IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 January 2024, accepted 27 January 2024, date of publication 6 February 2024, date of current version 16 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362803

RESEARCH ARTICLE

A Signature-Based Wireless Intrusion Detection System Framework for Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks

MANESH THANKAPPAN[1,2], (Member, IEEE), HELENA RIFÀ-POUS[1,3], (Member, IEEE), AND CARLES GARRIGUES[1,3]

[1]Estudis d'Informàtica Multimèdia i Telecomunicació, Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), 08018 Barcelona, Spain
[2]Adi Shankara Institute of Engineering and Technology, Kalady, Ernakulam, Kerala 683574, India
[3]CYBERCAT-Center for Cybersecurity Research of Catalunya, 43003 Tarragona, Spain

Corresponding author: Manesh Thankappan (mthankappan@uoc.edu)

This work was supported in part by the Ministerio de Ciencia e Innovación, la Agencia Estatal de Investigación and the European Regional Development Fund (ERDF) under Project PID2021-125962OB-C31; and in part by the ARTEMISA International Chair of Cybersecurity and the DANGER Strategic Project of Cybersecurity, both funded by the Spanish National Institute of Cybersecurity through the European Union-NextGenerationEU and the Recovery, Transformation and Resilience Plan.

ABSTRACT One of the advanced Man-in-the-Middle (MitM) attacks is the Multi-Channel MitM (MC-MitM) attack, which is capable of manipulating encrypted wireless frames between clients and the Access Point (AP) in a Wireless LAN (WLAN). MC-MitM attacks are possible on any client no matter how the client authenticates with the AP. Key reinstallation attacks (KRACK) in 2017-18, and the latest FragAttacks in 2021 are frontline MC-MitM attacks that widely impacted millions of Wi-Fi systems, especially those with Internet of Things (IoT) devices. Although there are security patches against some attacks, they are not applicable to every Wi-Fi or IoT device. In addition, existing defense mechanisms to combat MC-MitM attacks are not feasible for two reasons: they either require severe firmware modifications on all the devices in a system, or they require the use of several advanced hardware and software for deployment. On top of that, high technical overhead is imposed on users in terms of network setup and maintenance. This paper presents the first plug-and-play system to detect MC-MitM attacks. Our solution is a lightweight, signature-based, and centralized online passive intrusion detection system that can be easily integrated into Wi-Fi-based IoT environments without modifying any network settings or existing devices. The evaluation results show that our proposed framework can detect MC-MitM attacks with a maximum detection time of 60 seconds and a minimum TPR (true positive rate) of 90% by short-distance detectors and 84% by long-distance detectors in real Wi-Fi or IoT environments.

INDEX TERMS Attack signature, FragAttacks, intrusion detection, Internet of Things (IoT), KRACK, multi-channel MitM (MC-MitM), Wi-Fi, WPA, WLAN.

I. INTRODUCTION

A. CONTEXT

WLANs are susceptible to a wide array of wireless security attacks. A Man-in-the-middle (MitM) attack is a critical security threat towards wireless networks in which the perpetrator is positioned in the middle of the communication between the client and the Access Point (AP), allowing the attacker to eavesdrop, manipulate messages, and impersonate one of the parties. In the simplest form of such attacks, the attacker introduces a laptop with two Wi-Fi cards; one of them is connected to the legitimate AP or his own AP, and the other acts as a

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

rogue AP (also known as an evil-twin), spoofing the target AP so that clients will connect to it because of the commonly used automatic AP selection option [1]. In general, there are two approaches to perform MitM attacks in a WLAN.

In the first approach, which we will refer to as a traditional rogue AP MitM attack from now on, the attacker launches a new rogue AP, forces the clients to connect to it using a known Wi-Fi passphrase, and then manipulates the encrypted traffic. Therefore, traditional rogue AP MitM attacks require a known Wi-Fi passphrase for manipulating encrypted traffic between the client and the AP. Fluxion [2], Wifiphisher [3], WiFi-Pumpkin [4], airbase-ng [5], etc. are some commonly employed traditional rogue AP MitM attack tools. The second approach, our research focus, is the Multi-Channel MitM (MC-MitM) attack, introduced by Vanhoef and Piessens in 2014 [6], which consists of two Wi-Fi cards operating on two different channels but maintaining a single connection to manipulate encrypted wireless traffic between the client and the legitimate AP on the fly without possessing any legitimate Wi-Fi passphrases.

The rationale behind the MC-MitM attack is to clone the legitimate AP on a different channel, which facilitates the attacker exchanging all connection establishment and data frames between both channels so that he can communicate with both the client and the AP simultaneously [6], [7]. Moreover, exchanging frames between different channels is possible no matter how the client authenticates with the network. Therefore, MC-MitM attacks can be used in personal as well as enterprise Wi-Fi networks. Once the MC-MitM position is acquired, the attacker can use other attacks to block and modify encrypted frames between the client and legitimate AP. We note that the MC-MitM position does not break any encryption but is primarily used to perform attacks to exploit specific weaknesses (e.g., flaws in authentication or encryption) in Wi-Fi standards such as WPA, WPA2, or WPA3. A comprehensive security analysis of different Wi-Fi standards is available in our previous paper [13]. Fundamentally, to acquire the MC-MitM position, the attacker either employs special jamming techniques or channel switch announcements (CSAs) to force the clients to switch to their channels. In this paper, we refer to jamming-based MC-MitM as base variant attacks and CSA-based MC-MitM as improved variant attacks.

The most well-known MC-MitM base variant attack is the key reinstallation attack (KRACK). KRACK exploits severe nonce reuse vulnerabilities (discovered by Vanhoef et al. in October 2017 [8]) during 4-way handshake mechanisms in the IEEE 802.11 standards. Such vulnerabilities enable the attacker to trivially decrypt Wi-Fi frames, especially from Linux and Android devices supporting WPA/2 standards. This was the first non-vendor-specific vulnerability that impacted millions of Wi-Fi devices due to a faulty implementation of the standard.

Regarding the MC-MitM improved variants, the most significant attacks include FragAttacks and some extended versions of KRACK attacks. The FragAttack is the latest non-vendor-specific attack using the MC-MitM position (discovered by Vanhoef in May 2021 [9]). It exploits a set of authentication weaknesses in the fragmentation and aggregation features of IEEE 802.11 standards allowing the attackers to inject packets into encrypted Wi-Fi networks and obtain sensitive client data.

The aforementioned MC-MitM attacks also affect WPA3 standards. Although patches are available for both KRACK and FragAttacks, the critical problem is that they are not applicable on every Wi-Fi or IoT device because of factors like resource constraints, deprecated security protocols, expired product support periods, etc. Four years after KRACK first appeared, it is estimated that more than 75 per cent of Wi-Fi enabled devices still remain vulnerable to it [10], [11].

MC-MitM attacks have been exploited in some critical systems. For example, [12] showed how the MC-MitM position could be applied to obfuscate train control systems to cause emergency braking and system collapse. Surprisingly, they used the MC-MitM position to capture, decrypt, and modify protected Wi-Fi packets (train control messages). In our previous paper [13], we evaluated the capabilities of MC-MitM attacks and provided a detailed description of the different kinds of MC-MitM attacks reported so far.

B. CHALLENGES IN DETECTING MC-MITM ATTACKS

Detecting MC-MitM attacks is challenging because the attacker spoofs almost every characteristic of the legitimate AP and the client (victim) simultaneously, and operates as legitimately as possible in the target Wi-Fi network. More specifically, the attacker does not conduct any flooding attack using spoofed beacons, probe requests, or other frames to deceive and acquire the clients. Therefore, the frame arrival rate-based detection technique is also not helpful. In MC-MitM attacks, the attacker collects the beacons of real AP and retransmits them on his rogue channel. As a result, MC-MitM attackers can evade snooping-based rogue AP detection techniques, such as [14] and [15] which are based on verifying whether RSSI values are higher than that of the legitimate AP. Moreover, the MC-MitM attacker can easily configure the transmission power and forge other features if he knows them [16]. Furthermore, researchers show the feasibility of using CSAs for launching MitM attacks even with relatively lower RSSI values than that of legitimate APs [17]. Therefore, relying on RSSI values alone may not be an effective defense.

Communication channels can also be monitored. However, checking beacons only on the legitimate channel is not always beneficial because there are valid reasons for an AP to switch to different channels. For example, channel switching is essential to avoid interference from radar noise on certain channels, and is a dynamic action in modern routers enabled by the Dynamic Frequency Selection (DFS) feature [18]. Furthermore, the MC-MitM attacker can use a special kind of constant jamming or reactive jamming by using cheap off-the-shelf Wi-Fi dongles in order to establish

the MitM position, which is relatively hard to detect by existing intrusion detection systems [6], [19]. This is because the MC-MitM attack transmits random noise pulses during jamming, which are interpreted as any non-Wi-Fi device using a similar frequency band.

Traditional perimeter security measures (e.g., firewall, VPN) are generally employed to protect sensitive communications in a WLAN. However, such measures cannot prevent MC-MitM attacks from directly attacking various Wi-Fi devices since such attacks are link-layer attacks, and firewalls deal with upper layers stack.

The Wi-Fi Alliance enforced Protected Management Frame (PMF) beginning in 2018, which provides integrity protection mechanisms for WPA2 and WPA3 protocols to prevent rogue AP MitM or DoS attacks [20], [21]. The use of PMF only achieves protection for certain robust management frames such as deauthentication, disassociation, and action frames [22]. However, PMF is not sufficient to defend against MC-MitM attacks. This is mainly because: 1) the attacker does not use deauthentication packets to acquire the MC-MitM position [23]; 2) PMF cannot detect jamming attacks [24]; and 3) MC-MitM attacks use beacons or probe responses, which PMF does not protect. Moreover, if the MC-MitM attacker is an insider (authorized user), he can even steer clients to switch to his rogue AP using CSA action frames [17], [25]. This makes such attacks difficult to detect in practice.

In the aforementioned scenarios, it is difficult to appropriately identify MC-MitM attacks. Although specific defense mechanisms have been proposed in the literature, they require modifications to the Wi-Fi protocol or advanced hardware or software to be deployed on each Wi-Fi client and/or AP and are therefore only effective if all devices on a WLAN are compatible with them. This stringent security requirement is not always achievable with IoT devices or every Wi-Fi client.

C. MOTIVATION

In our previous paper [13], we extensively studied the technical feasibility of various MC-MitM defense mechanisms and demonstrated that their deployment is difficult to achieve, especially in IoT environments such as smart homes. On the one hand, there are no patches for all commercial devices, and on the other hand, the management and maintenance of these devices requires technical knowledge that the average user does not have. Moreover, existing defense mechanisms cannot handle such attacks due to several interoperability issues. Hence, there is a need for effective defense mechanisms. Given these considerations, we have designed a lightweight and signature-based intrusion detection framework that is tailored to meet the demands of smart environments based on IoT. Rather than depending on machine learning, our detection framework scrutinizes wireless network frames to quickly recognize attack signatures or behaviors of malicious network activity. Our approach is a plug-and-play system that can be easily integrated into any Wi-Fi or IoT setup without requiring changes to network configurations or pre-existing devices, and it delivers consistent security against all types of MC-MitM attacks.

In real Wi-Fi or IoT environments, our short-distance detectors achieved a minimum True Positive Rate (TPR) of 90%, while our long-distance detectors achieved a TPR of 84%. Furthermore, we have evaluated our proposed framework using the AWID3 dataset [26], which is a publicly available dataset containing KRACK attack signatures. Our framework showed good performance (above 99% in accuracy) compared to other mechanisms that utilize the AWID3 dataset.

D. CONTRIBUTIONS

In this paper, we make the following contributions:

1) Classification and analysis of attack traffic in MC-MitM attacks.
2) Theoretical and empirical analysis of attack traffic and creation of potential attack signatures for MC-MitM attacks.
3) Design of the first plug-and-play signature-based wireless intrusion detection system framework that can be used in any Wi-Fi network.
4) Development of an open-source prototype [27] of the proposed framework using the python-scapy library.
5) Empirical evaluation of the proposed framework in an industry-relevant smart home environment with off-the-shelf IoT devices.

E. ORGANIZATION OF THE PAPER

The remainder of the paper is organized as follows: Section II briefly discusses the background and related work; Section III classifies the specific attack traffic during MC-MitM attacks and presents their behavior; Section IV presents an in-depth combination of theoretical and empirical analysis of attack traffic, creates attack signatures, and indicates metrics to identify MC-MitM attacks; Section V introduces our proposed solution and architectural units; Section VI presents an evaluation of the proposed solution. Finally, Section VII presents conclusions and future research work.

II. BACKGROUND AND RELATED WORK

We first outline the working principles of the MC-MitM attack and its variants. We then classify and describe existing defense mechanisms for MC-MitM attacks.

A. BACKGROUND

MC-MitM attacks can sniff and manipulate encrypted wireless communication (e.g., WPA, WPA2, or WPA3) between clients and the AP in a WLAN. In such attacks, the attacker's goal is to identify the channel of the legitimate AP and then clone it on a different channel to exchange frames between both channels. The said exchange of frames enables the attacker to legitimately communicate with both end devices (the client and legitimate AP) simultaneously. Once the

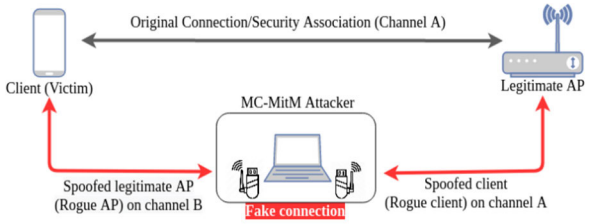


FIGURE 1. Multi-channel MitM Setup.

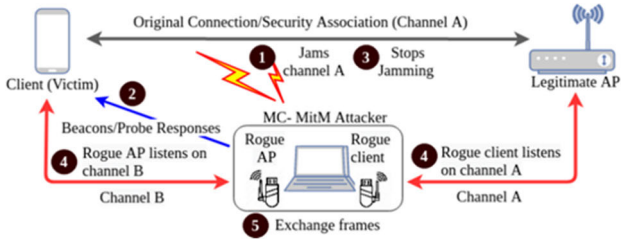


FIGURE 2. MC-MitM-BV attack.

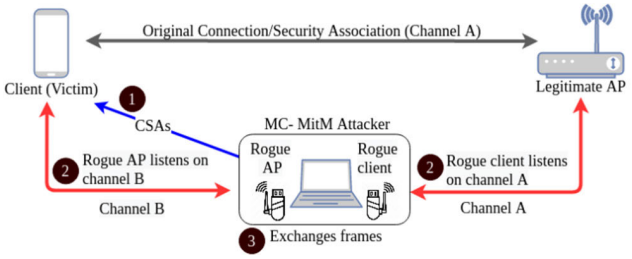


FIGURE 3. MC-MitM-IV attack.

MC-MitM position is acquired, the attacker can manipulate (e.g., delay, modify, inject, replay) encrypted frames between end devices. Figure 1 shows a typical MC-MitM attack setup. As we can see from Figure 1, the attacker uses two Wi-Fi dongles to spoof end devices on the opposite-side channel. Since both Wi-Fi dongles are physically close, they receive each other's frames even if they operate on two different channels.

The main advantage of employing the MC-MitM attack is that it does not require the legitimate Wi-Fi passphrase of a WLAN since the attacker does not break the original connection or security association between end devices. Thus, end devices retain a PMK (Pre-Master Key) stored in their Wi-Fi chips and use it for negotiating the same session key or PTK (Pairwise Transient Key) through a fake connection as shown in Figure 1. More specifically, the attacker exchanges authentication, association, and 4-way handshake frames between these two channels, which actually makes the end devices negotiate the same session key to encrypt the subsequent communication. This enables the MC-MitM attacker to bypass the authentication and 4-way handshake between the AP and the victim, capturing encrypted frames that can be manipulated by applying potential key reinstallation, aggregation, and fragmentation vulnerabilities.

In terms of forcing the clients towards the attacker, we classify MC-MitM attacks in two classes: base variant and improved variant.

1) BASE VARIANT

Vanhoef and Piessens introduced the MC-MitM base variant (MC-MitM-BV) attack in 2014 [6].

As shown in Figure 2, with this attack variant, the attacker: (1) jams the operating channel (channel A) of the legitimate AP (2) broadcasts beacons or probe responses (already collected from the legitimate channel A) instantly on the rogue channel (channel B) to force the client into connecting to his rogue AP (3) stops the jamming as soon as the client gets connected to the rogue AP (4) listens on channels B and A, respectively by the rogue AP and rogue client and (5) begins exchanging encrypted frames between the legitimate AP and client and vice versa.

Basically, two types of jamming techniques are used with this variant: constant jamming and reactive jamming. In this paper, we call MC-MitM-BV with constant jamming as MC-MitM-BVC and MC-MitM-BV with reactive jamming as MC-MitM-BVR attacks. When constant jamming is used, all the traffic on a target channel will be indiscriminately jammed while only specific frames (beacons or probe responses) are malformed with the reactive jamming. We highlight that jamming does not break the original security association. Instead, it just makes target networks unavailable for some time. As per the 802.11 standards, a client always chooses an available network or a network to which it was previously connected. Therefore, the victim switches with the available rogue AP by switching its channel and transmitting data on it. Some examples of MC-MitM enabled security downgrade attacks, KRACK, and DoS attacks performed using MC-MitM-BVC are described in the literature [6], [8], [28], [29], whereas in [19], MC-MitM-BVR is used.

2) IMPROVED VARIANT

Vanhoef and Piessens further proposed the MC-MitM improved variant (MC-MitM-IV) attack in 2018 [7], which is more practical compared to the MC-MitM-BV attack. As shown in Figure 3, with this attack variant, the attacker: (1) sends forged channel switch announcements (CSA) on channel B to force the clients into connecting to the rogue AP (2) listens on channels B and A, respectively by the rogue AP and rogue client and (3) begins exchanging encrypted frames between the legitimate AP and client and vice versa. The use of CSA significantly reduces the cost of jamming and the attacker's effort. Moreover, the attack requires only a few CSAs.

The use of CSAs is more reliable as it is an activity of the APs under radar noise conditions that the clients cannot decline. Similar to jamming, CSAs do not break the original security association. Instead, they instruct the client to switch to a new channel designated by the attacker. In addition, the attacker can transmit CSAs by forging a CSA information ele-

ment inside beacon frames, probe response frames, or action frames. Some prominent examples of MC-MitM-IV attacks, including KRACK, DoS, and the latest FragAttacks, have appeared in the literature [7], [9], [12], [30]. In Section III of our previous paper [13], we thoroughly explained the technical setup, inner workings, and extensive evaluation of various MC-MitM attacks that manipulate victim's data frames, resulting in frame decryption and potential extraction of sensitive data.

3) OTHER SPECIAL CAPABILITIES OF MC-MITM ATTACKS

The MC-MitM attacker behaves as normal in a WLAN and does not conduct any flooding attack using spoofed deauthentications, beacons, probe requests, or other frames to deceive and acquire the clients. Attackers can also circumvent IDS alerts with the special jamming methods employed in MC-MitM since they transmit noise pulses instead of injecting wireless frames [6]. Moreover, both attack variants can be effectively used against PMF-enabled devices. This is because management frames such as beacons or probe responses are not protected even if PMF is enabled. This ability enables the MC-MitM attacker to target the latest WPA2 and WPA3 devices as they use PMF by default [20]. Furthermore, the attacker can send CSA through action frames if he is an insider attacker, even when PMF is used [17]. It is also feasible to employ CSAs to acquire the MitM position from relatively longer distances with weaker signals [17]. Furthermore, the MC-MitM position facilitates the viability of certain MitM attacks such as chop-chop attacks [31], SSLStrip attacks [32], and Wi-Fi geolocation attacks [33], etc.

B. RELATED WORK ON DEFENSE MECHANISMS

We categorize the current defense mechanisms against MC-MitM attacks into two groups: stage 1 and stage 2 defense mechanisms. Stage 1 mechanisms aim to protect against attackers prior to obtaining the MC-MitM position by identifying genuine attack vectors, including rogue channels, rogue devices, or spoofed channel switch announcements. The second category concentrates on defending against MC-MitM enabled attacks (such as KRACK, cipher downgrades, and FragAttacks) after the attacker has gained control of the MC-MitM position.

1) STAGE 1 DEFENSE MECHANISMS

The authors of [34] introduced an Operating Channel Validation (OCV) technique to cryptographically validate the operating channel between two wireless stations. This technique proposes the utilization of a new Operating Channel Information (OCI) element as an extension to the 802.11 standards. During the 4-way handshake messages, the OCI element in EAPOL (Extensible Authentication Protocol over LAN) frames is authenticated to ensure that the sender and the receiver are using the same communication channels. Although the OCV has been ratified as a feature in IEEE standards, it is not compulsory in any of the WPA standards and has not yet been widely adopted in practical settings or implemented by device vendors. Furthermore, the OCV technique solely provides protection for PMF capable devices, as it requires the use of PMF to prevent unprotected channel switch announcements.

In another work, [16] proposed a beacon protection mechanism to defend against attacks that exploit unprotected beacons to prevent common rogue AP-based attacks and potential MitM attacks. They introduced an additional information element (IE) within each beacon, enabling clients to cryptographically verify the integrity of beacons when connecting to an AP. Similar to their previous defense mechanism [34], the beacon protection mechanism encounters practical challenges primary due to the requirement of PMF, which can create several interoperability issues while using devices supporting only WPA or WPA2 devices. Furthermore, the proposed mechanism does not block possible MC-MitM insider attacks, as demonstrated in [12].

In the WPA3-2020 updates, the WFA included another feature called Simultaneous Authentication of Equals-Public Key (SAE-PK) [35] to uniquely identify APs in a WLAN during the connection establishment process based on ECC (Elliptic Curve Cryptography) public key cryptography. SAE-PK also prevents insider attackers from setting up rogue AP and performing MitM attacks by using the using the AP's public key's digital signature. However, the detection of rogue APs is limited to the SAE-PK authentication phase or when the client initially connects to the AP. In contrast, an MC-MitM attacker typically positions themselves between an already connected client and the AP. The attacker can also bypass the SAE authentication because, according to [36], the WPA3 client uses an open authentication instead of an SAE authentication while reconnecting to an already connected network.

In [37], the authors proposed a defense mechanism based on Physically Unclonable Functions (PUF) to prevent rogue AP's actions during the MC-MitM attacks. Their approach involved generating a unique secret key from the AP's PUF signature and using it for mutual authentication between the AP and client devices. However, the PUF-based technique requires complex hardware modifications on all devices within a WLAN. Additionally, this method is vulnerable to certain types of MitM attacks [38].

In [19], the authors presented a defense method for Wi-Fi clients to detect rogue AP actions during MC-MitM-BVR attacks. They developed a patch for wpa_supplicant, an open-source implementation of Wi-Fi clients, to verify the uniqueness of a pair of identities such as SSID (Service Set Identifier) and BSSID (Basic Service Set Identifier) or when a client initiates a connection with an AP. However, the detection becomes challenging if the MC-MitM attacker employs continuous jamming on the legitimate AP channel, preventing the target client from retrieving and comparing the required beacon information. Moreover, relying solely on the uniqueness of the SSID and BSSID pair is not entirely

effective due to situations where the same pair of identities may be used. For example, if the AP supports a dual-band connection, there may be beacons with the same pair of identities.

2) STAGE 2 DEFENSE MECHANISMS

Most of the stage 2 defense mechanisms are intended to detect and mitigate MC-MitM enabled KRACK attacks due to their severe security impact on Wi-Fi systems. Various defense mechanisms such as [39], [40], [41], and [42] perform network analysis to identify a retransmitted or duplicated message 3 of the 4-way handshake mechanism during KRACK attacks. Nonetheless, as per the 802.11 standards, it is considered reasonable for an Access Point (AP) to retransmit message 3 in specific situations. This can occur when there is network traffic congestion or until the AP reaches its maximum retransmission limit. Consequently, indiscriminately blocking all retransmitted handshake message may lead to frequent handshake failures or increased false-positive rates.

In a recent study [43], an anomaly detection technique was proposed to identify handshake messages across multiple channels using supervised machine learning models, specifically targeting the detection of KRACK behavior. They used a state machine grouping algorithm to group retransmitted message 3 of the 4-way handshake on any channel other than the legitimate one. However, their focus was solely on detecting KRACK attacks. Similar works include [44], [45], and [46]. It is important to note that these machines learning based defense mechanisms have not been evaluated in real networks but rather assessed using the publicly available AWID3 dataset [26].

On the other hand, mechanisms described in [47], [48], and [49] propose new cryptographic verification techniques during the exchange of 4-way handshake messages to avoid nonce reuse weaknesses exploited by KRACK. These mechanisms also provide defense against cipher suite downgrade attacks on APs. However, the implementation of these proposals requires several changes to IEEE standards and has not been tested in real-world attack scenarios.

In [50], Snort rules are provided to detect network packets containing specific content (e.g., Dot11, RadioTap, FCfield) that may occur during the execution of KRACK attack tools or scripts. However, different implementations of the same KRACK attacks might not be detected by the current Snort rules. The content used by Snort rules to detect or match KRACK packets may even be present in legitimate WLAN packets or scripts of other tools and attacks developed using Scapy. Hence, relying solely on Snort with specific rules may prove ineffective or result in false alarms.

In order to protect against FragAttacks, there are currently no dedicated defense mechanisms available. However, there is a testing framework [51] for identifying fragmentation and aggregation vulnerabilities in Wi-Fi devices.

In general, the current defense mechanisms lack a comprehensive approach that can effectively detect all types of MC-MitM attacks. Additionally, a majority of these mechanisms have not undergone real-world evaluation in Wi-Fi or IoT environments, limiting their practical applicability. In Section VI-E, we present a comparison of the existing defense mechanisms, while in Section VI-F, we analyse the performance of systems that rely on the AWID3 dataset for evaluation purposes.

III. MULTI-CHANNEL MITM ATTACK ANALYSIS

In this section, we analyze the specific attack traffic related to different MC-MitM attack variants (see Section II-A). Towards this, we first classify MC-MitM attack traffic and then investigate the behavior of different attack variants.

Based on the behavior of MC-MitM attacks, we classify them into stage 1 attack traffic and stage 2 attack traffic. Stage 1 attack traffic appears first and indicates specific traffic during the acquisition of the MC-MitM position in Wi-Fi networks. Stage 1 attack traffic of MC-MitM-BVC and MC-MitM-BVR, respectively, can be the behavior of the network due to constant jamming and reactive jamming attacks; in the case of MC-MitM-IV, stage 1 traffic is the fake CSAs. Soon after the stage 1 attack traffic, i.e., after acquiring the MC-MitM position, stage 2 attack traffic arrives, which shows the behavior of the network when the attacker establishes two fake connections and exchanges authentication, association, 4-way handshake frames, and data frames between the client and the legitimate AP. Both MC-MitM attack variants exhibit similar stage 2 attack traffic.

A. ANALYSIS OF STAGE 1 ATTACK TRAFFIC

This section describes the specific network behavior of stage 1 attack traffic in terms of constant jamming, reactive jamming, and CSA attacks.

1) CONSTANT JAMMING ATTACK

When the attacker initiates a constant jamming attack targeting the operating channel of the AP, all traffic on that channel will be jammed indiscriminately. This means that there will be no Wi-Fi frames on a particular channel until the constant jamming stops. In particular, the MC-MitM attacker usually employs a specific type of constant jamming by transmitting noise pulses for a specific period of time. For instance, the attacker uses a jammer firmware with its Carrier Sense Multiple Access (CSMA) mechanism disabled, so that it injects random energy pulses to make the target channel appear to be always busy. As a result, nearby transmitters (APs) operating on the targeted channels would not send Wi-Fi frames, or clients would remain idle until the jamming on the AP's channel ends. This helps the MC-MitM attacker to force the clients to connect to the same or a cloned network, but on a different channel. The main advantage of this type of constant jamming attack is that it cannot be detected by intrusion detection systems. This is because instead of injecting random Wi-Fi frames, the tool transmits random noise pulses that would be seen as coming from any non-Wi-Fi device using a similar frequency band [6].

2) REACTIVE JAMMING ATTACK

The reactive jamming attack aims to jam beacons and probe responses from a target or AP's channel. The attacker first identifies the frames based on the MAC address and decodes the header on the fly while blocking the reception of the frames by the clients or victims. This is achieved by injecting dummy frames transmitted at higher data rates that resemble the original frames. This injection of dummy frames induces a collision and interference with the targeted beacons or probe responses. Subsequently, the FCS (Frame Check Sequence) of the targeted frame becomes incorrect or malformed, causing clients to ignore it or lose their connection to the AP. As a result, clients choose to connect to the cloned network of the MC-MitM attacker operating on a different channel. Like the constant jamming attack, the reactive jamming attack is also relatively difficult for intrusion detection systems to detect [19].

3) CHANNEL SWITCH ANNOUNCEMENT ATTACK

According to the IEEE standards [52], the channel switch announcement (CSA) is a normal behavior of an AP operating in the 5 GHz frequency bands with dynamic frequency selection (DFS) feature enabled. Typically, CSAs arrive with beacons or probe responses when the AP changes its channel due to the reception of radar pulses after booting up. The DFS feature allows the AP to use specific 5 GHz channels reserved for certain high-priority radar signals used for airport, military, satellite communications and meteorological purposes [18], [33], [53]. When the AP detects any of the high-priority radar signals mentioned above, it sends a CSA to all of its associated clients in order to switch to another 5 GHz channel. Further regulatory specifications for channel selection and DFS features can be found in [53].

CSAs can be easily spoofed regardless of the 2.4 or 5 GHz frequency band, due to the lack of appropriate authentication mechanisms for beacons and probe responses [34]. In either case, all Wi-Fi clients honor such CSAs by immediately switching channels. This allows the MC-MitM attacker to force channel switching using fake CSAs. To send fake CSAs, MC-MitM attackers first collect beacons and probe response frames from the legitimate AP and modify the spoofed CSA information element in them before transmitting them towards the targeted clients. With CSAs, the AP does not immediately switch to a new channel. Instead, it sends a certain number of beacons (the default is 4 CSA beacons as per the IEEE 802.11h standard) containing the CSA before switching to the new channel [53]. However, CSAs under the following three scenarios can be considered fake CSAs.

Scenario 1: The CSAs present in 2.4 GHz Wi-Fi networks must be considered fake CSAs as DFS does not apply to such Wi-Fi networks. This is critical because many home networks operate in the 2.4 GHz band, especially IoT devices.

Scenario 2: In 5 GHz Wi-Fi networks with DFS disabled, no CSAs can occur and those that do should be considered fake.

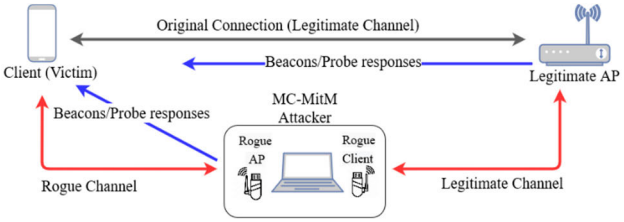


FIGURE 4. Concurrent beacon and probe response traffic.

Scenario 3: When the AP operates in the 5 GHz band and is booted (enabled) with the DFS feature, it first scans for radar signals as part of the Channel Availability Check (CAC) mechanism. Additionally, the AP continuously monitors the operating channel for radar signals throughout its lifetime and only switches to available DFS channels if such signals are detected [53]. DFS can be useful in home networks merely to find the best 5 GHz channel while powering up the AP. On the other hand, DFS is usually advisable for outdoor Wi-Fi devices or networks near airports, weather stations or military radars. Although this scenario is genuine for CSA occurrence when radar signals are detected, such signals are unusual events in home networks. Hence, the occurrence of such CSAs can be considered a warning sign of fake CSAs.

B. ANALYSIS OF STAGE 1 ATTACK TRAFFIC

This section describes the different network behaviors of stage 2 attack traffic.

1) CONCURRENT BEACON AND PROBE RESPONSE TRAFFIC

Concurrent beacon or probe response traffic corresponds to specific traffic that occurs simultaneously on two different channels (belonging to the same frequency band, 2.4 or 5 GHz) with the same SSID and BSSID and other parameters. Figure 4 shows the scenario of concurrent beacon or probe response traffic (blue colored arrows) arrival in a WLAN.

In WLANs, each AP transmits beacons periodically with an interval of 102.4ms. Beacons are essential to announce the presence of a network that synchronizes connected clients. Accordingly, soon after the stage 1 attack, the MC-MitM attacker copies the beacons from the operating channel of the legitimate AP and retransmits them on the rogue channel using his rogue AP. On the other hand, the legitimate AP continues transmitting beacons on its operating channel. This scenario results in the presence of concurrent beacon frames on two different channels immediately after the stage 1 attack.

Similarly, when a Wi-Fi client comes into proximity with previously connected networks in the Preferred Network List[1] (PNL), it starts scanning by sending probe requests to check for available Wi-Fi networks. The PNL, residing in the device's Wi-Fi chip, holds SSIDs and necessary connection details. In response, APs within the network send unicast

[1]PNL is stored in the device's Wi-Fi chip. It is a data structure with the list of SSIDs and any necessary credentials (passwords) for connecting.

probe responses, addressing the client's MAC, and relay information like SSID, BSSID on its operating channel. In the event of jamming or channel switching, clients in a particular network lose connection with the legitimate AP. As a result, the client broadcasts probe requests towards the visible rogue AP, resulting in the arrival of probe response frames to the clients on the rogue channel. On the other hand, the legitimate AP continues to send genuine probe responses to its clients on its operating channel. This scenario results in the presence of concurrent probe response frames on two different channels with the same SSID and BSSID during MC-MitM attacks.

However, such concurrent traffic is infeasible in Wi-Fi networks. The reason is that wireless networks operate on a single channel throughout their uptime or use a single channel to communicate with clients. Thus, the occurrence of such concurrent traffic can be considered an attack.

2) CONCURRENT CONNECTION ESTABLISHMENT TRAFFIC

In addition to concurrent beacon or probe response traffic, during MC-MitM attacks there will be concurrent connection establishment traffic such as authentication, association, EAPOL message exchanges on two different channels with the same SSID and BSSID. Such concurrent connection establishment traffic is essential to maintain the security association between the client and the legitimate AP, allowing them to negotiate the same session key through the MC-MitM setup (see Section II-A). Figure 5 illustrates the scenario of different types of concurrent connection establishment traffic.

When a Wi-Fi client receives probe responses from a previously connected AP, it establishes a connection with that AP on the designated channel. In the case of MC-MitM attacks, the client connects to the rogue AP (with the same SSID and BSSID of the legitimate AP) by sending an 802.1x open authentication frame on the rogue channel. At this moment, as shown in Figure 5(a), the MC-MitM attacker does the following: (1) captures the authentication request from the rogue channel using the rogue AP, (2) retransmits the captured authentication request on the legitimate channel using the rogue client, (3) captures the subsequent authentication response from the legitimate AP using the rogue client, and (4) retransmits the captured authentication response back to the rogue channel using the rogue AP. These frame exchanges constitute concurrent authentication traffic on two different channels with the same SSID and BSSID in a WLAN.

Similarly, the MC-MitM attacker exchanges association frames between two different channels, resulting in concurrent association traffic (see Figure 5(b)). Following the association traffic, the legitimate AP starts a 4-way handshake connection, consisting of four EAPOL messages. Consequently, the MC-MitM attacker collects each of such EAPOL frames from its originating channel and retransmits them on the other channel. Figure 5(c) shows the concurrent EAPOL traffic.

All combined, the above-discussed frame exchanges induce concurrent connection establishment traffic on two different channels with the same SSID and BSSID during MC-MitM attacks. On the other hand, such traffic is unfeasible in Wi-Fi networks because each Wi-Fi client tries to connect to the AP through a single channel at the same time.

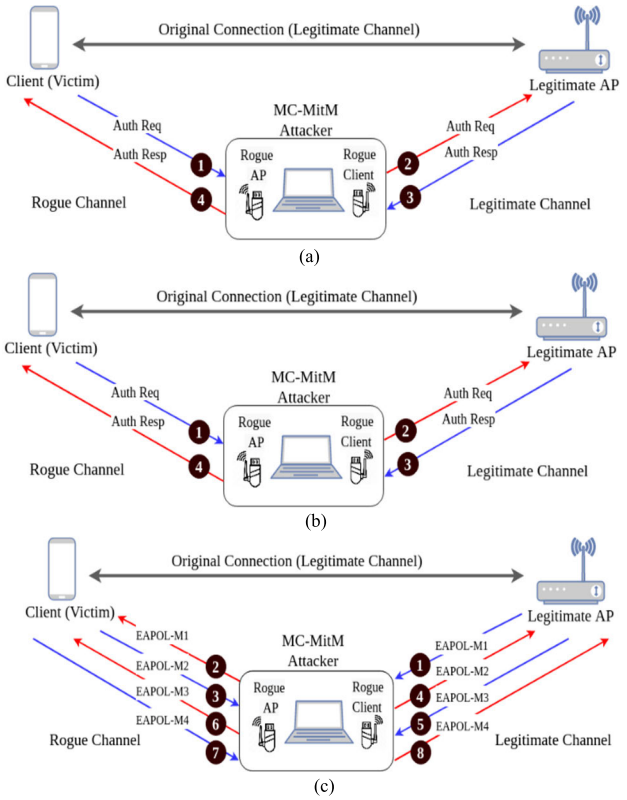


FIGURE 5. Concurrent connection establishment with (a) authentication traffic; (b) association traffic; (c) EAPOL traffic. Blue arrows indicate capturing frames and red arrows indicate retransmitting frames. Numbers on arrows indicate the order of exchange.

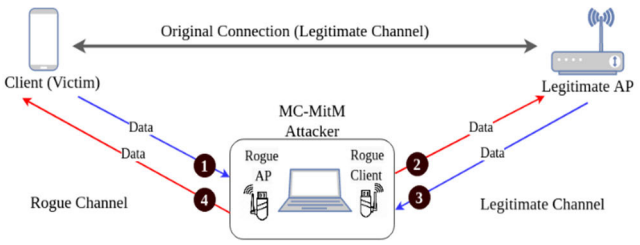


FIGURE 6. Concurrent data traffic. Blue arrows indicate capturing frames and red arrows indicate retransmitting frames. Numbers on arrows indicate the order of exchange.

3) CONCURRENT DATA TRAFFIC

Soon after the connection establishment traffic, both the client and the legitimate AP start communicating by encrypting their data. At this point, as explained in Figure 5, the MC-MitM attacker collects each data frame from its originating channel and retransmits it on the other channel (see Figure 6) to facilitate the communication between the client and the legitimate AP. This results in concurrent data traffic on two different channels with the same SSID and BSSID.

Yet, concurrent data traffic is unfeasible in Wi-Fi networks because the AP only transmits data on its operating channel to communicate with clients in a WLAN.

IV. SIGNATURE CREATION FOR MC-MITM ATTACKS

In this section, we present the stage 1 and stage 2 attack traffic signatures of MC-MitM attacks that we have determined from the network traffic behaviour presented in the previous section. These signatures are based on thresholds that can trigger the detection of these MC-MitM attacks. As we will see, some thresholds are derived from the theoretical analysis of the Wi-Fi protocol, while others are determined using an empirical analysis. We have employed a threshold-based approach in order to passively detect these MC-MitM attacks, since this is cost-effective and faster compared to machine learning-based solutions. Such signatures and their thresholds are used later in Section V, where we present the complete framework for the detection of MC-MitM attacks.

A. REFERENCE SCENARIO AND DETAILS OF EMPIRICAL ANALYSIS

We set up our reference scenario (see Figure 7) in our university research lab. It consists of three Wi-Fi clients (a smartphone and a laptop as WPA2 clients and another laptop as a WPA3 client) and an AP that operates in transition mode to provide WPA2 and WPA3 networks. We implement MC-MitM-BVC and MC-MitM-BVR attacks using the ModWifi platform [54] and MC-MitM-IV attacks using the multi-channel MitM package [55]. We also deploy different MC-MitM attacks interchangeably on WPA2 and WPA3 clients. We acquire the MC-MitM position and capture the traffic between the clients and the AP using Wireshark software.

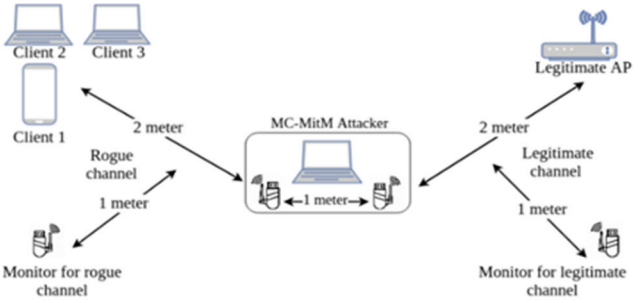


FIGURE 7. Reference attack scenario.

We also capture and thoroughly analyze the benign wireless traffic in different WPA2 or WPA3 based wireless networks, including enterprise (university) networks, home networks, and public networks to study their behavior. In the following sections, we monitor the traffic generated by abovementioned attacks over a specific period of time. From now on, we will refer to this period as the probe interval.

B. DESIGNING SIGNATURES OF STAGE 1 ATTACK TRAFFIC

In this section, we design signatures of the stage 1 attack traffic. Such signatures will be used as warning signs of imminent MC-MitM attacks. To do so, we monitor various types of stage 1 attack traffic (see Section III-A) specifically on the legitimate channel (operating channel) of the AP. This is because MC-MitM attackers first aim to interrupt connection between a victim and an AP on its designated operating channel.

TABLE 1. The resulting FIAT and FDR of Beacons in attack and benign traffic.

	FIAT (ms)		FDR (%)	
	AVG	SD	AVG	SD
Attack traffic	5	1.5	30	5.7
Benign traffic	0.1	0.02	90	1

1) CONSTANT JAMMING ATTACK

A constant jamming attack continuously produces high power noise that represents random bits on the AP' channel. Such attacks also act as intermittent jamming when the attacker stops and restarts MC-MitM attacks, causing sudden drops in frame availability. A drop in the wireless data reception can be detected using packet inter-arrival time (PIAT) and packet delivery ratio (PDR) metrics [56], [57]. In this paper, we refer to the above metrics as frame inter-arrival time (FIAT) and frame delivery ratio (FDR), as we analyze the MAC layer behavior of constant jamming attacks. Further, FIAT can be defined as the time elapsed between the reception of a frame and the next frame, whereas FDR is the ratio of the number of successfully delivered frames to the number of frames transmitted by the AP.

We have taken into account both of these metrics, since they can collectively signify intentional constant jamming activity. In our experiment to study constant jamming attacks, we calculate FIAT and FDR using beacon frames. Theoretically or as per standards [58], a Wi-Fi router typically transmits beacons every 100 milliseconds, resulting in the transmission of 10 beacons per second. In addition, it's crucial that the client successfully receives these beacons with a FIAT of 0.01 milliseconds so as to retain the Wi-Fi connection. Hence, we consider the above values as the foundation for establishing FIAT and FDR thresholds. We prepare the experiment by setting up a wireless connection between a client and an AP. Then, we start a probe interval of 60 seconds in which we first switch on the constant jamming for 30 seconds and monitor the network for 30 more seconds. We repeat the experiment 50 times. For each probe interval, we calculate the FIAT and FDR, and compare their values (average and standard deviation) with benign traffic (no attacks). Table 1 shows the resulting average (AVG) and standard deviation (SD) of FIAT and FDR in attack and benign traffic scenarios from our experiments.

As shown in Table 1, there is a significant variation in the FIAT and FDR values during intentional constant jamming

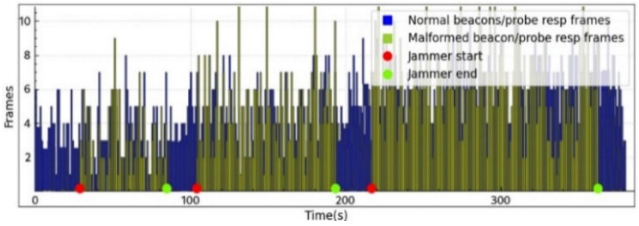


FIGURE 8. Attack traffic during a reactive jamming attack.

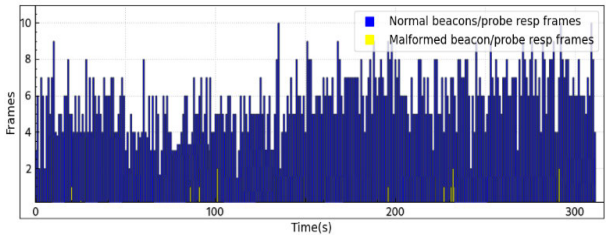


FIGURE 9. Benign traffic with malformed beacon/probe response frames.

attacks compared to the benign traffic. Accordingly, we set a FIAT threshold (TH1) of 2 ms and an FDR threshold (TH2) of 50% to identify the behavior of constant jamming attacks and provide a warning sign of possible MC-MitM-BVC attacks.

2) REACTIVE JAMMING ATTACK

We monitor the behavior of an intentional reactive jamming attacks during a 5-minute probe interval. During this time, we mount 3 periods of reactive jamming for 60, 100, and 150 seconds, as shown in Figure 8. We then separately capture frames during each period and found that more than 90% of the targeted beacons or probe response frames were malformed due to incorrect FCS.

On the other hand, the chances of occurrence of malformed frames in a Wi-Fi network can vary depending on various factors. These factors may include network conditions, the quality of hardware and software, interference, frame aggregation, and the presence of malicious actors. Theoretically, malformed frames should be non-existent and, in a well-maintained and secure network, the chances of malformed frames should be minimal [58].

As shown in Figure 9, when we analyse the benign traffic from the same AP for a probe interval of 15 minutes, we found only a negligible amount (less than 0.8%) of malformed beacon or probe response frames (with incorrect FCS). Such malformed frames are mainly due to incorrect frame reassembly or wrong frame size, which are common phenomena in wireless networks.

On the contrary, specific traffic consisting of malformed beacons or probe responses at higher rates (above 50%), especially on the operating channel of the AP, can be a good attack signature to indicate an intentional reactive jamming attack. Accordingly, we set the malformed rate threshold (TH3) to 50% in order to detect behavior of reactive jamming

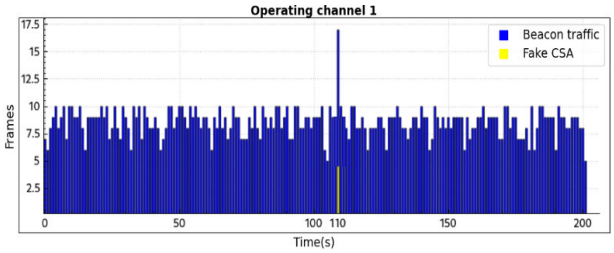


FIGURE 10. Traffic during fake CSA attack.

attacks during a probe interval and provide a warning sign for possible MC-MitM-BVR attacks.

3) CHANNEL SWITCH ANNOUNCEMENT ATTACK

CSA attacks can be conducted in three scenarios as discussed in Section III-A.3. The first two scenarios clearly identify an attack. To verify the third scenario (when an AP operates on 5 GHz with DFS enabled), we monitor the DFS characteristics in our home network for six months and confirm that the operating channel has not been changed. This supports our assumption that radar signals are uncommon in home networks. Therefore, the occurrence of CSAs can be considered as dubious network traffic even when DFS is enabled.

Figure 10 provides a view of fake CSA attacks on the AP's operating channel. It shows the traffic generated by the beacons on the operating channel of the AP even after the occurrence of CSAs at around 110 seconds, which should not happen when a genuine CSA occurs. This happens because the legitimate AP is unaware of the spoofed CSAs sent by the attacker, and it keeps broadcasting beacons on the same channel.

To study the behavior of benign traffic in home networks when a genuine CSA occurs, we invoked a channel switch (from channel 36 to 40) on a hostapd (access point daemon software) by sending the CSA command over the hostapd_cli interface [59].

Figure 11 depicts the behavior of traffic during a genuine channel switch. Here, we monitored the operating channel 36 and the new channel 40 simultaneously and observed that there is no traffic on operating channel 36 (see Figure 11(a)) after the occurrence of CSAs at around 100 seconds. At the same time, the legitimate AP begins its traffic on the new channel 40 only after 100 seconds (see Figure 11(b)).

In addition, we collected some real CSAs from a location near an airport by wardriving or sniffing on different DFS channels. For example, we observed a CSA instructing to switch from channel 60 to channel 64. We then monitored both channels simultaneously using the BSSID of the AP for a period of time (60 to 180 seconds) and were only able to collect traffic on the new channel 64, which is the same behavior as explained in Figure 11.

In essence, when a channel switch occurs, the AP stops transmitting on the current channel and starts transmitting on the newly designated channel. In accordance with

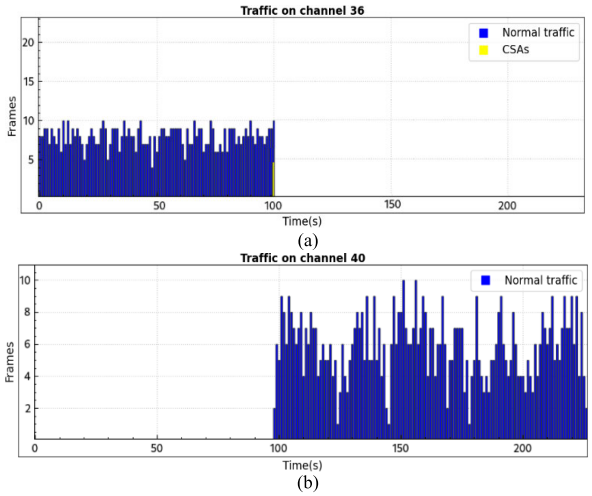


FIGURE 11. Traffic during a genuine channel switch on (a) current channel; (b) new channel.

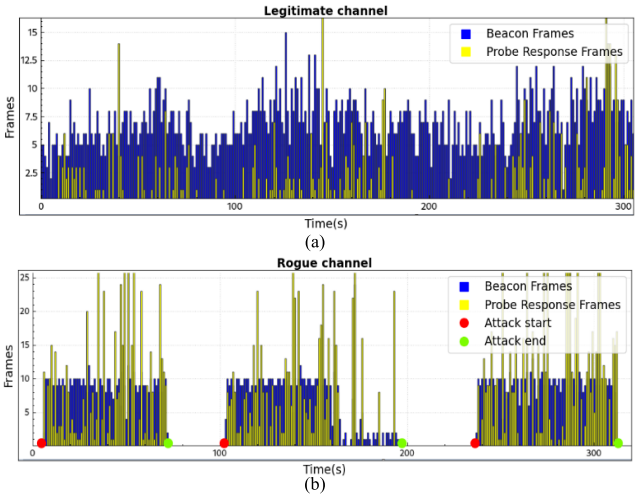


FIGURE 12. Attack network trace with beacons and probe responses on (a) legitimate channel; (b) rogue channel.

standards [58], a Wi-Fi network may typically experience 4 or 5 CSA frames during a channel switch. This can be observed in both Figure 10 and Figure 11, which respectively depict the CSA scenarios for attack and benign traffic. These observations serve as the foundation for establishing the threshold (TH4) to 1 CSA frame. Thus, we can quickly identify potential fake CSA frames and provide a warning sign for possible MC-MitM-IV attacks.

DFS detectors can also be used to verify the occurrence of CSAs [18]. However, such detectors recognize radar pulses and require advanced device setup, increasing the cost of attack detection significantly. This is the reason why, instead, we propose a simple way to detect potentially fake CSAs, and then we employ the attack signatures discussed in the following section to finally confirm the detection of an attack.

C. DESIGNING SIGNATURES OF STAGE 2 ATTACK TRAFFIC

In this section, we design signatures of stage 2 attack traffic (see Section III-B) to distinguish and confirm the presence of MC-MitM attacks. Furthermore, to identify the stage 2 attack traffic, we simultaneously monitor the legitimate APs and the rogue channels used for a probe interval of 5 minutes. We then launch 3 periods of MC-MitM attacks for 60-100 seconds.

1) CONCURRENT BEACON AND PROBE RESPONSE TRAFFIC

Figure. 12 shows an attack network trace with concurrent beacons and probe responses on two different channels with the same SSID and BSSID. We also analyze benign traffic scenarios, and we have not been able to detect any concurrent beacon or probe response traffic on multiple channels in the same frequency band with the same SSID and BSSID in the target Wi-Fi network.

On the other hand, there may be concurrent beacons if home APs broadcast the same SSID when operating on dual-band frequencies (both 2.4 GHz and 5 GHz). However, such concurrent beacons can be easily distinguished as benign traffic since the channels used in the 2.4 and 5 GHz bands are different.

Therefore, the sudden arrival of a significant number of concurrent beacons or probe response traffic on two different channels with the same SSID and BSSID in a WLAN, following the warnings generated by the stage 1 traffic analysis (see Section IV-B), clearly indicate the beginning of MC-MitM attacks. Accordingly, we set the threshold (TH5) to 1 beacon or probe response frame for quicker identification of concurrent beacon or probe response traffic accompanying the MC-MitM attacks during a probe interval. Furthermore, we confirm the presence of MC-MitM attacks by using the subsequent concurrent traffic in a WLAN.

2) CONCURRENT CONNECTION ESTABLISHMENT TRAFFIC

In Figure 13, we show an attack network trace with concurrent authentication frames on two different channels with the same SSID and BSSID. Figure 14 shows an attack network trace with the presence of concurrent association request and response frames on two different channels with the same SSID and BSSID. Finally, Figure 15 shows an attack network trace with the presence of concurrent EAPOL frames on two different channels with the same SSID and BSSID. The traffic shown in these three figures is only possible when an attack is in process, as no such concurrent traffic can occur on different channels with the same SSID/BSSID.

Therefore, this concurrent connection establishment traffic can be used as an attack signature to detect the presence of MC-MitM attacks in a WLAN. Taking this into account, we set the threshold (TH6) to 1 authentication, association, and EAPOL frames accompanying the MC-MitM attacks during a probe interval. This enables a fast identification of concurrent traffic.

3) CONCURRENT DATA TRAFFIC

Figure 16 shows an attack network trace with concurrent data on two different channels with the same SSID and BSSID.

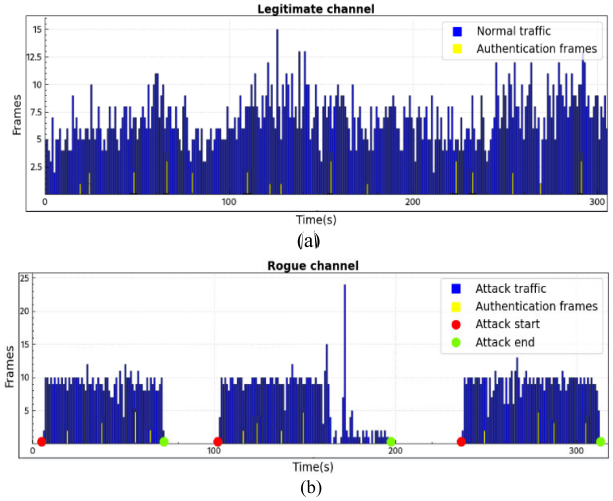


FIGURE 13. Attack network trace with concurrent authentication frames on (a) legitimate channel; (b) rogue channel.

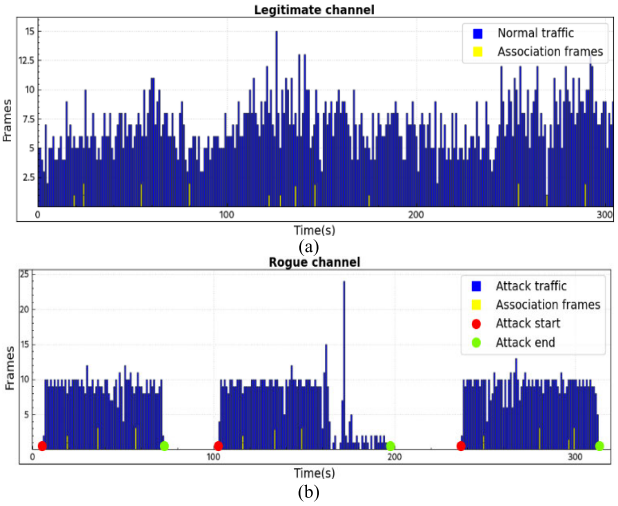


FIGURE 14. Attack network trace with concurrent association frames on (a) legitimate channel; (b) rogue channel.

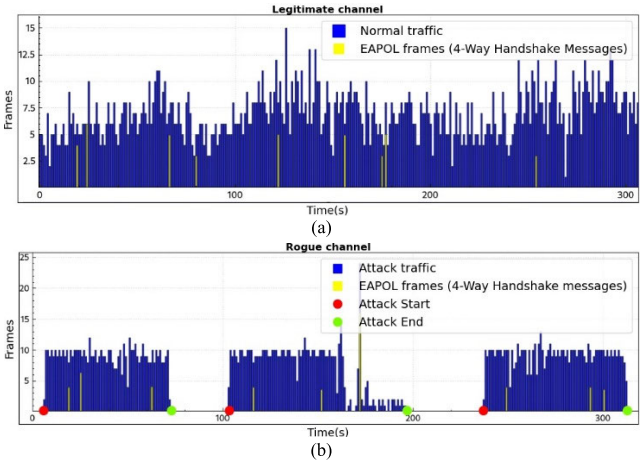


FIGURE 15. Attack network trace with concurrent EAPOL frames (4-Way Handshake messages on (a) legitimate channel; (b) rogue channel.

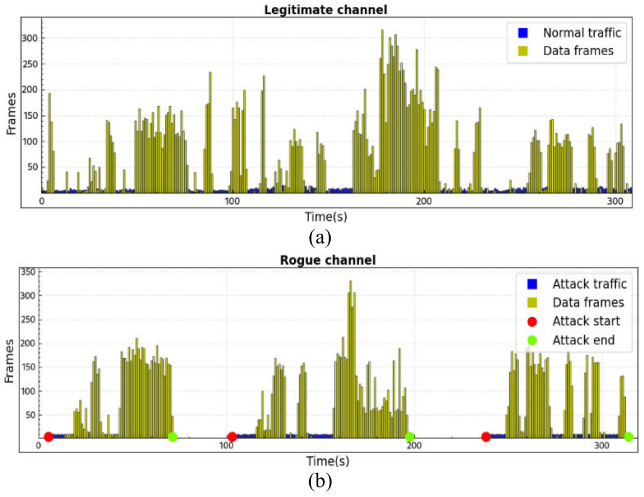


FIGURE 16. Attack network trace with data frames on (a) legitimate channel; (b) rogue channel.

As in the previous case, the data frames exchanged between the legitimate and rogue channels with the same SSID/BSSID can be used as a trigger for attack detection, since they are impossible considering the Wi-Fi protocol's normal operation. Therefore, we set the threshold (TH7) to 1 data frame for quicker identification of concurrent data traffic.

D. SUMMARY

Table 2 summarizes the attack signatures we propose for the detection of MC-MitM attacks during a probe interval. We must emphasize that thresholds TH4, TH5, TH6 and TH7 are grounded on the theoretical analysis of the operation of the Wi-Fi protocol. This makes it impossible for the MC-MitM attacker to execute an attack and remain undetected, unless some frames are missed due to network failures.

The remaining thresholds (TH1, TH2, TH3), instead, are derived from our empirical analysis and, therefore, we must admit that we cannot claim with complete certainty that no attack will go undetected in stage 1 using those thresholds. However, our empirical analysis and validation results shown in Section VI show that those thresholds are still very useful and can detect attacks in stage 1.

Even in the case that an attack was undetected in stage 1, we must emphasize that our protocol has two stages, and we can ensure that our approach would detect this attack in stage 2, because stage 2 uses only theoretical thresholds that no attack can elude (unless network conditions result in significant frame loss and this affects the detector's capabilities).

We also remark that, although stage 2 attack traffic can be used to detect MC-MitM attacks, both stage 1 and 2 attack signatures are necessary to distinguish between the MC-MitM different attack variants.

TABLE 2. Summary of attack signatures.

	Attack signature	Metrics used	Thresholds
Stage 1 attack traffic	Constant jamming	FIAT and FDR	TH1 ≥ 2 ms for FIAT and TH 2 < 50% for FDR
	Reactive Jamming	Malformed frame rate	TH3 ≥ 50%
	Fake CSAs	Number of CSAs	TH4 ≥ 1
Stage 2 attack traffic	Concurrent beacon traffic	Number of beacons or probe responses	TH5 ≥ 1
	Concurrent connection establishment traffic	Number of authentications, associations, or EAPOL frames	TH6 ≥ 1
	Concurrent data traffic	Number of data frames (optional metric)	TH7 ≥ 1

Furthermore, the datasets created in this work (attack network traces captured in the form of PCAP format with MAC layer frames) are made available in [60]. Our dataset is the first of its kind to provide traffic specifically from MC-MitM attacks and their variants.

V. PROPOSED SOLUTION: A SIGNATURE-BASED WIRELESS INTRUSION DETECTION FRAMEWORK FOR MC-MITM ATTACKS

In this section, we present the system architecture of the proposed solution, its architectural units, and the methodology to detect MC-MitM attacks by using attack signatures (malicious frames) discussed in the previous section.

A. SYSTEM ARCHITECTURE

Our proposed framework is based on a plug-and play, centralized, online passive monitoring system that can be easily integrated into any Wi-Fi or Wi-Fi-based IoT network. As presented in the previous section, we perform a signature-based network analysis to quickly and accurately detect abrupt and highly deviating changes in the network traffic due to MC-MitM attacks. Our framework is independent of the encryption techniques (WPA, WPA2, or WPA3), personal or enterprise networks, PMF standards, and Wi-Fi frequency bands (2.4 and 5 GHz) used in Wi-Fi networks.

Figure 17 shows the high-level system architecture of our proposed framework with overall workflow. It composed of four main units: traffic interceptor, device database unit, MC-MitM detection coordinator unit, and alert generator unit.

Below, we provide brief description of various units:

1) TRAFFIC INTERCEPTOR UNIT

The traffic interceptor unit passively monitors network traffic in a WLAN and collects suitable management frames (beacons, probe responses, action frames, connection establishment frames). This unit filters required frames based on

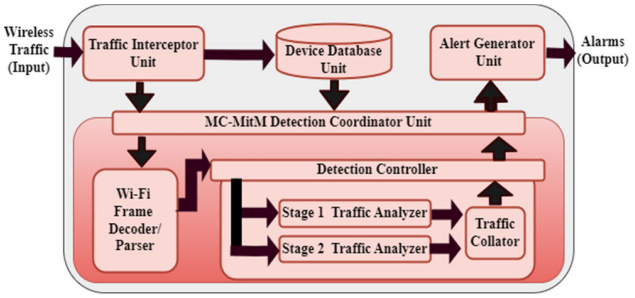


FIGURE 17. High-level system architecture.

MAC address of the AP and forwards them to the device database unit and the MC-MitM detection coordinator unit for further analysis.

2) DEVICE DATABASE UNIT

This unit automatically collects MAC addresses of clients connected to the legitimate AP in the targeted WLAN. Such information is provided to the MC-MitM detection coordinator unit to facilitate network analysis and scrutiny.

3) MC-MITM DETECTION COORDINATOR UNIT

This unit acts as the center of the detection process. Its main job is to analyze the network traffic and coordinate various processes to identify attack signatures associated with MC-MitM attacks during a probe interval. This unit also hosts the following two modules.

- **Wi-Fi frame decoder:** This module filters and analyzes network traffic between the AP and legitimate clients in a WLAN. It extracts low-level MAC layer header details from each frame, including type, subtype, ESSID, BSSID, operating channel, and more. These parsed frames are then sent to the detection controller.
- **Detection controller:** This module implements a detection methodology (see Section V-B) to identify the specific traffic associated with MC-MitM attack variants. It has three sub-modules. Sub-modules, such as stage 1 and stage 2 traffic analyzers, respectively, record the number of network frames that correspond to the stage 1 and stage 2 attack signatures (see Table 2). Finally, the traffic collator sub-module verifies the status of stage 1 and stage 2 traffic analysis and decides whether an MC-MitM attack is occurring, identifies its variant, and then hands over the details of the attack to the alert generator unit.

4) ALERT GENERATOR UNIT

This unit creates alerts in case of security events. It mainly logs the alerts with the MAC address of victims, time, and date of the attack.

B. DETECTION METHODOLOGY

In this section, we illustrate the detection methodology followed by the detection controller of our framework.

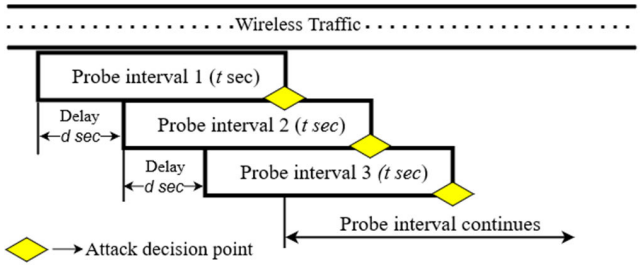


FIGURE 18. Probe interval management.

1) DETECTION LOGIC

The core of the proposed framework's operation is a set of network analysis algorithms that scrutinize filtered and parsed network traffic from the Wi-Fi frame decoder. The detection controller orchestrates the entire detection process using a probe interval structured as a sliding window [61] as shown in Figure 18. Each probe interval duration lasts up to t seconds. Conversely, the second probe interval starts with a delay of d seconds (referred to as the inter-probe interval delay) after the initiation of the first probe interval. Similarly, the third probe interval commences after a lapse of d seconds from the beginning of the second interval, and so forth. This sliding window mechanism serves to ensure that even if an MC-MitM attack fails to breach the predefined thresholds during a specific probe interval, subsequent intervals are still capable of detecting such attacks. This approach allows our framework to maintain continuous monitoring and make determinations about potential attacks every d second, beginning immediately after the first probe interval.

Figure 19 illustrates the overall attack detection logic in a probe interval. The controller module invokes the stage 1 and stage 2 traffic analyzers, which host specific algorithms to detect respective attack signatures. The detection logic comprises 11 algorithms (see Appendix A). Algorithms 1, 2, and 3 belong to the stage 1 traffic analyzer, while algorithms 4, 5, 6, 7, and 8 belong to the stage 2 traffic analyzer. Algorithms 9 and 10, respectively, determine whether the analyzed traffic is malicious or not based on the threshold values (see Table 2). Finally, algorithm 11 acts as a traffic collator that decides on the presence of MC-MitM attacks at the end of each probe interval.

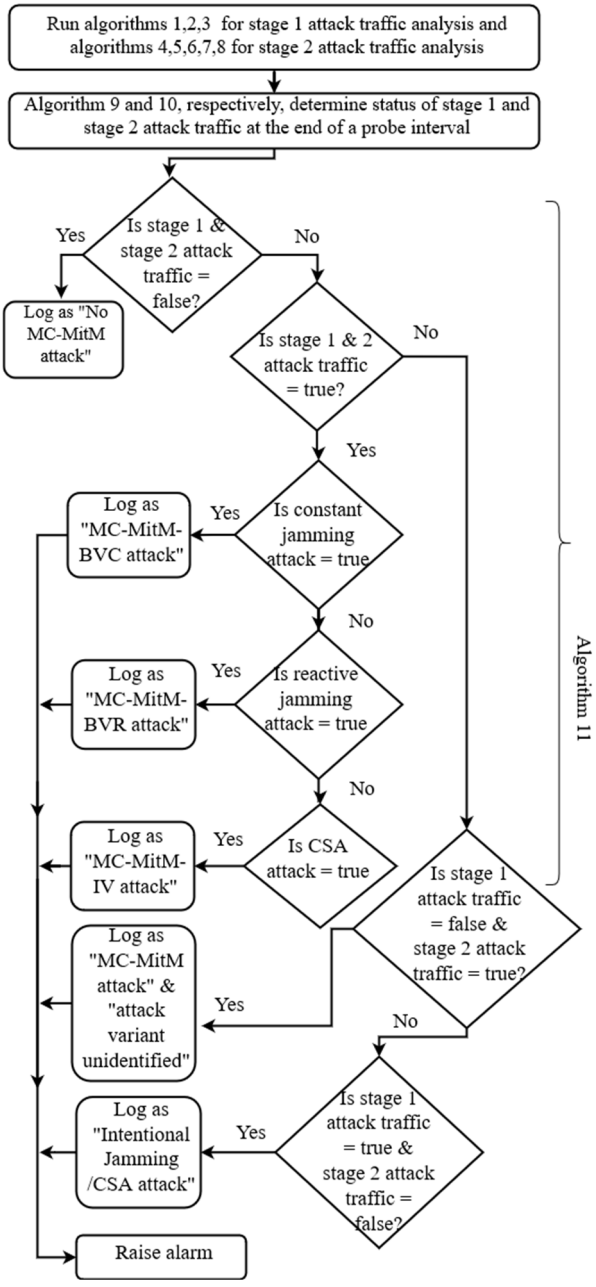


FIGURE 19. Overall detection logic in a probe interval.

VI. EVALUATION

This section is dedicated to the evaluation of the proposed signature-based wireless intrusion detection system (SWIDS) framework for detecting MC-MitM attacks in a representative set of scenarios, with a particular emphasis on personal networks, while still being applicable to enterprise networks.

A. FRAMEWORK IMPLEMENTATION

Our proposed SWIDS framework, outlined in Figure 17, consists of four units implemented in Python. The traffic interceptor unit utilizes different wireless interfaces: TL-WN722N for 2.4 GHz and Wi-Fi Nation for 5 GHz, chosen for cost-effectiveness and monitor mode support across Linux distributions. The device database and MC-MitM detection coordination units use the Scapy libraries to process network packets. Additionally, we incorporated a log file feature to facilitate the tracking of alerts generated by our framework. In this paper, we present the proof of concept (PoC) of our framework implemented on a laptop with Kali Linux OS and is made available in [27].

B. EXPERIMENTAL METHODOLOGY

We conducted our experiment in a practical smart-home environment having an area of approximately 100 sq.m

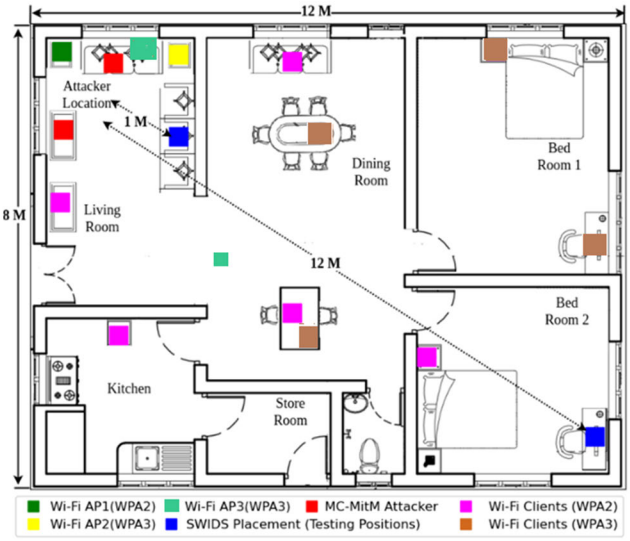


FIGURE 20. Experimental testbed.

(square meter). We employed 15 devices: 3 APs, 9 client devices, 2 attacker devices, and 1 SWIDS device (details of devices used are provided in Table 3). A mixed-mode Wi-Fi network encompassing both WPA2 and WPA3 standards was deployed to accommodate a diverse range of heterogeneous client devices. We connected 5 WPA2 compatible Wi-Fi clients to AP1 and 2 WPA3 compatible client to AP2, and 2 WPA3 compatible client to AP3. For the attackers, we used 2 laptops: one to conduct MC-MitM-IV attacks and another for MC-MitM-BVC or MC-MitM-BVR attacks. The system's performance was evaluated by placing the SWIDs device at two different locations: 1 meter away and 12 meters away from the attacker's location. These locations were chosen to examine the system's effectiveness at both short and possible long distances within our experimental testbed. Our experimental testbed is shown in Figure 20, which illustrates the placement of test devices.

We first performed a set of experiments with the aim of determining an optimal probe interval duration that achieves a true positive rate (TPR) of 90% or higher, considering factors such as the detection time, which refers to the overall time needed to detect different MC-MitM attacks. These experiments were performed within our experimental testbed under normal network traffic conditions, meaning there was no network congestion, and all devices were connected to their respective routers.

Our SWIDS detector node was placed at varying distances from the attacker's location. Specifically, we conducted a series of 75 tests, comprising 25 tests for each of the three MC-MitM attack variants, at a distance of 1 meter. Additionally, we conducted an equivalent number of 75 tests, with 25 tests allocated to each MC-MitM attack variant, at a distance of 12 meters. The results of this first set of experiments are described in Section VI-C.

Once we had determined the probe interval duration needed to reach the desired 90% TPR, we proceeded with the second set of experiments. In this phase, we aimed at testing how effectively our framework prototype could detect different MC-MitM attack variants at various distances under light and heavy traffic conditions. Following a similar approach to the first set of experiments, we conducted 25 detection tests of each MC-MitM attack variant at a distance of 1 meter and another 25 detections of each MC-MitM attack at a distance of 12 meters from the attacker's location. Further, we conducted these experiments in 2 GHz bands. We recreated the light and heavy traffic scenarios within the experimental testbed in the following manner:

TABLE 3. Devices used in the experimental testbed.

Device	Type/Role	Wi-Fi standard
TP-LINK Wireless (802.11 N) Router, Speed-144 Mbps, Channel 1(2 GHz), TX power-25-30 dBm	Wi-Fi AP1	WPA2-PSK
D-Link Wireless AX 1500 (802.11 B/G/N) Wi-Fi 6 Router, Speed- 1200 Mbps, Channel 36 (5 GHz), TX power-14 dBm	Wi-Fi AP2	WPA3-SAE
D-Link Wireless AC 1200 (802.11 B/G/N) Wi-Fi 5 Router, Speed-800 Mbps, Channel 36 (5 GHz), TX power-14-17 dBm	Wi-Fi AP3	WPA3-SAE
Samsung S7-Edge	Wi-Fi client of AP1	WPA2
TP-Link Smart Bulb L510E	IoT sensor (Wi-Fi client of AP1)	WPA2
TP-Link Smart Plug P100	IoT sensor (Wi-Fi client of AP1)	WPA2
TP-Link Smart Plug P100	IoT sensor (Wi-Fi client of AP1)	WPA2
Samsung Smart TV	Wi-Fi client of AP1	WPA2
Samsung S8	Wi-Fi client of AP2	WPA2
iPad Mini	Wi-Fi client of AP2	WPA3
Samsung S22	Wi-Fi client of AP3	WPA3
Dell Inspiron 15 30000 series	Wi-Fi client of AP3	WPA3
Lenovo-Thinkpad	Attacker (MC-MitM-IV)	WPA2/3
Toshiba Portege R500	Attacker (MC-MitM-BVC or MC-MitM-BVR)	WPA2/3
HP Elite 8300 with High Gain TP-Link TL-WN722N/Wi-Fi Nation Wi-Fi adaptors	SWIDS Framework	Any

1) LIGHT TRAFFIC SCENARIO

We set up a total of 5 Wi-Fi clients connected to Wi-Fi AP1. Wi-Fi AP1 was configured to support IEEE 802.11n mode, and we set the channel frequency to 2.4 GHz with a channel width of 20MHz. This configuration ensured a bitrate (data rate) of 144 Mbps [62]. During the experiments, the connected clients were engaged in web browsing, video streaming, and social media activities, generating a realistic workload representative of light network usage.

TABLE 4. Metrics summary.

Metric	Description	Method
True positive rate (TPR)	Proportion of correct positives relative to total real positives.	(TP) / (TP + FN)
True negative rate (TNR)	Proportion of correct negatives relative to total real negatives.	TN / (FP + TN)
F1-score	Harmonic mean of positive predictive value (precision) and true positive rates.	2TP / (2TP + FP + FN)

2) HEAVY TRAFFIC SCENARIO

For this scenario we utilized a total of 3 Wi-Fi clients connected to Wi-Fi AP1. To create a large volume of wireless traffic, we downloaded large Blu-ray files using P2P connected clients. In addition, we employed the *iperf* tool [63] to generate a maximum bitrate of 100 Mbps. It was ensured that the overall bitrate consistently saturated the channel bandwidth by exceeding 90% during the experiments. The results of this second set of experiments are presented in Section VI-D.

In our third set of experiments, we assess the detection performance of our framework in modern Wi-Fi routers, such as 802.11ac (Wi-Fi AP2) and 802.11ax (Wi-Fi AP3). This evaluation is particularly focused on examining the effects of various channel widths as well as effects of primary and secondary channels. We primarily conducted such experiments in 5 GHz bands. Primary and secondary channels in 5 GHz serve the purpose of optimizing spectrum utilization and minimizing interference, in accordance with the regulatory standards set by each country. In these experiments as well, we conducted 25 detection tests of each MC-MitM attack variant at a distance of 1 meter and another 25 detections of each MC-MitM attack at a distance of 12 meters from the attacker's location, all under light or normal traffic conditions. The results of this second set of experiments are presented in Section VI-E.

In our fourth and final set of experiments, we evaluate performance of our proposed SWIDS framework in terms of CPU and memory utilization. The results of this set of experiments are presented in Section VI-F.

To evaluate the performance capabilities of our framework, we examined the alarm or attack detection status in each experiment by analysing the log file generated by our SWIDS framework. We utilized various metrics as summarized in Table 4. The classification of each prediction result in our framework was based on the following categories: true positive (TP), when an alarm is correctly raised during an attack; true negative (TN), when no alarm is generated in the absence of no attack; false positive (FP), when an alarm is raised erroneously in the absence of an attack; or false negative (FN), when no alarm is generated during an actual attack.

C. RESULTS AND DISCUSSION OF THE FIRST SET OF EXPERIMENTS

In this section, we present the results obtained from our first set of experiments aimed at determining the appropriate probe interval duration and the corresponding detection time for MC-MitM attacks within the experimental testbed.

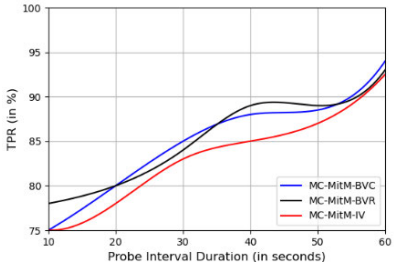


FIGURE 21. Average TPR observed from equal number of detection tests conducted at 1-meter and 12-meter distances under different probe interval duration.

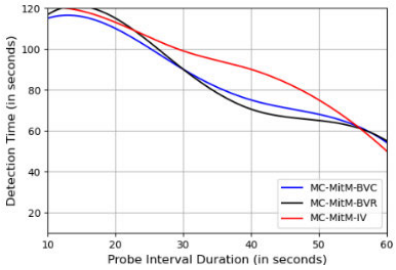


FIGURE 22. Average detection time observed from equal number of detection tests conducted at 1-meter and 12-meter distances under different probe interval duration.

In Figure 21, we illustrate the average TPR as a function of the probe interval duration for the three types of MC-MitM attacks, including the base variant attacks (MC-MitM-BVC and MC-MitM-BVR) as well as improved variant attack (MC-MitM-IV). As seen from Figure 21, we can observe that our framework achieves an average TPR exceeding 93% when employing a probe interval of 60 seconds. This is because, with a probe interval of 60 seconds, our framework collects a sufficient amount of attack frames and data to potentially distinguish different MC-MitM attacks and their variants. This also highlights the superior performance of our algorithms when longer observation times are employed.

In Figure 22, we illustrate the average detection time (time delay to detect different MC-MitM attacks) at 1-meter and 12-meter distances. As seen from Figure 22, we can observe that the detection time decreases as the probe interval increases due to the availability of a larger pool of attack data. Specifically, when the probe interval duration is set to 60 seconds, our framework achieves an average detection time of 50-55 seconds. This indicates an average improvement of 45-50% in the detection time compared to a probe interval duration of 10 seconds.

Since the 60 seconds probe interval duration allows our framework to achieve the desired TPR with a considerable low detection time, we have adopted this duration for all subsequent experiments in our framework. Additionally, based on our experiments where we observed a significant improvement in detection performance with a 10 seconds inter-probe

delay (d seconds in Figure 18), we have configured the inter-probe interval delay to 10 seconds in our detection logic to reduce the chances of missed attacks.

The values we have obtained for t and d can be applied to other environments with different hardware settings, and we can assure that the detection results obtained will remain equally acceptable (TPR $>$ 90%). This is due to the high detection capacity of stage 2. Even when the attack traffic in stage 1 lasts for a long time, usually as a result of reactive jamming attacks, the attack is swiftly detected in stage 2, because the thresholds in this stage are equal to one and, therefore, concurrent traffic is detected almost instantaneously. Consequently, by employing a probe interval of 60 seconds and an inter-probe delay of 10 seconds, we maximize the detection possibilities of stage 1, and when this stage fails and the specific type of attack cannot be determined, the alarm is triggered in stage 2 and the attack variant is marked as ''unidentified''. The last outcome can be caused by long reactive jamming attacks or a high packet loss ratio.

D. RESULTS AND DISCUSSION OF THE SECOND SET OF EXPERIMENTS

In this section, we present the results obtained from our second set of experiments, which aimed to evaluate the performance of the SWIDS framework in detecting various MC-MitM attacks under both light and heavy traffic scenarios. These experiments were conducted to assess the effectiveness and reliability of the SWIDS framework in real-world network environments with different traffic conditions.

In. Figure 23, we show the detection performance achieved under light and heavy traffic scenarios at a short-distance (1 meter) and long-distance (12 meters) from the attacker's location. As seen from Figure 23, our proposed framework demonstrates the capability to detect different MC-MitM attack variants with a minimum TPR of 83% at 1-meter distance and 70% at 12-meter distance under various traffic scenarios. Among the results, the detection of MC-MitM-BVC (see Figure 23(a) and (b)) and MC-MitM-BVR (see Figure 23(c) and (d)) attacks exhibits the most favorable performance. This can be attributed to the effectiveness of our framework's stage 1 attack traffic detection. In the case of MC-MitM-BVC attacks, constant jamming results in abrupt changes in the corresponding FIAT and FDR values, which our framework can promptly detect even at longer distances and under heavy traffic scenarios. Similarly, reactive jamming employed in MC-MitM-BVR attacks induces many malformed frames, which provide sufficient evidence for our framework to detect such attacks during specific probe intervals. However, the detection of MC-MitM attacks presents some challenges. In certain instances of MC-MitM-IV attacks, fake CSA attacks remain undetected as there are only a few CSAs (4 CSA beacons as per standards) in an attack, which may be lost or dropped during detection. This is mainly observed at 12 meters and in heavy traffic scenarios (see Figure 23(f)).

Moreover, the stage 2 attack introduces frame loss, especially in the case of concurrent connection establishment traffic, since such traffic consists of fewer frames (2 authentication, 2 associations, and 4 EAPOL frames) than concurrent beacon/probe response and concurrent data traffic. Consequently, the detection of all MC-MitM attack variants is affected.

The decrease in the obtained TPR at longer sensing distances can be primarily attributed to an increased frame loss rate experienced by our framework during different probe intervals. Frame loss can occur due to the network conditions, parsing and processing time for each frame, and the processing power of the Wi-Fi cards. Consequently, our framework may misclassify a certain fraction of attacks as benign traffic (see Figure 23(d), (e), and (f)).

Due to the frame loss at distances of 12 meters or under heavy traffic scenarios, stage 1 attack traffic remains undetected in a few cases. Yet, our framework successfully detected MC-MitM attacks that involved a combination of concurrent beacon/probe response traffic with concurrent connection establishment traffic or concurrent data traffic. However, identifying the specific attack variants in such cases proved challenging, resulting in an average of 3% of uncategorized MC-MitM attacks during our experiments.

Regarding the performance difference between light and heavy traffic scenarios, our framework exhibits good performance under both scenarios at 1-meter distances, with only an average 5% drop in detection accuracy under heavy traffic scenarios compared to light traffic scenarios. However, at a distance of 12 meters, there is an average performance drop of 14% under heavy traffic scenarios. This is because, frame loss is more prevalent at long distances.

Furthermore, our framework shows good confidence in correctly distinguishing attacks, with 100% TNR in all test scenarios. As a consequence, there are also no false positives (although not explicitly shown in Figure 23) as we used predefined rules to identify the signatures of stage 1 and stage 2 traffic during MC-MitM attacks. Finally, our framework maintained reasonable F1-scores (above 82%) in all test scenarios.

E. RESULTS AND DISCUSSION OF THE THIRD SET OF EXPERIMENTS

In this section, we provide the outcomes of our third set of experiments. These experiments were conducted to assess the performance of our SWIDS framework in the detection of various MC-MitM attacks across varying channel bandwidths, and primary and secondary channels in the 5 GHz bands of modern Wi-Fi networks. Furthermore, the evaluation covered a different detector location from the attacker.

In Figure 24, we show the average TPR while detecting different MC-MitM attacks under each different channel bandwidths of the Wi-Fi AP2 (802.11ac) and Wi-Fi AP3 (802.11ax) as specified in Table 3, from detection tests conducted at 1-meter and 12-meter distances from the attacker location. As shown in Figure 24 (a) and (b), we can see that

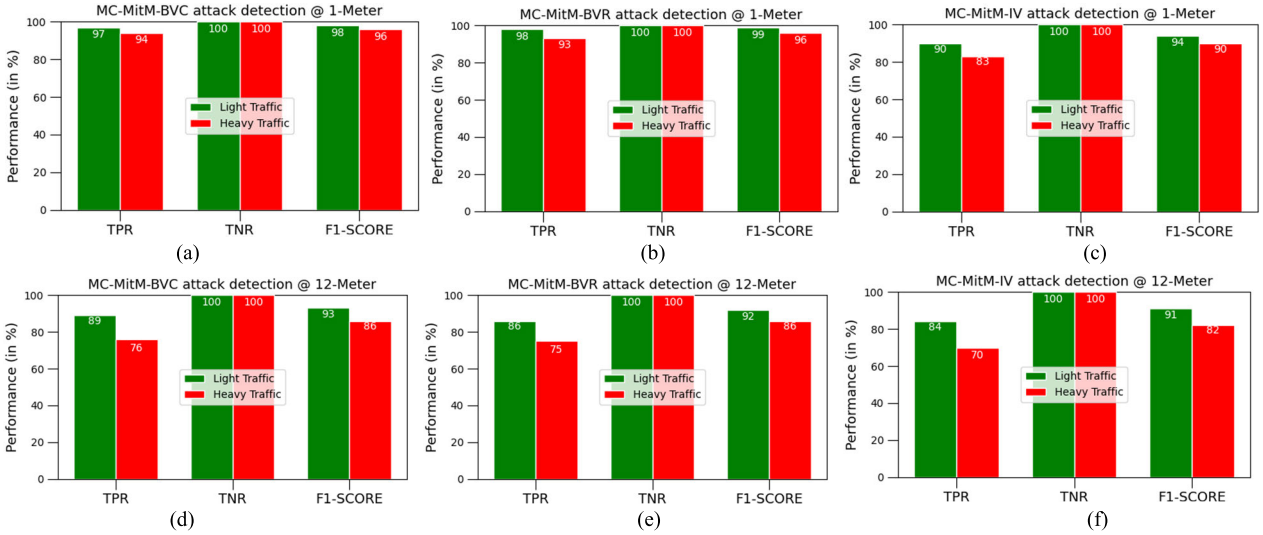


FIGURE 23. Detection performance achieved under light and heavy traffic scenarios with (a) MC-MitM-BVC at 1-meter; (b) MC-MitM-BVR at 1-meter; (c) MC-MitM-IV at 1-meter; (d) MC-MitM-BVC at 12-meter; (e) MC-MitM-BVR at 12-meter, and (f) MC-MitM-IV at 12-meter.

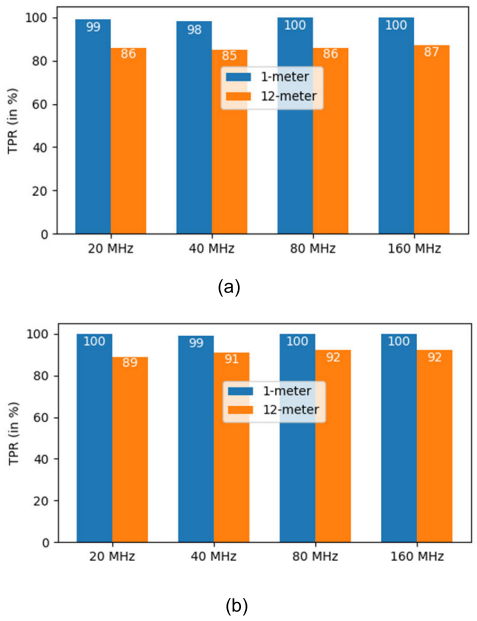


FIGURE 24. Average TPR observed under different channel bandwidths at 1-meter and 12-meter distances (a) with 802.11ac networks; (b) with 802.11ax networks.

our SWIDS framework effectively detect various MC-MitM attack variants, achieving an average TPR of up to 99% in both 802.11ac and 802.11ax networks at a 1-meter distance.

At a 12-meter distance, the TPR averages at 86% for 802.11ac and 91% for 802.11ax networks. This signifies a decline in TPR, about 13% for 802.11ac and 8% for 802.11ax, when comparing the 1-meter and 12-meter distances. This clearly demonstrates that distance of detector from the attacker is the primary factor influencing the performance of our framework. On the other hand, the detection performance remains relatively consistent across all channel bandwidths of both 802.11ac and 802.11ax networks.

This is because, as per standards, the maximum transmitted power (e.g., 14 dBm in our experiments) set in an AP remains constant regardless of the channel bandwidth [63], [64], which mainly affect the reception of frames by our detector. While this high transmit power enables Wi-Fi frames to cover extended distances, it results in a lower Received Signal Strength Indicator (RSSI) when these frames encounter obstacles such as walls in a home or buildings, leading to potential frame loss. Therefore, it becomes apparent that a wider channel bandwidth does not significantly impact our framework's detection performance.

In Figure 25, we show the average TPR while detecting different MC-MitM attacks across primary and secondary channels (any adjacent channel) under each different channel bandwidths. These results stem from an equal number of detection tests conducted in both 802.11ac and 802.11ax networks. Additionally, the evaluation considered different detector locations from the attacker.

In the context of our channel experiments, we chose commonly used non-overlapping channels to minimize the potential for interference from adjacent networks. Specifically, we selected primary and secondary channel pairs as follows: 36 and 40 for 20 MHz, 36 and 44 for 40 MHz, 36 and 52 for 80 MHz, and 36 and 100 for 160 MHz [64], [65] both in 802.11 ac and 802.11ax networks.

As shown in Figure 25 (a), we can see that our SWIDS framework effectively detect various MC-MitM attack variants with an average TPR exceeding 97% in both primary and secondary channels across different channel bandwidths at a 1-meter distance.

Similarly, at a 12-meter distance, our framework maintains an average TPR of at least 85% in both primary and secondary channels, regardless of the channel bandwidth. Furthermore,

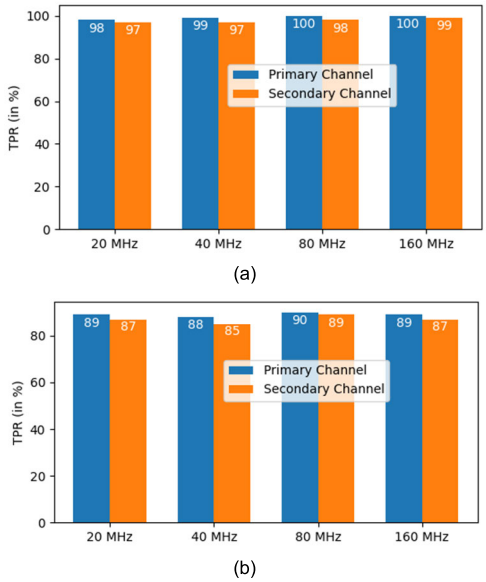


FIGURE 25. Average TPR observed under primary and secondary channels of different channel bandwidths in both 802.11ac and ax networks at: (a) 1-meter; (b) 12-meter distances.

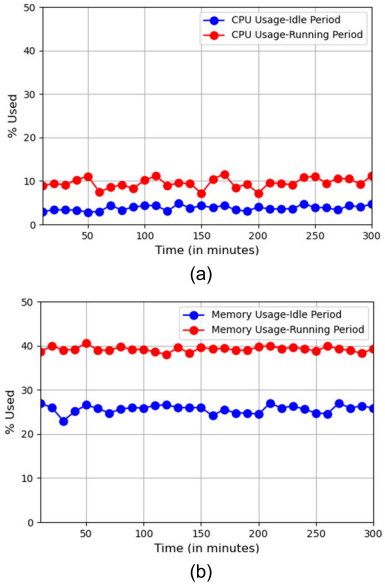


FIGURE 26. Performance overhead of SWIDS framework in terms of (a) CPU usage; (b) Memory usage.

in Figure 25 (a) and (b), it is evident that there is a decrease in detection performance of approximately 11-12% when comparing a 1-meter distance to a 12-meter distance. This further reinforces our findings that distance is the primary factor influencing the detection performance. Consistent with our previous experiments, it is evident from Figure 25 (a) and (b) that the detection performance remains relatively stable across various channel bandwidths. This indicates that the choice of the operating channel in Wi-Fi also does not significantly impact our framework's ability to detect MC-MitM attacks.

Finally, from Figures 24 and 25, we conclude that the primary factor contributing to the performance drop lies in frame loss due to the distance between the detector and attacker locations or frame processing delays within our SWIDS framework. Additionally, network conditions and environmental factors, including traffic volume, building materials, and network overhead, contribute to reduced wireless signal range and throughput. These experiments further reinforce the findings presented in Section VI-D. Nevertheless, our framework exhibits relatively good detection performance, particularly in modern 802.11ax or Wi-Fi 6 enabled networks. This improvement is attributed to the increased Received Signal Strength Indicator (RSSI) in wireless frames received at our detectors as well as improved transmission features and throughput.

F. RESULTS AND DISCUSSION OF THE FOURTH SET OF EXPERIMENTS

In order to test the performance overhead of our proposed defense mechanism in terms of CPU and memory utilization, we conducted an experiment on a Kali Linux laptop (Intel core i3 with 4GB RAM) hosted with our proposed defense mechanism. More specifically, we measured how much CPU and memory were used in the previous N minutes (in our case, we used 300 minutes) by our SWIDS framework during its idle and running periods (see Figure. 26).

As shown in Figure 26 (a), we can observe that the CPU usage increases by an average of only 5% when SWIDS framework is active, which can be primarily attributed to Wi-Fi frame capture and subsequent extraction procedures. Regarding memory consumption (see Figure 26(b)), when our SWIDS framework is active, there is merely a 12% average increase (0.48GB). This is because the defense mechanism stores solely the quantity of malicious frames and the status of the corresponding attack traffic during each probe interval duration.

The efficiency of the proposed framework becomes evident with the aforementioned results. It qualifies as a lightweight solution ideal for low-cost devices like a Raspberry Pi 4 with 8GB of RAM and featuring a quad-core cortex-a72 processor[2], which has a performance of 20% of an Intel Core i3-7100. Note that in a raspberry pi 4, SWIDS would consume around 25% of the CPU and 6% of the RAM.

G. COMPARISON WITH EXISTING DEFENSE MECHANISMS

In this section, we compare our proposed SWIDS framework with existing state-of-the-art defense mechanisms, particularly stage 1 defense mechanisms (see Section II-B.1) since they identify the root causes or attack vectors for MC-MitM attacks.

[2]Comparison of the performance of a single-thread CPU arm-cortex-a72 vs intel-core-i3-7100 https://versus.com/en/arm-cortex-a72-vs-intel-core-i3-7100

TABLE 5. Comparison of SWIDS with exisiting defense mechanisms.

Defense mechanism/ Metrics	Detect MC-MitM attack against WPA2/3 clients	Detect MC-MitM attack against PMF capable /incapable clients	Detect insider/outsider MC-MitM attacks	Provides detection and/or prevention of MC-MitM attacks	Recognize MC-MitM attack variant	Mandates Protocol changes/ Integration of software/ hardware /no changes	Provides backward compatibility
Proposed SWIDS framework	○	○	○	□	○	○	○
OCV [34]	■	□	○	○	■	□	■
Beacon Protection [16]	■	□	■	○	■	□	■
Stupify [37]	□	■	■	□	■	□	■
SSAD [19]	□	■	○	□	■	■	■
SAE-PK [35]	■	□	■	○	■	□	■

We do not consider stage 2 defense mechanisms since they focus only on detecting or preventing specific attacks (e.g., KRACK) using MC-MitM positions. Further, for comparison purposes, we consider various metrics, such as: (1) whether the defense mechanism detects MC-MitM attacks against WPA/2 clients (□), WPA3 clients (■), or both (○); (2) whether the defense mechanism detects MC-MitM attacks against PMF capable clients (□), against PMF incapable clients (■), or both (○); (3) whether the defense mechanism detects insider MC-MitM attacks (□), detects outsider MC-MitM attacks (■), or both (○); (4) whether the defense mechanism provides detection only (□) or both detection and prevention (○) of MC-MitM attacks; (5) whether the defense mechanism detects or recognizes all MC-MitM attack variants (○) or not (■); (6) whether the defense mechanism requires any protocol or firmware modification (□), integration of software/hardware (■), or no changes (○) for its deployment and (7) whether the defense mechanism provides backward compatibility to safeguard old or outdated devices (○) or not (■) from MC-MitM attacks. The comparison is illustrated in Table 5. The more open circles (i.e., icon ○) are shown in the row of a particular defense mechanism, the more effective the defense mechanism is for detection of MC-MitM attacks.

According to Table 5, OCV [34] and Beacon Protection [16] defense mechanisms detect and prevent MC-MitM attacks. However, these mechanisms are currently only available with WPA3 devices or PMF-enabled devices, since they have only recently been included in the WPA3 standards. Regarding the detection of insider/outsider attacks, while OCV effectively identifies both of these attacks as it checks for unauthorized communication channels during a 4-way handshake process, the Beacon Protection mechanism cannot detect insider attacks because attackers can forge legitimate beacons. Although the above mechanisms detect the presence of MC-MitM attacks, they cannot correctly identify which specific attack variant is being used.

Stupify [37] only detects attacks against WPA2 devices because it introduces changes to the WPA2 authentication mechanisms. It does not protect PMF devices as they do not include a group key (IGTK) in their authentication mechanism, and cannot detect insider attacks since such attackers can forge/bypass authentication details. Similarly, SSAD [19] can only detect and prevent attacks against WPA/2 devices because it introduces a new patch for wpa_supplicant for the WPA2 standards. It can also detect attacks against PMF devices and identifies insider and outsider attacks as they passively monitor for multiple beacons with the same combination of SSID and BSSID. However, SSAD only identifies base variant (MC-MitM-BV) attacks, not improved variant (MC-MitM-IV) attacks because it cannot recognize fake CSAs.

SAE-PK [35] protects only PMF-capable or WPA3 clients using SAE authentication and mainly aims at defending against insider attacks, especially in public Wi-Fi networks. However, MC-MitM attackers can bypass this defense (see Section II-B.1). Also, SAE-PK is not able to distinguish between MC-MitM attack variants.

From an implementation standpoint, all existing defense mechanisms require complex firmware updates or hardware/software integration across all Wi-Fi devices, which is impractical, especially in IoT networks. Finally, none of the existing defense mechanisms are backward compatible with old or obsolete devices.

Contrastingly, our proposed SWIDS framework is a plug-and-play system that passively monitors specific signatures of MC-MitM attacks. It has a very low complexity that can be easily operated by a common user, and can be easily integrated into any Wi-Fi or IoT environment to detect attacks against all kinds of devices in a WPA2/3 network, including PMF-capable devices. Our SWIDS framework can also effectively defend against insider and outsider attacks and different MC-MitM attack variants. In addition, our SWIDS is backward compatible with old or legacy devices and is easy to use, as it does not require any protocol or device modifications on each Wi-Fi client and/or AP. Therefore, from the comparison in Table 5, we can state that our SWIDS outperforms the existing defense mechanisms and is a generalizable defense with improved security against MC-MitM attacks.

H. DISCUSSION ON EXISTING SIGNIFICANT DATASETS

The AWID3 dataset [26] is widely utilized as a publicly available dataset for studying various Wi-Fi attacks. It includes multiple attack traces stored as PCAP files,

TABLE 6. Comparison of performance in identifying/classifying Krack attacks from AWID3 dataset.

Reference	F1 Score	Accuracy	Detection Type
[45] Table 17	-	98.69	ML based
[44] Table 4	88.07	98.73	ML based
[46] Table 1	90.17	90.15	ML based
[43] Table 4	98.51	98.50	ML based
Proposed SWIDS framework	99.08	100	Threshold Based

including instances of KRACK attacks. However, when considering the detection of MC-MitM attacks, the AWID3 dataset can only be used to identify KRACK attacks, which are just one type of MC-MitM enabled attacks. Therefore, the AWID3 dataset is not a generalizable dataset to correctly distinguish all types of MC-MitM attacks. In contrast, as detailed in Section IV-D, we have created our own dataset that includes traffic from the different types of MC-MitM attacks and their variants. This dataset has been used to define our own attack signatures, which have been later used in the experiments described in Section VI-B to evaluate our framework's performance.

We also tested our SWIDS framework using the external AWID3 dataset. To evaluate the performance of our framework in detecting KRACK signatures, we input the AWID3 PCAP file directly into the Traffic Interceptor Unit of our framework (see Figure 17), instead of performing online monitoring or passive capturing. We employed our proposed signatures to detect KRACK behavior in this dataset and we successfully identified the retransmission of message 3 of the 4-way handshake, a behavior that signals the presence of MC-MitM attacks, occurring across multiple channels (channel 2 and 13). Thus, our SWIDS framework effectively detected retransmitted handshake messages in this scenario. In Table 6, we present a performance comparison (F1 Score and/or accuracy, whichever available) of existing detection mechanisms that make use of publicly available AWID3 dataset to identify KRACK attacks.

As we can see from Table 6, the F1 Score and accuracy achieved by our proposed SWIDS framework is higher than in other proposals that utilize the AWID3 dataset. This is because our framework exhibits minimal instances of undetected attack frames. The undetected attacks can be attributed to slight delays in frame processing during the attack detection process. This also demonstrates that our proposed SWIDS framework is adequate to accurately detect the presence of MC-MitM attacks.

We must bear in mind, however, that our proposal is based on real-time detection, whereas the methods reviewed in this section are based on offline analysis of network data using machine learning algorithms. Therefore, the results of our framework are those shown in Section VI-D. Here, we have shown the result of feeding the AWID3 data into our Traffic Interceptor Unit just for comparative purposes.

I. SECURITY CONSIDERATIONS

Our SWIDS framework is the first of its kind to identify MC-MitM attacks and is applicable to all Wi-Fi networks and devices. Since our framework passively monitors Wi-Fi networks, it can identify both insider and outsider threats against any type of Wi-Fi device. Moreover, our framework is difficult for an attacker to circumvent, even if he is aware of the deployed defense mechanisms and algorithms used. This is due to the fact that we defined the thresholds for identifying the appearance of malicious frames as part of the essential operations (stage 1 and stage 2 attack traffic) required for successful MC-MitM attacks, and it is impossible to carry out such attacks without meeting or surpassing those thresholds. Furthermore, even if the attacker devises any other new tactics to deceive the victim besides jamming or CSA attacks as part of stage 1 traffic, the stage 2 traffic remains visible to our SWIDS. Lastly, our framework follows a plug-and-play deployment and does not require any protocol or device modifications on Wi-Fi clients and/or AP. Thus, standard users will be able to set up our proposed defense mechanism with significantly less technical difficulty.

J. LIMITATIONS OF OUR FRAMEWORK

While our SWIDS framework is versatile and applicable to both personal and enterprise networks, it currently monitors a single AP/single Wi-Fi network at a time. This design choice aligns with the nature of MC-MitM attacks, which typically target one AP at a time, focusing on specific SSID, BSSID, and operating channels. As of now, our framework does not support concurrent monitoring of two APs or different channels, such as 2.4 GHz and 5 GHz. We also indicate that our current framework is focused on detecting MC-MitM attacks and does not include prevention capabilities. However, our future work involves addressing these limitations by developing a distributed detection system that will enable multiple detectors to concentrate on different APs with varying channel frequencies, thereby enhancing the framework's detection capabilities.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we highlighted the capabilities and impact of MC-MitM attacks on Wi-Fi networks. We described various challenges posed by MC-MitM attacks regarding effective detection and implementation difficulties of existing defense mechanisms. To this end, we proposed a lightweight signature-based intrusion detection system framework to detect different MC-MitM attack variants. We first classified and investigated network traffic behavior during MC-MitM attacks. We then designed attack signatures and identified useful metrics to detect MC-MitM attacks through various theoretical and empirical analyses of the attack and benign traffic behavior. From these signatures, we created detection algorithms for identifying different MC-MitM attack variants. We then implemented our framework using scapy, a python library for packet capturing and manipulation, and

commercially available wireless interfaces. Our framework is a centralized, passive monitoring system that can be easily integrated with Wi-Fi-based IoT environments. Further, our framework is independent of any Wi-Fi protocols or standards, does not require modifying existing network settings or device modifications, and provides continuous security against MC-MitM attacks

We then evaluated our framework with real MC-MitM attacks in an experimental IoT network setup and specifically analyzed detection performance at different distances. We found that our framework exhibits a minimum TPR of 90% using short-distance detectors and 84% using long-distance detectors with a detection delay of maximum 60 seconds. In addition, we analyzed performance of our framework under various channels and channel bandwidths. We showed that the choice of any specific channel or channel bandwidth does not significantly impact our framework's detection performance. We also showed that our SWIDS framework incurs minimal overhead in terms of CPU and memory usage. These results emphasize the versatility of our detection logic, suggesting its applicability to diverse smart home network contexts.

We also showed that frame loss affects detection performance with long-distance detectors, especially in 2.4 and 5 GHz bands. Based on our evaluation, we plan to extend the present framework to include a distributed and cooperative intrusion detection system to enhance performance in our future works. Specifically, our intention is to deploy this implementation on single-board computers, such as Raspberry Pis, which are commonly used for various smart home applications like Home Assistants or OpenHAB. This approach will not only reduce the cost-effectiveness of our framework but also enable its evaluation in wide-ranging practical Wi-Fi based IoT environments hosting multiple APs.

APPENDIX A
NETWORK ANALYSIS ALGORITHMS

In this Appendix, we briefly discuss various network analysis algorithms and their operations.

A. ALGORITHM 1: CONSTANT JAMMING ANALYSIS

During a probe interval, this algorithm computes: (1) an array of FIAT, where each FIAT is measured from two successive beacons; (2) total number of beacons captured on the legitimate channel of the AP.

Algorithm 1 Constant Jamming Analysis (Detect Constant Jamming Behavior)

```
Data: Wireless traffic
Result: Array of FIAT (A-FIAT), Number of beacons (NB)
while probe-interval do
    Calculate FIAT between two successive beacons; Record each
      FIAT to A-FIAT;
    Count number of beacons(NB);
end
```

B. ALGORITHM 2: MALFORMED FRAME ANALYSIS

This algorithm counts the number of malformed frames due to reactive jamming in a probe interval. This is done by verifying the FCS flags present in the header of the beacon and probe response frames, especially those arriving on the legitimate channel of the AP.

Algorithm 2 Malformed Frame Analysis (Detect Reactive Jamming Behavior)

```
Data: Wireless traffic
Result: Number of malformed frames (MF) AP-MAC=MAC ID of
the AP;
C-CHANNEL=Current channel of the AP;
while probe-interval do
    if frame.haslayer(Dot11) then
        Extract bssid and channel of the frame;
        if bssid == AP-MAC and (frame.haslayer(Dot11Beacon)
          or frame.haslayer(Dot11ProbeResp)) and channel
          ==C-CHANNEL then
            RT = frame.getlayer(RadioTap);
            if RT.Flags == "FCS+badFCS") then
                Count malformed-frame (MF); Store current
                  channel;
            end
        end
    end
end
```

C. ALGORITHM 3: CHANNEL SWITCH ANALYSIS

This algorithm counts the number of beacons, probe responses, or action frames with CSA information elements in the legitimate channel of the AP. Such information elements are extracted from the frames using the tag ID. 37.

Algorithm 3 Channel Switch Analysis (Detect CSAs)

```
Data: Wireless traffic
Result: Number of CSA (CSA) AP-MAC=MAC ID of the AP;
C-CHANNEL=Current channel of the AP;
while probe-interval do
    Extract bssid of the frame;
    if bssid == AP-MAC and (frame.haslayer(Dot11Beacon) or
      frame.haslayer(Dot11ProbeResp) or frame.subtype == 13)
      then
        Extract each Information Element (IE);
        if IE-ID is 37 then
            Count CSA (CSA);
        end
    end
end
```

D. ALGORITHM 4: CONCURRENT BEACON OR PROBE RESPONSE TRAFFIC ANALYSIS

This algorithm simultaneously monitors and counts the beacon or probe response traffic with the targeted SSID and BSSID on the legitimate channel of the AP and those beacon or probe response traffic with the same SSID and BSSID on any other channel (channel hopping) during a probe interval.

Algorithm 4 Concurrent Beacons/Probe Response Traffic Analysis

```
Data: Wireless traffic
Result: Number of concurrent beacons (BCC and BRC)/probe
responses(PCC and PRC)
AP-MAC=MAC ID of the AP, SSID-AP=SSID of the AP;
C-CHANNEL=Current channel of the AP;
while probe-interval do
    Extract ssid,bssid, ad channel of the frame;
    if (frame.haslayer(Dot11Beacon) then
        if bssid == AP-MAC and ssid == SSID-AP and channel
          == C-CHANNEL then
            Count beacon-current-channel(BCC);
        end
        if bssid == AP-MAC and ssid == SSID-AP and (channel
          != C-CHANNEL) then
            Count beacon-rogue-channel(BRC);
        end
    else
        if frame.haslayer(Dot11ProbeResp) then
            if bssid == AP-MAC and ssid == SSID-AP and
              channel == C-CHANNEL then
                Count probe-current-channel(PCC);
            end
            if bssid == AP-MAC and ssid == SSID-AP and
              (channel != C-CHANNEL) then
                Count probe-rogue-channel(PRC);
            end
        end
    end
end
```

E. ALGORITHMS 5, 6, AND 7: CONCURRENT CONNECTION ESTABLISHMENT TRAFFIC ANALYSIS

Similar to algorithm 4, these algorithms simultaneously monitor connection establishment traffic between specific clients and the AP on the legitimate channel and any other channel during a probe interval.

Algorithm 5 Concurrent Authentication Traffic Analysis

```
Data: Wireless Traffic
Result: Number of Concurrent Authentication frames(AUTHCC
and AUTHRC) AP-MAC=MAC ID of the
AP,C-CHANNEL=Current Channel of the AP;
while Probe-interval do
    Extract Smac,dmac,channel of the Frame;
    if frame[Dot11].Type == 0 and frame[Dot11].Subtype == 11
      then
        while Client-Mac in Device database do
            if (smac == AP-MAC and Dmac == Client-Mac) or
              (smac == Client-Mac and Dmac == AP-MAC)
              and Channel == C-CHANNEL then
                Count
                  Authentication-Current-channel(AUTHCC);
            end
            if (smac == AP-MAC and Dmac == Client-Mac) or
              (smac == Client-Mac and Dmac == AP-MAC)
              and Channel != C-CHANNEL) then
                Count Beacon-Rogue-channel(AUTHRC);
            end
        end
    end
end
```

More specifically, algorithm 5 counts concurrent authentication traffic, algorithm 6 counts concurrent association traffic, and algorithm 7 counts concurrent EAPOL traffic. Further, all these algorithms work in parallel and analyse the traffic using the device's source and destination MAC addresses.

Algorithm 6 Concurrent Association Traffic Analysis

```
Data: Wireless traffic
Result: Number of concurrent association frames (ASSOCC and
ASSORC) AP-MAC=MAC ID of the AP,C-CHANNEL=Current
channel of the AP;
while probe-interval do
    Extract smac,dmac,channel of the frame;
    if frame[Dot11].type == 0 and frame[Dot11].subtype == 1
      then
        while client-mac in device database do
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel == C-CHANNEL then
                Count association-current-channel(ASSOCC);
            end
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel != C-CHANNEL) then
                Count association-rogue-channel(ASSORC);
            end
        end
    end
end
```

Algorithm 7 Concurrent EAPOL Traffic Analysis

```
Data: Wireless traffic
Result: Number of EAPOL frames (EAPOLCC and EAPOLRC)
AP-MAC=MAC ID of the AP,C-CHANNEL=Current channel of
the AP;
while probe-interval do
    Extract smac,dmac,channel of the frame;
    if frame.haslayer(EAPOL) and (frame[Dot11].type != 1 then
        while client-mac in device database do
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel == C-CHANNEL then
                Count EAPOL-current-channel(EAPOLCC);
            end
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel != C-CHANNEL) then
                Count EAPOL-rogue-channel(EAPOLRC);
            end
        end
    end
end
```

F. ALGORITHMS 8: CONCURRENT DATA TRAFFIC ANALYSIS

This algorithm monitors and counts concurrent data traffic following the concurrent connection establishment traffic.

Algorithm 8 Concurrent Data Traffic Analysis

```
Data: Wireless traffic
Result: Number of data frames (DATACC and DATARC)
AP-MAC=MAC ID of the AP,C-CHANNEL=Current channel of
  the AP;
while probe-interval do
    Extract smac,dmac,channel of the frame;
    if frame[Dot11].subtype == 32 and frame[Dot11].subtype
      == 40 then
        while client-mac in device database do
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel == C-CHANNEL then
                Count data-current-channel(DATACC);
            end
            if (smac == AP-MAC and dmac == client-mac) or
              (smac == client-mac and dmac == AP-MAC) and
              channel != C-CHANNEL) then
                Count data-rogue-channel(DATARC);
            end
        end
    end
end
```

G. ALGORITHM 9: MC-MITM STAGE 1 ATTACK TRAFFIC COLLATOR

At the end of the first sub-probe interval, this algorithm: (1) calculates the overall FIAT from the standard deviation of the FIAT values and the FDR from the number of beacons received during the probe interval, as provided by algorithm 1; (2) calculates malformed rate (MF-rate) from the number of malformed frames provided by algorithm 2 and (3) obtains the number of CSAs from algorithm 3. Based on the threshold values (see Table 2) of these stage 1 attack traffic, algorithm 4 determines whether the stage 1 attack traffic is dubious or not.

Algorithm 9 MC-MitM Stage 1 Attack Traffic Collator

```
Data: Output of Algorithms 1,2 and 3
Result: Status of stage 1 attack traffic
FIAT =SD(A-FIAT),FDR=(NB/600)*100, MF-rate=
  (MF/60)*100;
if (FIAT≤ TH1 and FDR≤ TH2 and MF− rate≤ TH3 and CSA <
  TH4) then
    STAGE-1-ATTACK-TRAFFIC = False;
else
    if (FIAT≥ TH1 and FDR≥ TH2)) then
        CONST-JAM-ATTACK = True;
        LOG as "Intentional jamming attack";
    end
    if (MF− rate≥ TH3) then
        REACTIVE-JAM-ATTACK = True;
        LOG as "Intentional jamming attack";
    end
    if ((CSA≥ TH4) then
        CSA-ATTACK = True;
        LOG as "CSA attack";
    end
end
```

H. ALGORITHM 10: MC-MITM STAGE 2 ATTACK TRAFFIC COLLATOR

This algorithm determines the status of the stage 2 attack traffic at the end of every probe interval based on threshold values (see Table 2).

Algorithm 10 MC-MitM Stage 2 Attack Traffic Collator

```
Data: Output of Algorithms 4, 5, 6, 7 and 8
Result: Status of Stage 2 Attack Traffic
if (BRC == 0 and AUTHRC == 0 and ASSORC == 0 and
  EAPOLRC == 0 and DATARC == 0) then
    STAGE-2-ATTACK-TRAFFIC = False;
else
    if (BCC≥ TH5 and BRC≥ TH5 and PCC≥ TH5 and PRC≥
      TH5) then
        CON-BEACON-PROBE = True;
    end
    if (AUTHCC≥ TH6 and AUTHRC≥ TH6 or ASSOCC≥ TH6
      and ASSORC TH6 or EAPOLCC TH7 and EAPOLRC TH7)
      then
        CON-CONNECTION-EST = True
    end
    if (DATACC TH8 and DATARC TH8) then
        CON-DATA = True
    end
    if (CON-BEACON-PROBE = True and
      (CON-CONNECTION-EST = True or CON-DATA = True))
      then
        STAGE-2-ATTACK-TRAFFIC = True;
    else
        STAGE-2-ATTACK-TRAFFIC = False;
    end
end
```

I. ALGORITHM 11: ALARM GENERATION

Based on the status of stage 1 and stage 2 attack traffic provided by algorithms 9 and 10, algorithm 11 predicts the presence of MC-MitM attacks and variants.

Algorithm 11 Alarm Generation

```
Data: Output of Algorithm 9 and 10
Result: Alarms
if (STAGE-1-ATTACK-TRAFFIC = False and
  STAGE-2-ATTACK-TRAFFIC = False) then
    LOG as "No MC-MitM attack found"
end
if (STAGE-1-ATTACK-TRAFFIC = True and
  STAGE-2-ATTACK-TRAFFIC =True) then
    if (CONST-JAM = True and STAGE 2 ATTACK TRAFFIC =
      True) then
        Raise Alarm;
        LOG as "MC-MitM-BVC attack";
    end
    if (REACT-JAM = True and STAGE 2 ATTACK TRAFFIC =
      True then
        Raise Alarm;
        LOG as "MC-MitM-BVR attack";
    end
    if (CSA-ATTACK = True and STAGE 2 ATTACK TRAFFIC =
      True then
        Raise Alarm;
        LOG as "MC-MitM-IV attack";
    end
else
    if (STAGE-1-ATTACK-TRAFFIC = False and
      STAGE-2-ATTACK-TRAFFIC = True) then
        Raise Alarm;
        LOG as "MC-MitM-attack";
        LOG as "Attack variant unidentified";
    end
end
```

ACKNOWLEDGMENT

The authors wish to extend their special thanks to Mathy Vanhoef for providing source codes and helping in resolving some issues related to MC-MitM attacks.

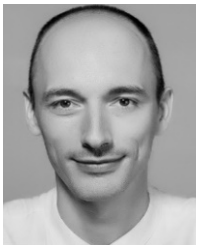
REFERENCES

[1] D. A. D. Zovi and S. A. Macaulay, "Attacking automatic wireless network selection," in *Proc. 6th Annu. IEEE Syst., Man Cybern. (SMC) Inf. Assurance Workshop*, Apr. 2005, pp. 365–372.
[2] B. Fajar. (2021). *Fluxion Kali Linux Tutorial*. [Online]. Available: https://linuxhint.com/fluxion-kali-linux-tutorial
[3] KaliTut. (2021). *Wifiphisher Evil Twin Attack*. [Online]. Available: https://kalitut.com/Wifiphisher-evil-twin-attack
[4] KaliTut. (2021). *WiFi Pumpkin Framework for Rogue WiFi Access Point Attack*. [Online]. Available: https://kalitut.com/wifi-pumpkin-framework-for-rogue-wi-fi
[5] Thecybersecurityman. (2018). *PenTest Edition: Creating an Evil Twin or Fake Access Point on Your Home Network Using Aircrack-NG and Dnsmasq*. [Online]. Available: https://thecybersecurityman.com/2018/08/11/pentest-edition-creating-an-evil-twin-or-fake-access-point-using-aircrack-ng-and-dnsmasq-part-1-setup
[6] M. Vanhoef and F. Piessens, "Advanced Wi-Fi attacks using commodity hardware," in *Proc. 30th Annu. Comput. Secur. Appl. Conf.*, Dec. 2014, pp. 256–265.
[7] M. Vanhoef and F. Piessens, "Release the kraken: New KRACKs in the 802.11 standard," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 299–314.
[8] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1313–1328.
[9] M. Vanhoef, "Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation," in *Proc. 30th USENIX Secur. Symp. (USENIX Security)*, 2021, pp. 161–178.
[10] J. Freudenreich, J. Weidman, and J. Grossklags, "Responding to KRACK: Wi-Fi security awareness in private households," in *Human Aspects of Information Security and Assurance*. Cham, Switzerland: Springer, 2020, pp. 233–243.
[11] Security Focus. (2019). *WPA2 Key Reinstallation Multiple Security Weaknesses*. [Online]. Available: https://www.securityfocus.com/bid/101274
[12] M. Chi et al., "Multi-channel man-in-the-middle attack against communication-based train control systems: Attack implementation and impact," in *Lecture Notes in Electrical Engineering*. Singapore: Springer, 2020, pp. 129–139.
[13] M. Thankappan, H. Rifà-Pous, and C. Garrigues, "Multi-channel man-in-the-middle attacks against protected Wi-Fi networks: A state of the art review," *Expert Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118401.
[14] S. Nikbakhsh, A. B. A. Manaf, M. Zamani, and M. Janbeglou, "A novel approach for rogue access point detection on the client-side," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 684–687, doi: 10.1109/WAINA.2012.108.
[15] L. Wang and A. M. Wyglinski, "Detection of man-in-the-middle attacks using physical layer wireless security techniques," *Wireless Commun. Mobile Comput.*, vol. 16, no. 4, pp. 408–426, Mar. 2016.
[16] M. Vanhoef, P. Adhikari, and C. Pöpper, "Protecting Wi-Fi beacons from outsider forgeries," in *Proc. 13th ACM Conf. Security Privacy Wireless Mobile Netw.*, 2020, pp. 155–160.
[17] C. Louca, A. Peratikou, and S. Stavrou, "802.11 man-in-the-middle attack using channel switch announcement constantinos," in *Proc. 12th Int. Netw. Conf.* Cham, Switzerland: Springer, 2021, pp. 62–70.
[18] C. Louca, A. Peratikou, and S. Stavrou, "On the detection of channel switch announcement attack in 802.11 networks," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Jul. 2021, pp. 281–285.
[19] S. Gong, H. Ochiai, and H. Esaki, "Scan-based self anomaly detection: Client-side mitigation of channel-based man-in-the-middle attacks against Wi-Fi," in *Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2020, pp. 1498–1503.
[20] S. Burke. (2018). *Wi-Fi Alliance Introduces Security Enhancements*. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-security-enhancements
[21] Philipp Ebbecke (Wi-Fi Alliance). (2020). *Protected Management Frames Enhance Wi-Fi Network Security*. [Online]. Available: https://www.wi-fi.org/beacon/philipp-ebbecke/protected-management-frames-enhance-wi-fi-network-security
[22] B. Bertka, "802.11w security? DoS attacks and vulnerability controls," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012.
[23] M. Vanhoef. (2021). *FragAttacks: Clarifying Some Aspects*. Accessed: May 10, 2023. [Online]. Available: https://www.mathyvanhoef.com/2021/05/fragattacks-clarifying-some-aspects.html
[24] CWNP. (2009). *Wireless LAN Security and IEEE 802.11w*. [Online]. Available: https://www.cwnp.com/wireless-lan-security-and-ieee-802-11w
[25] MTROI. (2021). *Protected Management Frames (802.11w)*. [Online]. Available: https://wlan1nde.wordpress.com/2014/10/21/protected-management-frames-802-11w
[26] E. Chatzoglou, G. Kambourakis, and C. Kolias, "Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset," *IEEE Access*, vol. 9, pp. 34188–34205, 2021.
[27] M. Thankappan. (2023). *Signature-Based-WIDS-for-Detecting-MC-MitM-Attacks*. [Online]. Available: https://github.com/maneshthankappan/Signature-Based-WIDS-for-detecting-MC-MitM-attacks
[28] W. J. Tom Van Goethem, M. Vanhoef, and F. Piessens, "Request and conquer: Exposing cross-origin resource size," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 447–462.
[29] M. Vanhoef and F. Piessens, "Predicting, decrypting, and abusing WPA2/802.11 group keys," in *Proc. 25th USENIX Secur. Symp. (USENIX Assoc.)*, 2016, pp. 673–688.
[30] L. F. Epia Realpe, O. J. S. Parra, and J. B. Velandia, "Use of KRACK attack to obtain sensitive information," in *Proc. Int. Conf. Mobile, Secure, Program. Netw.*, in Lecture Notes in Computer Science, vol. 11005, 2019, pp. 270–276.
[31] E. Tews and M. Beck, "Practical attacks against WEP and WPA," in *Proc. 2nd ACM Conf. Wireless Netw. Secur.*, Mar. 2009, pp. 79–85.
[32] J. Selvi, "Bypassing HTTP strict transport security," 2014. [Online]. Available: https://www.blackhat.com/docs/eu-14/materials/eu-14-Selvi-Bypassing-HTTPStrict-Transport-Security-wp.pdf
[33] C. Matte, J. P. Achara, and M. Cunche, "Device-to-identity linking attack using targeted Wi-Fi geolocation spoofing," in *Proc. 8th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2015, pp. 1–6.
[34] M. Vanhoef, N. Bhandaru, T. Derham, I. Ouzieli, and F. Piessens, "Operating channel validation: Preventing multi-channel man-in-the-middle attacks against protected Wi-Fi networks," in *Proc. 11th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jun. 2018, pp. 34–39.
[35] Wi-Fi Alliance. (2020). *SAE Public Key*. [Online]. Available: https://www.wi-fi.org/beacon/thomas-derham-nehru-bhandaru/wi-fi-certified-wpa3-december-2020-update-brings-new-0
[36] Huawei. (2020). *Wireless Access Controller Configuration Guide*. [Online]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1100008282/b27702df/understanding-WLAN-security-policies
[37] U. Chatterjee, R. Sadhukhan, D. Mukhopadhyay, R. Subhra Chakraborty, D. Mahata, and M. M. Prabhu, "Stupify: A hardware countermeasure of KRACKs in WPA2 using physically unclonable functions," in *Proc. Companion Web Conf.*, Apr. 2020, pp. 217–221.
[38] A. Babaei and G. Schiele, "Physical unclonable functions in the Internet of Things: State of the art and open challenges," *Sensors*, vol. 19, no. 14, p. 3208, Jul. 2019.
[39] T. Chin and K. Xiong, "KrackCover: A wireless security framework for covering KRACK attacks," in *Wireless Algorithms, Systems, and Applications*, vol. 10874. Cham, Switzerland: Springer, 2018.
[40] Y. Li, M. Serrano, T. Chin, K. Xiong, and J. Lin, "A software-defined networking-based detection and mitigation approach against Krack," in *Proc. 16th Int. Joint Conf. E-Business Telecommun.*, 2019, pp. 244–251.
[41] T. Naitik, L. Raiton, V. Pradnya, and S. Vamshi, "Mitigation of key reinstallation attack in WPA2 Wi-Fi networks by detection of nonce reuse," *Int. Res. J. Eng. Technol.*, vol. 5, no. 5, pp 1528–1531, 2018.
[42] Securingsam. (2017). *KRACK Detector*. [Online]. Available: https://github.com/securingsam/krackdetector.
[43] A. Agrawal, U. Chatterjee, and R. R. Maiti, "CheckShake: Passively detecting anomaly in Wi-Fi security handshake using gradient boosting based ensemble learning," *IEEE Trans. Dependable Secure Comput.*, pp. 1–13, 2023.
[44] E. Chatzoglou, G. Kambourakis, C. Smiliotopoulos, and C. Kolias, "Best of both worlds: Detecting application layer attacks through 802.11 and non-802.11 features," *Sensors*, vol. 22, no. 15, p. 5633, Jul. 2022.
[45] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101752.

[46] A. Agrawal, U. Chatterjee, and R. R. Maiti, "KTRACKER: Passively tracking Krack using ML model," in *Proc. 12th ACM Conf. Data Appl. Secur. Privacy*, Apr. 2022, pp. 364–366.
[47] G. Abare and E. J. Garba, "A proposed model for enhanced security against key reinstallation attack on wireless networks," *Int. J. Sci. Res. Netw. Secur. Commun.*, vol. 7, no. 3, pp. 21–27, 2019.
[48] R. R. Singh, J. Moreira, T. Chothia, and M. D. Ryan, "Modelling of 802.11 4-way handshake attacks and analysis of security properties," in *Security and Trust Management*. Cham, Switzerland: Springer, 2020, pp. 3–21.
[49] C. Cremers, B. Kiesl, and N. Medinger, "A formal analysis of IEEE 802.11's WPA2: Countering the kracks caused by cracking the counters," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1–17.
[50] SNORT. (2018). *Policy-Other WPA2 Key Reuse Tool Attempt*. [Online]. Available: https://www.snort.org/rule_docs/1-44640
[51] D. Schepers, M. Vanhoef, and A. Ranganathan, "A framework to test and fuzz Wi-Fi devices," in *Proc. 14th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jun. 2021, pp. 368–370.
[52] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard 802.11ad-2012, 2012.
[53] *Broadband Radio Access Networks (BRAN); 5 GHz High Performance RLAN; Harmonized EN Covering the Essential Requirements of Article 3.2 of the R Directive, V1.8.1*, ETSI EN 301893, Mar. 2015. [Online]. Available: https://www.etsi.org/deliver/etsi_en/301800_301899/301893/01.07.00_40/en_301893v010700o.pdf
[54] M. Vanhoef. (2015). *Advanced Wi-Fi Attacks Using Commodity Hardware*. [Online]. Available: https://github.com/vanhoefm/modwifi#constant-jamming
[55] L. Woody. (2018). *Mitm-Channel-Based-Package*. [Online]. Available: https://pypi.org/project/mitm-channel-based.
[56] O. Osanaiye, A. Alfa, and G. Hancke, "A statistical approach to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 18, no. 6, p. 1691, May 2018.
[57] O. Punal, I. Aktas, C. J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jul. 2014, pp. 1–10.
[58] *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 Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE Standard 802.11ax, May 2021.
[59] Ubuntu. (2005). *Hostapd_Cli*. [Online]. Available: http://manpages.ubuntu.com/manpages/bionic/man1/hostapd_cli.1.html
[60] M. Thankappan. (2023). *MC-MitM Attack Signatures*. [Online]. Available: https://github.com/maneshthankappan/-MC-MitM-Attack-Dataset
[61] W. Zhou, A. Marshall, and Q. Gu, "A sliding window based management traffic clustering algorithm for 802.11 WLAN intrusion detection," in *Proc. Int. Fed. Inf. Process.*, 2006, p. 213.
[62] InscapeData. (2011). *Introduction To 802.11n Outdoor Wireless Networks*. [Online]. Available: https://www.inscapedata.com/pdf/80211n Technology.pdf
[63] J. Dugan. (2020). *What is IPerf/IPerf3*. [Online]. Available: https://iperf.fr
[64] V. Sathya, M. I. Rochman, and M. Ghosh, "Measurement-based coexistence studies of LAA & Wi-Fi deployments in Chicago," *IEEE Wireless Commun.*, vol. 28, no. 1, pp. 136–143, Feb. 2021.
[65] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 197–216, 1st Quart., 2019.

MANESH THANKAPPAN (Member, IEEE) received the B.Tech. degree in information technology from Mahatma Gandhi University, Kerala, India, in 2006, and the M.Tech. degree in information security from the National Institute of Technology Karnataka, Surathkal, India, in 2011. He is currently pursuing the Ph.D. degree in networks and information technologies with Universitat Oberta de Catalunya (UOC), Spain, under the supervision of Dr. Helena Rifà-Pous and Dr. Carles Garrigues. From February 2006 to October 2012, he was with the Faculty of Computer Science and Engineering, Adi Shankra Institute of Engineering and Technology (ASIET), Cochin, India. From October 2012 to October 2018, he was with the Department of Computer Science, Prince Sattam Bin Abdulaziz University (PSAU), Saudi Arabia, as a Lecturer. He is a member of the K-riptography and Information Security for Open Networks (KISON) Research Group, UOC. His research interests include cybersecurity and network forensics, with a special focus on the security of wireless networks and the IoT systems.

HELENA RIFÀ-POUS (Member, IEEE) received the Ph.D. degree from Universitat Politècnica de Catalunya, in 2008. Since 2007, she has been an Associate Professor with the Department of Computer Science, Universitat Oberta de Catalunya (UOC). She is also a Coordinator of the M.Sc. Cybersecurity and Privacy Course, UOC, and conducts her research within the K-riptography and Information Security for Open Networks (KISON) Research Group, UOC. She has authored numerous articles in journals and conferences. Her research interests include security and privacy protocols, with a special interest in distributed and wireless networks, such as smart homes and the IoT. She participates as a reviewer for several journals and also serves as an editor.



CARLES GARRIGUES received the Ph.D. degree from Universitat Autònoma de Barcelona, in 2008, and the research accreditation degree from the Catalan Quality Agency (AQU), in 2018. He has two recognized research periods from AQU. He is currently an Associate Professor with the Department of Computer Science, Universitat Oberta de Catalunya. He also conducts his research with the K-riptography and Information Security for Open Networks (KISON) Research Group, UOC. In terms of scientific production, he has authored several publications indexed in the ISI JCR. He has published numerous articles at national and international congresses. His research interests include computer security and privacy, with a special focus on security in smart cities, smart homes, and the IoT environments in general. He participates as a reviewer for several scientific journals. He has also served on the program committee of several conferences.