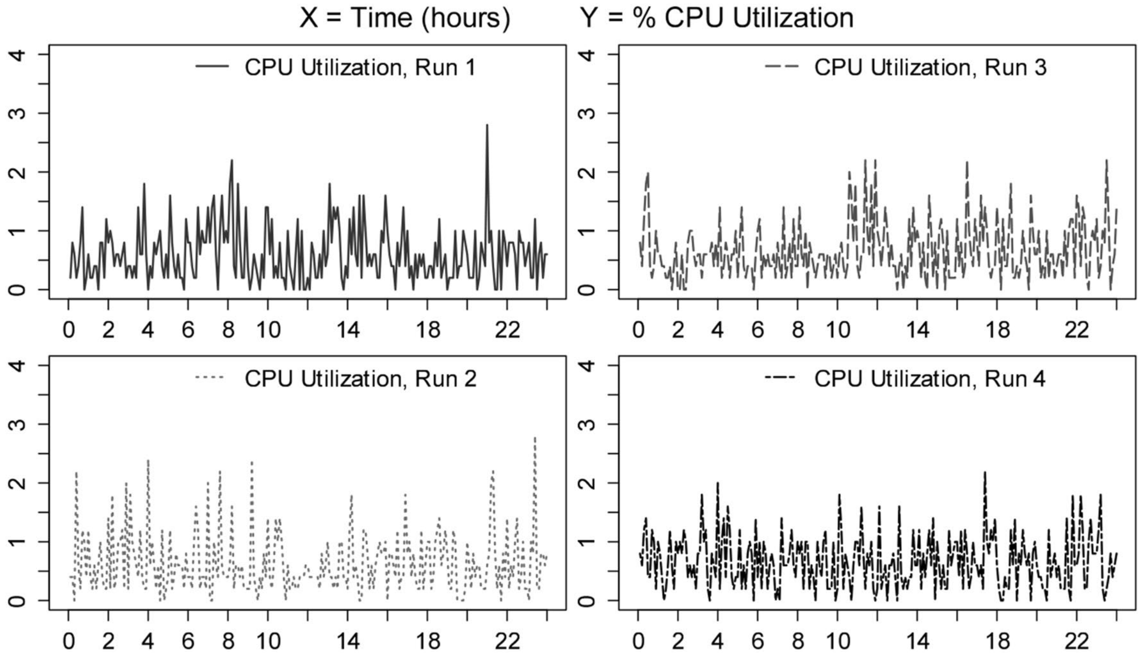


**Fig. 14** CPU utilization of the proposed IDS

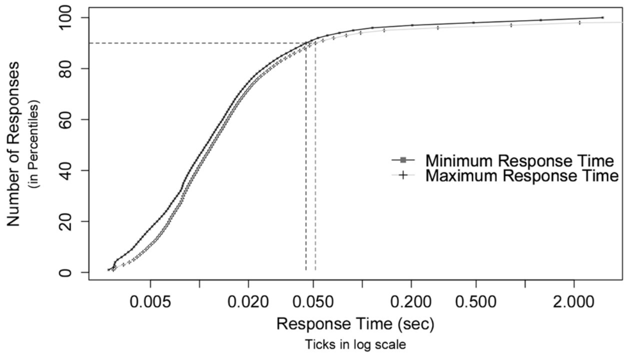


**Fig. 15** Distribution of association response times

Case 1 (for the evil twins $E_1$, $E_2$ targeting $G_1$) and Case 2 (for the evil twin $E_3$ targeting $G_1$).

It is obvious that if one evil twin ($E_1$) targets $G_1$ and two evil twins ($E_2$, $E_3$) target $G_2$, the situation can be detected using Case 2 on the former and Case 1 on the later.

As already discussed in Case 1 and Case 2, that the detection methodology works for $n$ number of evil twins. So, Case 4 (which is a combination of Case 1 and Case 2) will also work for any number of evil twins. Also, the detection methodology will work for any number of genuine APs, by applying Case 1 and Case 2 appropriately on each genuine AP.

Thus a single evil twin, multiple evil twins for single AP and multiple evil twins for multiple APs, all can be detected by the proposed scheme making it a robust solution for the detection of evil twin attack.

## 4 Experimental Results

In this section we describe the network setup for evil twin attack. We also show the accuracy, detection rate, CPU and memory utilization of the proposed IDS.

### 4.1 Network Setup, Accuracy and Detection Rate of Proposed IDS

The network setup for the evil twin attack is shown in Fig. 11. We denote C as the client, G as the genuine AP with SSID as “FreeWiFi” having the MAC address as 00:19:D2:AC:B6:23 running on channel 6 and E as the evil twin AP. The evil twin AP is setup using the *airbase-ng* utility of the *aircrack-ng* suite [28]. To obtain the MAC address, SSID and the channel number on which the genuine AP is running the attacker analyzes the information sent in the beacon frames by the genuine AP. The SSID, MAC address and the channel on which the evil twin runs is identical to that of the genuine AP. The evil twin AP is configured on a machine having Kali Linux operating system. The IDS is located close to the genuine AP so that frames traveling to and from the genuine AP are not lost. The IDS is implemented in C language running on a machine with Ubuntu 16.04 installed. We used smartphones, laptops and desktops equipped with wireless cards as clients for testing the the IDS. The vital information required for detecting the presence of evil twin are extracted from the association request and response frames and stored in the MySQL database. The experiments lasted for a period of 24 h which is later divided into ten runs as shown in Table 4.

The metrics used for measuring the performance of IDS are detection rate and accuracy. Table 4 shows the detection

**Table 5** Comparison of existing mitigation solutions for evil twin attack

| Method | Detects evil twin attack | Maintaining white-list of authorized AP(s) | Requires timing characteristics | Requires traffic profiling | Requires specialized hardware | Overhead | Remarks |
|---|---|---|---|---|---|---|---|
| Maintaining white-list [20, 29] | N | Y | N | N | N | N | Does not detect evil twin AP |
| Maintaining RTT, IAT [16, 18, 19, 30] | Y | N | Y | Y | Y | High | High requires precise timing characteristics |
| Frame feature extraction [23, 31, 32] | Y | N | Y | Y | Y | High | Require precision in measurements of traffic. Prone to false positives |
| Proposed scheme | Y | N | N | N | N | Low | Easy to deploy. Encryption free. Low resource overhead. No protocol modification required |

rate and accuracy for the proposed IDS. As seen, the detection rate is in the range of 92–100% mark. During our experiments the evil twin terminal is always kept active. So the detection ratio under ideal circumstances should had been 100%. However, it falls below that mark in certain runs. Our scheme works on the assumption of receipt of two association responses in lieu of one association request frame. As wireless is a noisy medium, it is observed that in certain cases the IDS fails to capture one of the two association responses. As a result, it treats the evil twin scenario as normal network scenario (since it receives only one association response frame). On the other hand, the accuracy touches 100% in all the runs. It is because once the IDS receives multiple association responses, presence of evil twin is accurately determined.

Figures 12, 13 and 14 represent the database size growth, memory usage, and CPU usage, respectively of the IDS over a period of 24 h. As seen in Fig. 12, database occupies a maximum space of 2750 KB over 24 h usage. From Fig. 13, we see that the memory utilization of the IDS reaches a peak value of mere 62 MB when it runs non-stop for a period of 24 h. Similar observations can be made for the CPU utilizations. During the four runs of the IDS, we can see the average CPU utilization by the IDS is just 1–2% as seen in Fig. 14. Some spikes can be seen in the CPU utilization graph, but they do not exceed 3% usage. Hence the proposed IDS is CPU friendly. We also observe that the proposed IDS is lightweight and does not consume excessive amounts of system resources. The distribution of the response times in Fig. 15 shows that almost 92% of the responses are received within a time window of 0.05 seconds. So for an attacker to succeed, it needs to capture a frame, note the sequence number and AID (or predict it), craft a frame and send it within the small time window. Given such a small time window, its highly improbable that the attacker would be able to spoof the required parameters.

Table 5 compares the various existing mitigation techniques for the evil twin attack. The other features that we compare are the requirements of maintaining white list of authorized APs, timing based methods, collecting frame features, proprietary hardware and resource overhead. The remarks column of Table 5 describes the pros and cons of the techniques discussed. The pros and cons of the techniques compared here are already discussed in detail in the literature Sect. 2.3.

## 5 Conclusion

In this manuscript, we have proposed an IDS for the evil twin attack. In evil twin attack an attacker creates a fake AP by cloning the legitimate AP in order to hijack client's communication, re-direct clients to malicious websites, steal credentials of the clients connecting to it. Existing schemes for evil twin detection and prevention require protocol alternation, encryption, precise timing synchronization techniques, have maintenance, scalability and compatibility issues. The proposed IDS overcomes the issues associated with the current methods and also provides a high detection rate exceeding 92% mark and 100% accuracy while being light on the system resources as well. We also show that our proposed scheme is robust enough to detect a single evil twin, multiple evil twins for single AP and multiple evil twins for multiple APs. Furthermore, the proposed IDS does not require any sort of injection into network packet making it an efficient scheme for the detection of evil twin attack.

## References

1. Wireless Infrared Communication Systems and Networks, *International Journal of Wireless Information Networks*, Vol. 4, No. 4, pp. 257–258, 1997.
2. J. A. Gutiérrez, On the use of IEEE Std. 802.15.4 to enable wireless sensor networks in building automation, *International Journal of Wireless Information Networks*, Vol. 14, No. 4, pp. 295–301, 2007.
3. P. Johansson, R. Kapoor, M. Kazantzidis and M. Gerla, Personal area networks: Bluetooth or IEEE 802.11?, *International Journal of Wireless Information Networks*, Vol. 9, No. 2, pp. 89–103, 2002.
4. D. C. Cox, Wireless loops: What are they?, *International Journal of Wireless Information Networks*, Vol. 3, No. 3, pp. 125–138, 1996.
5. R. D. Pietro, and G. Oligeri, Silence is golden: exploiting jamming and radio silence to communicate. *ACM Transactions on Information and System Security (TISSEC)*, Vol. 17, No. 3, pp. 9:1–9:24, 2015.
6. Y. Gilad, and A. Herzberg, Off-path tcp injection attacks. *ACM Transactions on Information and System Security (TISSEC)*, Vol. 16, No. 4, pp. 13:1–13:32, 2014.
7. J. Dong, R. Curtmola, and C. Nita-Rotaru, Practical defenses against pollution attacks in wireless network coding. *ACM Transactions on Information and System Security (TISSEC)*, Vol. 14, No. 1, pp. 7:1–7:31, 2011.
8. J. Bellardo, and S. Savage, 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In: *Proceedings of the 12th Conference on USENIX Security Symposium*, Vol 12, SSYM'03, pp. 15–28, 2003.
9. C. Anagnostopoulos, Intelligent contextual information collection in internet of things, *International Journal of Wireless Information Networks*, Vol. 23, No. 1, pp. 28–39, 2016.
10. W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, D. Towsley, and S. Jaiswal, Passive online detection of 802.11 traffic using sequential hypothesis testing with TCP ACK-Pairs. *Mobile Computing, IEEE Transactions*, Vol. 8, No. 3, 398 –412, 2009.
11. L. Ma, A. Y. Teymorian, X. Cheng, and M. Song, RAP: protecting commodity Wi-Fi networks from rogue access points. In: *The Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness Workshops, QSHINE '07*, pp. 21:1–21:7. ACM, New York, NY, USA 2007.

12. AirWave Wireless Management Suite, Whitepaper, Aruba. URL www.moonblinkwifi.com/files/airwave-solution-guide.pdf 2006.
13. P. K. Dubey, and J. N. Verma, *Method And Apparatus For Detecting A Rogue Access Point In A Communication Network*. URL http://www.google.com/patents/US20120124665 2012.
14. R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland, Rogue access point detection using temporal traffic characteristics. In: *IEEE Global Telecommunications Conference, GLOBECOM '04.* , Vol. 4, pp. 2271–2275, Vol.4, 2004.
15. P. Chumchu, T. Saelim, and C. Sriklauy, A new MAC address spoofing detection algorithm using PLCP header. In: *Information Networking (ICOIN), 2011 International Conference*, pp. 48 –53, 2011.
16. H. Han, B. Sheng, C. Tan, Q. Li and S. Lu, A timing-based scheme for rogue AP detection, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 11, pp. 1912–1925, 2011.
17. 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, 2010.
18. C. D. Mano, A. Blaich, Q. Liao, Y. Jiang, D. A. Cieslak, D. C. Salyers, and A. Striegel, RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning. *ACM Transactions on Information Systems Security*, Vol. 11, NO. 2, pp. 2:1–2:23, 2008.
19. Y. Song, C. Yang, and G. Gu, 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* , pp. 323 –332, 2010.
20. 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: *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, pp. 32–36. ACM, New York, USA 2011.
21. V. Sriram, G. Sahoo, and K. Agrawal, Detecting and eliminating rogue access points in IEEE-802.11 WLAN—A multi-agent sourcing methodology. In: *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pp. 256 –260, 2010.
22. M. K. Chirumamilla, Agent based intrusion detection and response system for wireless LANs. In: *Proceedings of IEEE International Conference on Communications*, pp. 492–496, 2003.
23. T. Kohno, A. Broido, and K. Claffy, Remote physical device fingerprinting. In: *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, SP '05, pp. 211–225, 2005.
24. S. Bratus, C. Arackaparambil, A. Shubina, and D. College, *Detection of Rogue APs Using Clock Skews: Does it Really Work?* . URL www.cs.dartmouth.edu/~cja/papers/toorcon11_ver6a.pdf 2009.
25. IEEE Standard for Information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999) pp. C1–1184, 2007.
26. N. Agrawal and S. Tapaswi, The performance analysis of honeypot based intrusion detection system for wireless network, *International Journal of Wireless Information Networks*, Vol. 24, No. 1, pp. 14–26, 2017.
27. K. Chae, J. Shao, S. Jung, C. Han, S. Bae, and I. Jeong, A scheme of detection and prevention rogue AP using comparison security condition of AP. *UACEE International Journal of Advances in Computer Networks and Its Security*, 2012.
28. Aircrack-ng Suite. URL http://www.aircrack-ng.org/ 2015.
29. R. Whelan, L. Van Wagenen, and R. Morris, *System and method for detecting unauthorized wireless access points*, US Patent 8,787,576 2014.
30. A. Aggarwal, E. Hardie, S. Das, R. Gupta, and A. Naguib, *Detection of falsified wireless access points*, US Patent 8,750,267 2014.
31. B. Alotaibi, and K. Elleithy, An empirical fingerprint framework to detect rogue access points. In: *Systems, Applications and Technology Conference (LISAT), 2015 IEEE Long Island*, pp. 1–7, 2015.
32. J. Yu, Applying TCP profiling to detect wireless rogue access point. In: *Proceedings of the International Conference on Wireless Networks (ICWN). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, pp. 1–7, 2014.

**Mayank Agarwal** is a post-doctoral student in Ben-Gurion University of the Negev, Israel. He has completed his Ph.D. from IIT Guwahati in 2017. His research interest covers developing intrusion detection frameworks for wired, ad-hoc network and wireless sensor networks.

**Santosh Biswas** received B.E. degree from NIT, Durgapur, India, in 2001. He completed his M.S. and Ph.D. from IIT Kharagpur, India, in the year of 2004 and 2008, respectively. He works as an Associate Professor at the Department of Computer Science and Engineering, IIT Guwahati. His research interests include network security, VLSI testing and discrete event systems.

**Sukumar Nandi** is senior member of IEEE and is in Department of Computer Science and Engineering at the Indian Institute of Technology Guwahati. Professor Nandi did his Ph.D. from IIT Kharagpur under Professor P. Pal Chaudhri. Following that he joined IIT Guwahati, and has been teaching there since 1995.