Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems

Naureen Hoque, Hanif Rahbari, and Cullen Rezendes
ESL Global Cybersecurity Institute, Rochester Institute of Technology, Rochester, NY
{naureen.hoque, rahbari, cxr5971}@mail.rit.edu

***Abstract*—To establish a secure Wi-Fi connection, several unprotected management frames are exchanged between an access point and a station before they mutually authenticate each other and start a protected session. In this paper, we are the first to formally model and analyze this connection establishment phase, based on the latest IEEE 802.11 standard, and accordingly, expose a new denial of service (DoS) vulnerability and three new variants of a known man-in-the-middle (MitM) attack. We also formally show that the optional operating channel validation technique introduced in the latest standard is capable of protecting the system only against multi-channel MitM. To validate our identified DoS vulnerability, we test it against the latest *wpa_supplicant* daemon, showing that an adversary can stealthily prevent a station from connecting to a preferred AP for up to 90 minutes, likely more. We also propose a mitigation approach to counter it.**

***Index Terms*—Wi-Fi security, denial of service, formal analysis**

I. Introduction

From 2014 to 2021, a series of compound attacks successfully targeted widely deployed wireless local area networks (WLANs) even when those networks were protected by the latest Wi-Fi security protocols [1]–[7]. In most cases, these attacks were initiated by first exploiting a *pre-authentication vulnerability* during the connection initiation (before the completion of the mutual authentication phase). A pre-authentication exploit can enable an adversary to next launch an elaborated protocol attack to subsequently decrypt (data) packets, replay them, or in certain cases, retrieve the authentication key [2], [3]. The pre-authentication phase is vulnerable because the security protocols WPA2, WPA3, and IEEE 802.11w (for management frames), are able to secure frames only at the MAC layer and only after a pairwise transient key is correctly installed following a successful mutual authentication; leaving the frame exchanges, operating channels, and training signals used prior to that point largely unprotected.

Deceiving a station into connecting to a relay or a man-in-the-middle (MitM) is among the known attacks in the pre-authentication phase [1]–[3]. Offering a higher signal strength on a different *channel*, abusing the unprotected *channel* switching mechanism, and jamming the *channel* of the real access point (AP) during this phase are common methods an attacker can employ to deceive the station. The adversary then creates an MitM position to connect to the real AP (on the original channel) on one side, and to the station on the new channel on the other side, so as to selectively delay, block, or alter certain management frames. This *multi-channel* MitM position can then be used in the next steps of the attack to, for instance, force the station to reset the nonce used in the AES algorithm, as demonstrated in key reinstallation attacks (KRACKs) [2].

This and other pre-authentication vulnerabilities are present not only in traditional *personal* and *enterprise* WLANs, but, as we argue in this paper, also in the growing interoperable *public* Wi-Fi networks, such as Passpoint® [8], eduroam [9], OpenRoaming™ [10], WiFi4EU initiative [11], and others, because their architecture and connection establishment process are essentially the same as in the enterprise mode [9], [12]. It is particularly important as the enterprise WLAN market has reached $7.6 billion worldwide in 2021 [13] and Wi-Fi Passpoint® is increasingly being adopted in high-density public venues like airports, convention centers, and stadiums [14].

The key reinstallation vulnerability of the WPA2 protocol was further confirmed using formal analysis in [15], where it was also proven that once the nonce reset issue is fixed, WPA2 has no more vulnerabilities. We contend that if the WPA2 protocol had been formally verified before its release, then KRACKs would not have surprised consumers after more than a decade of use. Nevertheless, the analysis in [15] covered only the mutual authentication (4-way handshake) phase, it did not capture potential spoofing or jamming (e.g., channel/training signal) attacks outside of what WPA2 is supposed to protect, including the root cause of the KRACKs—the multi-channel MitM.

In this paper, we fill one of the gaps in the formal analysis of the Wi-Fi protocol by formally modeling and analyzing the pre-authentication phase per the IEEE 802.11-2020 rollup [16], which will allow us to avoid future huge-impact vulnerabilities such as KRACKs. The pre-authentication phase is known to be insecure and yet, to the best of our knowledge, we are the first to perform systematic formal verification on it. It is critical to perform this analysis and ensure we evaluate all edge cases before taking steps to properly secure the connection initiation phase[1] at both the physical (PHY) and MAC layers. We also formally analyze the operating channel validation (OCV) technique, a new (optional) mechanism recently (in 2020) added to the standard to protect only the operating channel element with the goal of preventing a multi-channel MitM [16]. We validate our novel finding from our analysis, a new denial-of-service (DoS) vulnerability, using the *Wi-Fi Framework* [17].

***Challenges*—**To carry out this research, we needed to first read and interpret hundreds of pages of the standard to model the pre-authentication phase as accurately as possible despite being ambiguous in many cases. Second, we had to account

[1]We use connection establishment, connection initiation, and pre-authentication phase interchangeably in this paper.

for certain values or a function of certain parameters (e.g., the frame retransmission limit) that the standard does not specify. The unspecified values/parameters create inconsistencies across different implementations and make it difficult to model the protocol accurately. Likewise, there are some MAC header fields that the standard explicitly defines as vendor-specific or leaves them as an *implementation decision* but without any warnings about their ranges or (possible) corner cases, hoping that the developers/vendors would take care of them.

Contributions—Our main contributions are as follows:

- We present the first formal symbolic model of Wi-Fi's connection establishment phase capturing different modes of WPA2/WPA3 as well as public interoperable networks.
- Our formal analysis exposes three variants of multi-channel MitM attacks including the classic one (that is already disclosed in [1]) and a new DoS vulnerability in the 802.11 standard. We further formally and empirically analyze the system with the OCV technique in place and show that it is unable to fully protect the system against all types of MitM and also increases the overall latency.
- We validate our identified vulnerability against the latest version of *wpa_supplicant* daemon, widely used along with *hostapd* on Linux and (with modifications) Android. Specifically, we show that our DoS attack can stealthily prevent a station from successfully connecting to a preferred AP for up to 5370 seconds, with a 300-second delay after every additional failure. It is enormously higher than a regular Wi-Fi connection establishment delay. Additionally, we propose a mitigation technique that provides a more secure alternative to the current system.

We publicly released our formal analysis and testbed validation codes for the community at *https://github.com/hoquenaureen/wifi-preauthvul-analyze* after we responsibly disclosed it to the Wi-Fi Alliance.

Paper Organization—We provide a primer on Wi-Fi frames and connection initiation in Section II, and a description of our system and adversary models in Section III. Next, we provide the formal verification of the current systems and OCV technique in Sections IV and V, respectively. In Section VI, we provide the results of our experiments and discuss a potential countermeasure. We discuss related work in Section VII before concluding the paper in Section VIII.

II. Background

We start by reviewing the Wi-Fi connection establishment phase, key attributes of Wi-Fi frames, and multi-channel MitM.

A. Connection Establishment in Personal & Enterprise Wi-Fi

An enterprise WLAN essentially consists of an authentication server, APs, and potentially a large number of stations. The server stores the user credentials and is responsible for authenticating the users and eventually generating a pairwise master key (PMK) for each AP-station pair. As such, a station that seeks a secure Internet connection needs to first talk to the server via an AP. Fig. 1 depicts the steps in more details. The steps in the personal mode exclude step ③ as there is no server, and the PMK is generated based on a shared passphrase.

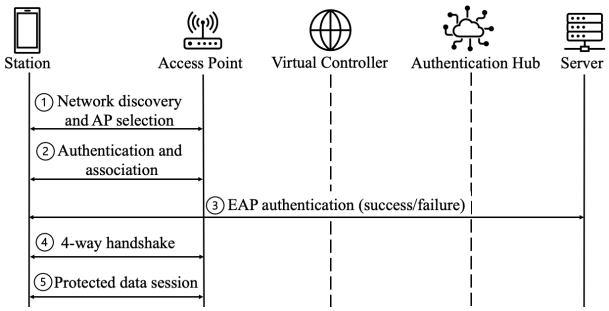

Fig. 1: Steps of connection establishment in Wi-Fi. EAP authentication (step 3) takes place only in enterprise and interoperable public Wi-Fi networks (e.g., Passpoint® fasciliatted by a hub).

1) Connecting to a nearby AP: IEEE standard 802.11-2020 defines a network discovery protocol in which a number of (unauthenticated) management frames are exchanged between an AP and a station to create an initial (unsecured) connection [16]. In active scanning mode, a station scans the network by periodically broadcasting *probe requests*. An AP may reply with a *probe response*. In passive scanning mode, an AP periodically broadcasts *beacon* frames to announce its presence to the nearby devices. When multiple APs are present, a station always prefers an AP with the highest received signal strength measured using those frames' training (preamble) signals [16, § 17.3.12]. After receiving either a probe response or a beacon, the station proceeds to the authentication and association steps with its preferred AP (step ②). The authentication step involves transmitting an *authentication request* to the AP, which is open (void) in WPA2 and WPA3-Enterprise, and the AP replies with an *authentication response*. Under WPA3-Personal, this step instead consists of two Dragonfly handshake message exchanges, *Commit* and *Confirm*. Next, the station sends an *association request* and receives an *association response*. In case an associated station wants to leave an AP (e.g., when roaming to another AP), it sends a deauthentication/disassociation notification to the last AP. An AP immediately disassociates a station upon receiving that station's deauthentication/disassociation notification [16, § 4.5].

2) IEEE 802.1X/EAP: Once the station is connected to an AP, an extensible authentication protocol (EAP) method along with IEEE 802.1X are used in enterprise and public modes for mutual authentication between the station and an authentication server (step ③ in Fig. 1). The main goal of EAP is to securely pass authentication information between a station and the server, where the AP (together with the controller and the hub in public networks) is just a bridge between them. The AP moves to the next step only if the server notifies it that the user is verified. The PMK is derived at the end of this process using a master session key (MSK), which is sent to the station via the AP using one of the *EAP* frames.

3) 4-way Handshake: Once the PMK is derived at both the AP and station, they perform the 4-way handshake in step ④ to mutually authenticate each other and derive a pairwise transient key (PTK) based on the (supposedly common) PMK to protect the imminent data (and when 802.11w is enabled, management) frames. Once the handshake succeeds, they install this new PTK and start a protected session (step ⑤).

B. Interoperable Public Wi-Fi Networks

Connecting to an AP follows similar steps in emerging public Wi-Fi networks as they require to use the same 802.1X/EAP for user authentication—see Fig. 1. That is in contrast to the traditional open Wi-Fi networks that offer no security at all, or Wi-Fi Enhanced Open™ that provides only unauthenticated data encryption for public networks [18]. Recent approaches for public Wi-Fi, such as, Passpoint® (a.k.a. Hotspot 2.0) by Wi-Fi Alliance [8], eduroam [19], OpenRoaming™ by Wireless Broadband Alliance (WBA) [10], and Open-Architecture based Wi-Fi Access Network Interface (WANI) by Telecom Regulatory Authority of India [20], deploy an authentication server to enable mutual authentication between (a large number of) stations and the server through an authentication hub. This architecture allows a station to go through the same connection establishment steps as in Wi-Fi enterprise [9], [12], [19]–[21].

C. Relevant Wi-Fi Frame Attributes

Frame Timeout & Retransmissions: To account for possible transmission failures, IEEE 802.11-2020 defines a timeout interval for every frame and, in turn, allows retransmission of a lost or corrupted frame after its expiration [16, § 9-10]. This interval accounts for processing delays, inter-frame spacing and slot time, etc., and ranges from $\sim 0.045 - 20$ ms depending on the frame type (e.g., management, control). Multiple retransmissions are allowed within a predefined *retry limit* until the frame is successfully received, but the standard does not specify the retransmission limit [16, § 10].

Channel Switch Announcement: An AP may change its operating channel in the middle of the connection establishment for various reasons, such as, the current operating channel is congested, has poor quality, or needs to be vacated for a weather radar in proximity (specific to 5 GHz). It uses the channel switching announcement (CSA) element, which can be sent within a beacon *anytime* during this phase, to advertise a new channel and when it intends to switch the channel.

D. Multi-channel MitM

In a protected Wi-Fi system, it is not trivial to establish an MitM position with an *arbitrary* MAC address as the station verifies the integrity of the AP's MAC address. It is not trivial prior to the start of the protected session either as the PTK is derived based on the MAC address of the real AP and the 4-way handshake will fail if the rogue AP uses a different one. If an adversary wants to instead use the same MAC address of the real AP on the same channel, its activities can be detected by the real AP. Therefore, a rogue AP with the same MAC address needs to be on a channel other than the real AP's, requiring a multi-channel MitM. To elaborate, assume that a real AP is operating on channel x. The adversary installs a rogue AP with the same MAC address, but on channel y, where $x \neq y$. Then, a rogue AP can force a station to connect to it on channel y either by spoofing a beacon frame with a CSA element on channel x (classic CSA-based MitM [2]), by offering higher signal strength on channel y, or by jamming the real AP [3]. An adversary as an MitM cannot decrypt the frames; its main goal is to monitor and manipulate the transmission of those frames.

E. Operating Channel Validation

To counter multi-channel MitM attacks, one optional MAC layer mechanism, OCV, has been added to the IEEE 802.11-2020 standard to protect CSA elements [16, § 12.2]. This technique mandates an authenticated operating channel information (OCI) element in a beacon or probe response frame to specifically protect a CSA, preventing CSA-based MitM attacks [22]. OCV also requires additional frames to be exchanged, named security association (SA) Query, for every unprotected channel switch to confirm the reception of the CSA element by all of the associated stations. To avoid any query collision, OCV sets a random delay. According to the standard:
"If the STA chooses to perform the specified switch [...] and the AP has indicated OCVC capability, after switching to the new channel the STA shall wait a random delay [...] and then initiate the SA query procedure once any applicable conditions for transmitting on the new channel are met."

III. System and Threat Model

System Model: We consider a WLAN configured based on the IEEE 802.11-2020 standard rollup [16] and secured with WPA3 (or 802.11w-enabled WPA2) protocol in either personal or enterprise mode, or with Passpoint® or OpenRoaming™ in public mode. We note that the 802.11ax amendment [23] to the 802.11-2020 standard has *not* amended any of the functions related to this paper. We further assume that while an AP is in a connection establishment phase with one station, it can continue broadcasting periodic beacons and connect with other stations. We also consider that at least three channels are available to the devices in the system, denoted by x, y, and z.

Adversary Model: We consider the powerful *Dolev-Yao* adversary model [24], commonly used for formally analyzing the security of network systems (e.g., in [15], [25]) and proving specific properties of their protocols. Under this model, an adversary can eavesdrop, drop, inject, and modify any message, but it cannot encrypt/decrypt messages or guess the secret key used by the underlying security protocol. In the context of our paper, this adversary has the following capabilities:

The adversary, potentially at the MitM position, can eavesdrop, jam/drop, spoof/inject, and modify the legitimate AP's pre-authentication frames, but cannot decrypt the communications between that AP and the server or physically tamper with a real AP (or station). It cannot be a part of the server's trusted network either (i.e., not an insider). In addition, the adversary has unlimited resources to create a fake AP with its desired MAC address. Both APs (real and fake) can be active at the same time, but potentially on different channels since having two entities with the same MAC addresses on the same channel will likely be easily detected. The adversary cannot obtain or guess the user password of a station.

Goal. The adversary's main goal is to alter or spoof pre-authentication frame(s) to launch an attack (e.g., MitM, or other means as a basis for launching other advanced attacks).

IV. Formal Analysis: Connection Establishment in Wi-Fi

In this section, we formally model and analyze the connection establishment phase of a Wi-Fi system. We take a

similar symbolic model checking approach to [25], where the attach, detach, and paging processes of 4G/LTE systems are formally analyzed. However, we do not apply any cryptographic verification since the Wi-Fi pre-authentication phase does not involve any cryptography except during the 4-way handshake, and the security of this handshake has already been cryptographically verified [15]. Therefore, we just need a model checker (MC) to specifically model the connection establishment phase that involves, among other things, possible operating channel switches. Model checking is an appropriate choice for such a system, specifically for modeling the sequence of frame exchanges and inspecting whether that model satisfies the temporal traces defined by the standard.

A. Ambiguities in the Standard Specifications

The descriptions of the standard specifications are sometimes ambiguous, and that makes it challenging to reliably interpret and accurately model the standard. In the following we discuss a few of those ambiguities with their consequences:

First, the standard does not specify the value (or an equation) of certain parameters. For example, the standard clearly refers to retransmission and its limit for error recovery, but not to their exact values [16, § 10.3]. As can be seen below, the standard does not even state if these values should be determined by a vendor, developer, or any specific algorithm.

> *"Error recovery shall be attempted by retrying transmissions for frame exchange sequences that the initiating STA infers have failed [...] until the transmission is successful, or until the relevant retry limit."*

This unspecified instruction has resulted in various interpretations by different ends, such as, Xin *et al.* in their work [26], referring to a discrete-event network simulator (*ns3*) documentation, considered default retransmission limit 7, but this limit is 3 in *hostapd* and *wpa_supplicant* implementation [27]. To address this, we abstract these parameters as *Boolean* variables rather than integers with specific values.

Second, some of the MAC header fields are indeed defined as *"Vendor Specific"* or are left as *"implementation decision"* by the standard [16, § 4, 6, 9–12], but without providing a list of acceptable options, range of values for numerical fields, etc. We understand that certain parameters and/or frame elements may need to be vendor- and implementation-dependent, but the lack of any warning or sanity check for the ranges or corner cases in the standard and leaving it entirely to the interpretation of the developer can result in vulnerable implementations. For example, the standard leaves the delay between two authentication failures as an implementation decision without providing any suggested range. We show in the following how it can lead to a major consequence.

B. Abstract Model and Desired Property

1) High-level Approach: We first model the existing connection establishment portion of the Wi-Fi system. Our model is designed based on propositional logic. Next, we use a MC to find a violation/counterexample (if there is any) of a desired property of the standard under our adversarial model, similar to [25], aiming at discovering the vulnerabilities in the system.

2) Modeling the Connection Establishment Phase: Our abstract model for the Wi-Fi connection establishment, $\mathcal{M}_1$, is designed using finite state machines (FSMs), one for the station and one for the AP, and they can communicate with each other. We assume that one or multiple channel switches are possible during a connection establishment.

Because a channel switch can occur at any time during this phase, and the action of the system does *not* depend on the specific frame, we can consolidate the states corresponding to all the intermediate $N_T - 2$ frames, where N_T denotes the total number of unique pre-authentication frames exchanged by an AP-STA pair after the first and before the last frame, into one state—the connection establishment (CE) state. Therefore, without loss of generality, we simplify the model by including only the first and last frames (see Fig. 2).

States: Each of the FSMs has five states and initialized at disconnected state (*STA Discon, AP Discon*), as shown in Fig. 2.

State Transitions: Each transition has two main components: *condition* followed by *action*. Condition implies a logic specifying when a certain transition will happen after an action is triggered. For a transition, an action can be null, but there must be a condition. See Fig. 2, where for simplicity, we only provide the main transitions between the states. A transition between two states in a FSM depends on the last transmitted or received frame. The station goes to the next state, *STA CE*, when it transmits its first pre-authentication message (*probe request*) to an AP. Similarly, an AP moves to CE state only when it receives that frame from the station. They enter the connected state when the last message of the 4-way handshake is received (by the AP) or transmitted (by the station). The connected state of each of them can be over any of the three channels x, y, or z.

3) Implementation: We use a symbolic MC, NuSMV [28], to implement and check our model. It supports the analysis of system specifications expressed in computational tree logic (CTL) and linear tree logic (LTL) based on their properties. However, it is not straightforward to check a model in NuSMV since the process of property checking for each adversary action is manual. To explain how NuSMV is used to model and check the properties, we provide a few simplified examples below.

```
VAR
    STA_location:
            {STA_Discon,
             STA_CE,
             STA_Con_ChX,
             STA_Con_ChY,
             STA_Con_ChZ};
```

In the example snippet above, all of the five locations (i.e., states) of the station are together defined as a variable $STA\_location$. Note that the Wi-Fi's connection establishment includes AP FSM as well and many more other variables, transitions, etc.

Then in the following example snippet, we can see that the station's first state is initialized at the STA_Discon and the next states are based on the condition and action. The $STA\_location$ moves from STA_Discon to STA_CE if STA_firstp (the first pre-authentication frame transmitted by

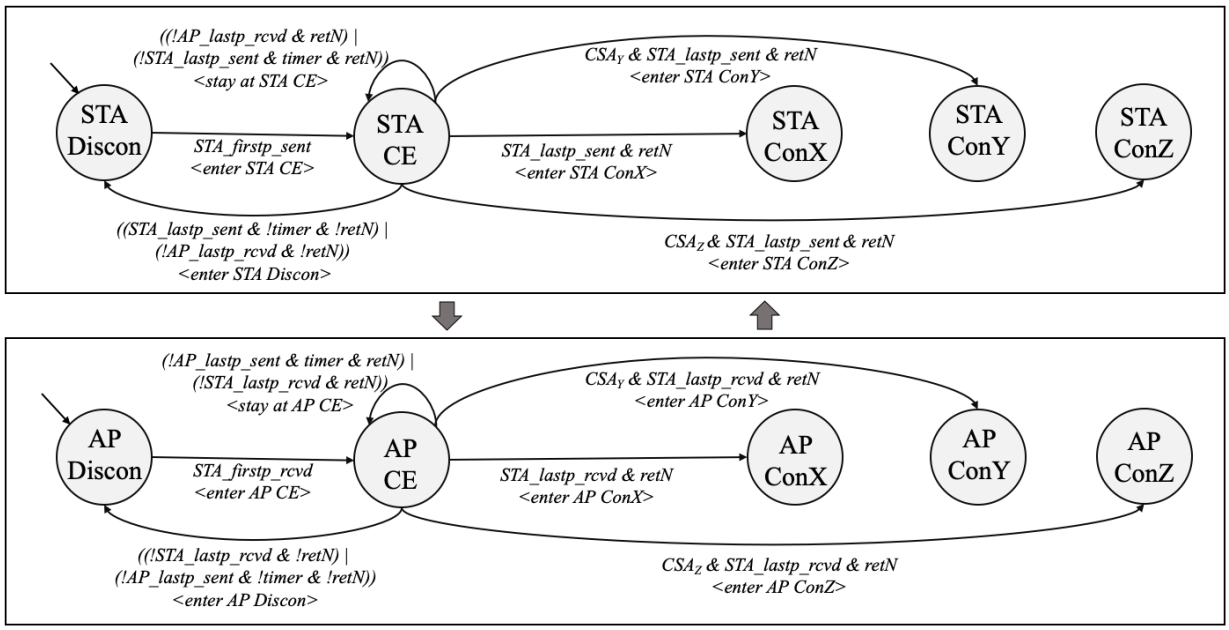

Fig. 2: Simplified abstract model for the existing connection establishment phase ($\mathcal{M}_1$).

the station) is sent. You can see this transition from disconnected state to CE state of the station in the Fig. 2. We provide further implementation details below.

```
ASSIGN
    init(STA_location) := STA_Discon;
    next(STA_location) := case
          (STA_location = STA_Discon)
        & (!STA_firstp): STA_Discon;
          (STA_location = STA_Discon)
        & (STA_firstp) : STA_CE;
```

4) Adversary-controlled Model: We define the model $\mathcal{M}_{adv1}$ as the model $\mathcal{M}_1$ taken over by the adversary who performs certain action(s) based on its capabilities. The actions of the adversary, as listed in our adversary model in Section III, include injecting a spoofed CSA element in one (or more) of the pre-authentication frames to force a station and/or an AP to change its channel, dropping/blocking one of the messages to interrupt the connection establishment, etc.

In the following simplified example snippet of the $\mathcal{M}_{adv1}$ model, you can see that a fake CSA element for channel y ($fake\_CSA\_Y$) is activated by the adversary. This will force a station to move from channel x to channel y.

```
ASSIGN
    init(fake_CSA_Y) := TRUE;
```

5) Property to Check: The main property we verify is Q:

It is always the case that a station and an AP will eventually move to the connected state whenever they are in connection establishment state, and there does not exist a case when they connect to each other over two different channels.

This property is desired as its violation can indicate attacks, such as, a DoS attack (e.g. AP and station never move to connected state), or a multi-channel MitM (i.e. station ends up connecting to a malicious AP over a different channel).

```
check_ltlspec -p
        "F((STA_location = STA_Con_ChX)
         & (AP_location  = AP_Con_ChX))"
```

In the snippet above, we are checking a simplified property p, which states that the station and the AP always end up connecting over the same channel x. The command $check\_ltlspec$ suggests to check the LTL logic using the property p.

```
Trace Description: LTL Counterexample
Trace Type: Counterexample
-> State 1.1 <-
STA_location = STA_Discon
AP_location. = AP_Discon
-> State 1.2 <-
STA_location = STA_CE
AP_location  = AP_CE
-> State 1.3 <-
STA_location = STA_Con_ChY
AP_location  = AP_Con_ChX
```

The MC takes $\mathcal{M}_{adv1}$ as input and checks whether all possible executions of $\mathcal{M}_{adv1}$ satisfy a desired property. If it finds a violation (i.e., it returns false), then it provides a counterexample with the traces. We consider the counterexample as the steps of an attack that reveal a vulnerability in the connection establishment model. In the snippet of a counterexample above, since the adversary has activated a fake CSA for channel y to the station, it's final state is STA_Con_ChY, but you can see the AP location is AP_Con_ChX.

C. Findings

We explore all possible actions of the adversary by refining the property Q to verify the model $\mathcal{M}_{adv1}$. We publish[2] the implementation of our MC and present our findings below:

① We first revise the property Q as Q_1: *There does not exist a case when an AP and station connect to each other over two different channels.* This checks the $\mathcal{M}_{adv1}$ for any vulnerability related to channel switching. We explore all possible cases of AP or station switching, including single to multiple switch(es).

1) *Action:* The adversary sends fake CSA(s) to the station. There are different variations of attempts it can take during one connection establishment. For example, it can (i) send a fake CSA to switch to channel y, or channel z; (ii) send multiple fake CSAs to switch to channel y first and then channel z (and possibly again to channel x, etc.).
   MC Finding: As expected, we find a counterexample corresponding to the classic multi-channel MitM (where the station is deceived to switch its channel only once, see Section II-D). We find a total of six instances of this attack variant considering three channels and multiple switches, assuming the AP stays on the original channel x (state *AP ConX*) but the station is eventually found on either one (*STA ConY*, *STA ConZ*, or even again *STA ConX*).
2) *Action:* The adversary forces only the AP to switch channel (by completely blocking its activities on the original channel, i.e., no receiving/transmitting takes place at the AP). It can try different channels (y or z) for single or multiple switches, force the AP to even return to the original channel x, etc.
   MC Finding: The AP is forced to switch to a different channel while the station remains on the original one (opposite of what happens in a classic MitM case). This is possible when, e.g., an adversary jams the AP's original channel. We find a total of six instances of this same attack variant considering three channels and multiple switches.
3) *Action:* The adversary sends a fake CSA to the station and, at the same time, forces the AP to switch channel.
   MC Finding: This reveals a new (third) variant of multi-channel MitM where both the AP and station can be forced to switch their channels to two different ones (e.g., y and z) other than the original one (e.g., x), as shown in Fig. 3. We find two instances of this attack variant.

② Now, we revise the property Q as Q_2: *It is always the case that a station and an AP will eventually move to the connected state whenever they are in the connection establishment state.* This checks the $\mathcal{M}_{adv1}$ for any DoS vulnerability aside from persistent jamming (i.e., under a good channel quality).

According to the standard, if a pre-authentication frame's retransmission limit is exhausted following repeated timeouts, then the state of the station (and the AP) should move from connection establishment (*STA CE, AP CE*) to its initial disconnected state [16, § 6.4]. It is not specified in the standard, but we conjecture based our validations (see Section VI) that if the exhausted frame is an authentication or association frame, then the station would send a deauthentication or disassociation notification before it would go to disconnected state.

[2]https://github.com/hoquenaureen/wifi-preauthvul-analyze

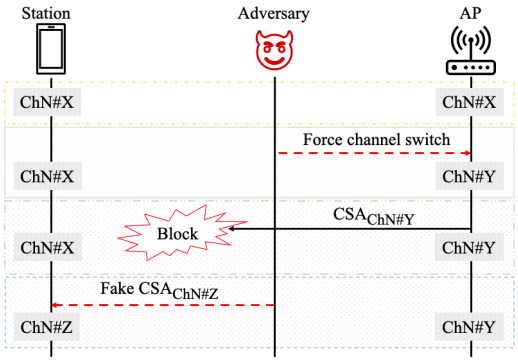

Fig. 3: A new variant of multi-channel MitM where both the AP and station can be forced to switch to two different channels other than the original one. ChN indicates the channel number.

Action: The adversary repeatedly exhausts one of the pre-authentication frames by, e.g., selectively jamming a portion of that frame so as to not easily being detected or noticeably increase the noise floor (a trigger to try a different AP).

MC Finding: If the retransmission limit is exhausted, then the station can never finish the CE phase as it moves back to *STA Discon* after a long delay. If the attacker keeps doing it, then the station will keep resetting its connection establishment with the *same* AP, as that AP continues to exhibit the highest received signal strength.. However, the standard does not specify how a station should handle such cases, e.g., verifying the channel quality and taking actions accordingly, or temporarily blacklisting that AP and trying another one. The consequences of such an attack may include:

- DoS: The victim station will continuously fail to connect to an available APs, and hence, the Internet.
- Battery depletion: Repeatedly resetting the entire process with long delays will gradually drain the station's battery.
- User frustration: A user who urgently needs to have access to the Internet may manually choose to connect to a random AP which could be malicious.

V. Analysis of OCV Technique

We also model and formally analyze a Wi-Fi system with the *optional* OCV mechanism. Fig. 4 is the model $\mathcal{M}'_1$ for the existing connection establishment process with the OCV technique in place.

The adversary in our model takes over $\mathcal{M}'_1$ and performs actions that we have explored in Section IV. Let the adversarial controlled model be $\mathcal{M}'_{adv1}$ and the MC takes it as an input. The MC checks whether all possible executions of $\mathcal{M}'_{adv1}$ satisfy that property. In the following, we describe our findings based on specific adversarial actions:

① We use the revised property Q_1. Although the MC finds no counterexample when an adversary establishes an MitM using a fake CSA, it returns a counterexample when it establishes an MitM by abusing other methods (e.g., relaying attack; a rogue AP stays in the same channel but in a set up where the AP and station cannot directly communicate).

② With the revised property Q_2, we find that when an adversary tries to exhaust the limit of a frame's timeout

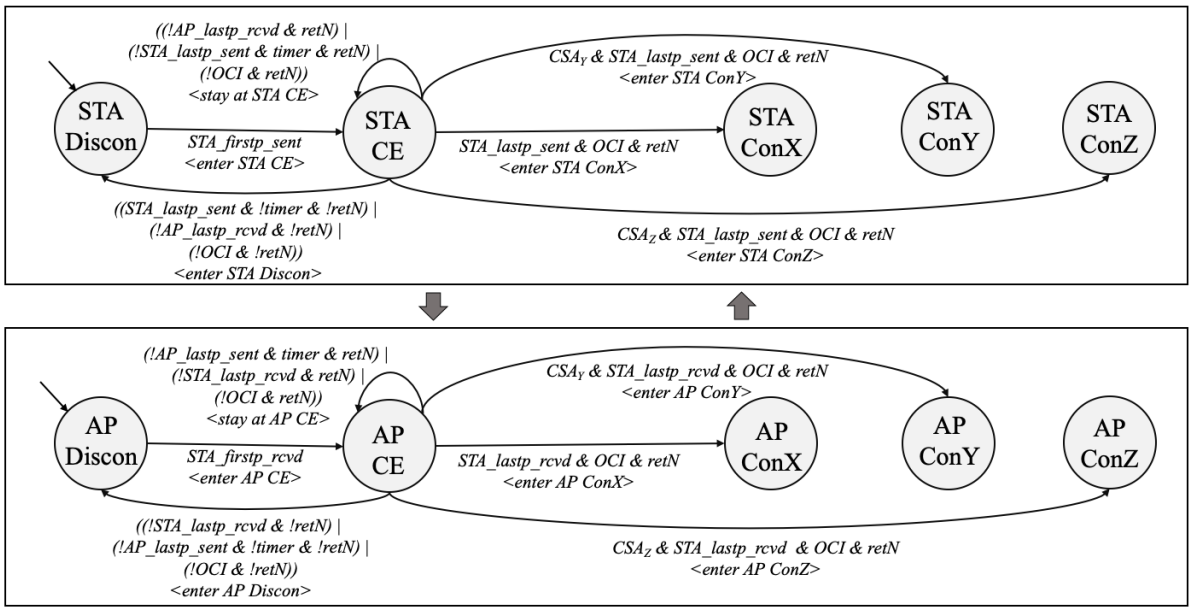

Fig. 4: Simplified model for the existing connection establishment process with the optional OCV technique ($\mathcal{M}_1'$).

and retransmission, then the state of station (and AP) from connection establishment state (*STA CE, AP CE*) to their initial disconnected state; the similar vulnerability that we find in the existing system (② in Section IV-C). If the adversary keeps repeating this action, then a station will keep resetting the whole process. The consequences of this DoS attack are same what we discuss in ② of Section IV-C, likely with longer delays.

In summary, our MC confirms that the OCV technique is capable of eliminating any CSA-based MitM (including the classic one), but it is not aware of other types of MitM and it does not protect the system from the DoS vulnerability.

Empirical Evaluation: To estimate the delay introduced by the OCV mechanism during each channel switch, which is added to the latency of each connection establishment attempt under multiple adversarial (e.g., IV-C-①) channel switches or likely under our DoS attack, we consider one AP with one associated station under a typical enterprise Wi-Fi network. We use an iPhone 6s (version 13.3.1) as a station of an Aruba AP and record the frame arrival times using Wireshark. We find that connection establishment takes an average of 300.18 ms (standard deviation of 18.03 ms) to complete. We observe that during this phase, on average, each frame takes 10.72 ms to travel from the AP to the station (or vice versa) and the average inter-frame time for the frames sent by the AP is 18.76 ms.

As discussed in Section II-E, a pair of SA query frames will be exchanged between an AP and each of its associated stations when a channel switch occurs. If an AP has b associated stations, then a channel change will introduce $2b$ new frames into the system [16, § 11]. To avoid any collisions, the OCV scheme sets a backoff counter for all the stations where this delay depends on the collision window (CW) size (can be anything between 7 to 255 slot-time). Therefore, for each channel switch, it incurs at least $2bt + \sum_{i=1}^{b} CW_i \times S_t$ ms of communication overhead, where, t is the travel time of the SA Query frame and $i = \{1, 2, .., b\}$. S_t is the slot time, a constant 0.009 ms. During each channel switch, the station will face an additional $2 \times 1 \times 10.72 + 7 \times 0.009$ = 21.503 ms of delay. Therefore, for each channel switch, the OCV technique requires a minimum of 7.16% extra time on top of the baseline 300.18 ms —it is *only* from the SA query exchange per associated station. This overhead increases with the number of associated stations (and the number of channel switches). The total connection establishment time with the OCV scheme is more because the delay that it adds from frame extension for including the OCI elements and the computation of channel validation are not considered here.

VI. DoS Vulnerability – Validation & Countermeasure

Next, we validate the DoS vulnerability that we have identified from our formal analysis and propose a countermeasure to mitigate it.

A. Validation of DoS Vulnerability

Our formal analysis above identified that if one of the pre-authentication frames' retransmission limit was repeatedly exhausted, then the station would be going back to the disconnected/probing state. An adversary needs to keep stealthily blocking any specific pre-authentication frame (i.e., selective jamming) to launch such an attack. In the following, we validate this DoS vulnerability. Our results show that the latest version of widely used *wpa_supplicant*[3] (v2.10) is vulnerable as it is not able to handle missing pre-authentication message(s). We have responsibly disclosed our findings to the Wi-Fi Alliance and publish our implementation. The Wi-Fi Alliance, in their most recent response, acknowledged the receipt of our disclosure and expressed interest in discussing our findings with their members.

[3]It is the supplicant implementation found in Linux and Android systems comprising a large share of devices on the Internet [29].

TABLE I: Validation tests and the observed behaviors.

Test	Blocked Frame	Observed Behavior	Reaches Connected State?
1	Probe response	The station continues the connection process (by sending an authentication commit frame) regardless of a probe response from the AP.	Yes
2	Authentication commit	The station transmits an authentication commit frame three times before sending a deauthentication notification. Then, it returns to probing, and once the timeout delay ends, it repeats sending the authentication frames with new scalar and finite field element values.	Never
3	Authentication confirm	The station transmits an authentication commit frame three times before sending a deauthentication notification. Then, it returns to probing, and once the timeout delay ends, it repeats sending the authentication frames with new scalar and finite field element values.	Never
4	Association response	The station sends an association request frame three times before re-attempting the CE process. Upon each set of three requests, an authentication success frame is sent to confirm the re-usage of previously sent authentication frames.	Never
5	Handshake msg 1	The station transmits an authentication success frame and re-executes the association steps. Then, the station returns to probing and repeats this process.	Never
6	Handshake msg 3	The station transmits an authentication success and re-executes an association request. The Handshake msg 1 and 2 are then transmitted (they contain different nonce values).	Never

1) Implementation and experiments: We use the Wi-Fi framework [17], that is integrated with *hostapd* and *wpa_supplicant*, on a virtual machine running Ubuntu 20.04.3 LTS 64-bit. This framework enables us to simulate an actual Wi-Fi system, selectively block any pre-authentication frames sent by the AP, and monitor the behavior of a station. This setup allows us to easily implement tests using widely used Wi-Fi daemons (*hostapd* and *wpa_supplicant*). We utilize WPA3-Personal mode to validate our findings. We bypass the functions that send each pre-authentication message by toggling new *Boolean* variables we introduce that correspond to each message type, thus making the *hostapd* assume that the message was sent properly. In this way, we can measure *a station's (wpa_supplicant's) behavior* by simulating a stealthy jamming attack. Since the Linux and Android clients use *wpa_supplicant* by default, and we do not know the extent of adopting *hostapd* (as AP/router manufacturers may use proprietary software), we only test the station's behavior via *wpa_supplicant*. We execute our experiment 30 times, each time we block one specific pre-authentication frame of the AP until it exceeds the retransmission limit, and observe the station's behavior.

2) Results: We first notice that the retransmission limit is set to 3 in the *wpa_supplicant* implementation. Then in test cases 2-6 (see Table I), we observe that the station consistently returns to probing once the transmission limit is exhausted. It supports our formal analysis finding– the station keeps going back to the disconnected/probing state under such a DoS. Additionally, we show the authentication failure delay values in Table II. Following each authentication failures, the station will wait for the corresponding timeout value for the total number of failures up to that point, resulting in a self-inflicted DoS where an attacker can be sure there will be no connection attempt. This DoS is self-inflicted because *wpa_supplicant* has a function that prevents initiating the connection process based on the number of authentication failures for a (long) period. We noticed the station was still able to connect to other APs regardless of the delay state regarding the first AP. This behavior of the station can be abused to coerce it into connecting to a malicious AP.

A Collateral Damage: Battery Depletion– The process of forcing a station into a constant state of connection attempts, each time with a set of fresh random values for Dragonfly handshake (tests 2 and 3 in Table I) and a set of new nonces for 4-way handshake (tests 5 and 6), may have some implications on its battery usage. Due to the nature of our testing environment using the *Wi-Fi framework* on virtual machines, it is difficult to model hardware battery usage on a target station. We leave testing on hardware devices to future work.

TABLE II: Authentication failure delays. Here, num_f is the number of authentication failures and t_d is the delay before retransmitting. Delays are in *seconds.*

num_f	t_d (sec)	Accumulated Delay (sec)
$num_f = 1$	10	10
$num_f = 2$	20	30
$num_f = 3$	30	60
$num_f = 4$ or 5	60	120 ($num_f = 4$)
$5 < num_f \leq 10$	90	270 ($num_f = 6$)
$10 < num_f \leq 50$	120	750 ($num_f = 11$)
$num_f > 50$	300	5370 ($num_f = 51$)

B. Possible Mitigations

An adversary can take advantage of this DoS vulnerability and intentionally put a station into an infinite loop of disconnection. This attack is not easily detectable (and it is power-efficient) since an adversary only needs to apply selective jamming. To mitigate this attack, one possible approach is to randomize the pre-authentication frame's timeout window and retry count values since the adversary in our attack takes advantage of fixed values for the timeout. We suggest randomly selecting the number of retransmissions and timeout values within a fixed range. We test adding a random number generator that changes the timeout value to any value between 5 and 60 seconds instead of fixed delay values (Table II). This technique ultimately forces an adversary to continuously deny messages by continuous jamming (an expensive attack), as opposed to knowing exactly when to selectively and stealthily activate a DoS condition. Our mitigation technique shows that it provides a more secure alternative to the current station functionality by forcing an adversary to commit to consistent jamming. Additionally, we suggest that the standard should state how a station should behave while in such scenarios.

VII. Related Work

In [30], Eian *et al.* use formal methods to discover DoS (specifically, protocol deadlock vulnerabilities) in IEEE 802.11w or management frame protection (MFP) protocol (a standard to improve management frames security) [31]. This protocol only supports the frames that are exchanged after the key-generation stage. Therefore, this study does not include the pre-authentication phase of a Wi-Fi system.

The 4-way handshake has been analyzed widely [6], [7], [15], [32]. The key reinstallation vulnerability of the WPA2 protocol is further confirmed using formal analysis in [15], but it covered only the 4-way handshake phase, and did not capture potential spoofing or jamming (e.g., channel/training signal) attacks. Vanhoef *et al.* analyzed the implementation of *hostap* and show that the 4-way handshake is vulnerable to downgrade and DoS attacks by forging [6] and blocking [32] the fourth message in the handshake. It is required to study if any vulnerability exists before the 4-way handshake phase. For example, the first step in KRACKs was to establish an MitM position by abusing an unprotected CSA element. Without analyzing other Wi-Fi pre-authentication's different stages (network discovery, open authentication, association, EAP authentication), it cannot be claimed to be *fully analyzed* and *secure*.

VIII. Conclusion and Future Directions

While many research efforts in Wi-Fi security focus only on the analysis of the 4-way handshake, the pre-authentication phase of the connection establishment in Wi-Fi was largely unexplored. In this paper, we formally analyzed this phase and showed that there were corner cases (i.e., vulnerabilities) not previously identified. Specifically, our analysis exposed one new variant of multi-channel MitM attack and a DoS vulnerability in the standard. An adversary can abuse them to launch attacks that may have severe consequences. We demonstrated the practicality of the new DoS vulnerability through experiments to further confirm that this attack could stealthily prevent a station from connecting to a preferred AP for around 90 minutes, with additional 5-minute delays after an additional failure. We also developed a mitigation technique. We plan to further investigate the DoS vulnerability conducted with real (battery-run) devices and against *hostapd* as our future work.

References

[1] M. Vanhoef and F. Piessens, "Advanced Wi-Fi attacks using commodity hardware," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, New Orleans, LA, USA, Dec. 2014, pp. 256–265.

[2] ——, "Key reinstallation attacks: Forcing nonce reuse in WPA2," in *Proc. ACM SIGSAC Conf. Comput. and Commun. Secur. (CCS)*, Dallas, TX, USA, Oct. 2017, pp. 1313–1328.

[3] M. Vanhoef and E. Ronen, "Dragonblood: analyzing the Dragonfly handshake of WPA3 and EAP-pwd," in *Proc. IEEE Symp. Secur. & Privacy (S&P)*, San Francisco, CA, USA, May 2020, pp. 517–533.

[4] M. Vanhoef, "Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation," in *Proc. USENIX Symp. Secur.*, Virtual, Aug. 2021.

[5] T. V. Goethem, M. Vanhoef, F. Piessens, and W. Joose, "Request and conquer: Exposing cross-origin resource size," in *Proc. USENIX Symp. Secur.*, Austin, TX, USA, Aug. 2016, pp. 447–462.

[6] M. Vanhoef and F. Piessens, "Predicting, decrypting, and abusing WPA2/802.11 group keys," in *Proc. USENIX Symp. Secur.*, Austin, TX, USA, Aug. 2016, pp. 673–688.

[7] M. Vanhoef, D. Schepers, and F. Piessens, "Discovering logical vulnerabilities in the Wi-Fi handshake using model-based testing," in *Proc. Asia Conf. Comput. and Commun. Secur. (ASIACCS)*, Abu Dhabi, UAE, Apr. 2017, pp. 360–371.

[8] "Wi-Fi Alliance: Discover Wi-Fi Passpoint," accessed: June 2, 2022. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/passpoint

[9] "Achieving nomadicity: Accessing the Internet anytime, anywhere," accessed: June 2, 2022. [Online]. Available: https://www.ciscopress.com/articles/article.asp?p=1681067

[10] "Wireless Broadband Alliance: Openroaming," accessed: June 2, 2022. [Online]. Available: https://wballiance.com/openroaming/

[11] "WiFi4EU: Free WiFi for Europeans," accessed: June 2, 2022. [Online]. Available: https://wifi4eu.ec.europa.eu

[12] "Wi-Fi CERTIFIED Passpoint® deployment guidelines, rev 1.3," accessed: June 2, 2022. [Online]. Available: https://www.wi-fi.org/file/wi-fi-certified-passpoint-deployment-guidelines

[13] "Worldwide enterprise WLAN market shows strong growth in Q4 and the full year 2021, according to IDC," accessed: June 2, 2022. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS48940222

[14] "WBA annual industry report 2020," accessed: June 2, 2022. [Online]. Available: https://wballiance.com/resource/wba-industry-report-2020/

[15] 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. USENIX Secur. Symp.*, Virtual Conf., Aug. 2020.

[16] "*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11," 2020.

[17] D. Schepers, M. Vanhoef, and A. Ranganathan, "A framework to test and fuzz Wi-Fi devices," in *Proc. ACM Conf. Secur. and Privacy in Wireless and Mobile Netw. (WiSec)*, Abu Dhabi, UAE, Jun. 2021, pp. 368–370.

[18] "Discover Wi-Fi security," accessed: June 2, 2022. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/security#EnhancedOpen

[19] "eduroam Security. Keeping you safe, wherever you are," accessed: June 2, 2022. [Online]. Available: https://eduroam.org/eduroam-security/

[20] "TRAI releases report on public Wi-Fi open pilot project." accessed: June 2, 2022. [Online]. Available: https://trai.gov.in/notifications/press-release/trai-releases-report-public-wi-fi-open-pilot-project

[21] "Solving the indoor wireless coverage problem: Passpoint and Wi-Fi calling, Aruba, white paper," accessed: January 20, 2021. [Online]. Available: https://www.arubanetworks.com/assets/wp/WP_Passpoint_Wi-Fi.pdf

[22] 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. ACM Conf. Secur. and Privacy in Wireless and Mobile Netw. (WiSec)*, Stockholm, Sweden, 2018, pp. 34–39.

[23] "*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE 802.11ax," 2021.

[24] D. Dolev and A. C. Yao, "On the security of public key protocols," *Trans. IEEE Info. Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[25] S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector: A systematic approach for adversarial testing of 4G LTE," in *Proc. Netw. and Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Aug. 2018.

[26] L. Xin and D. Starobinski, "Countering cascading denial of service attacks on Wi-Fi networks," *IEEE/ACM Trans. on Netw.*, vol. 29, no. 3, pp. 1335–1348, 2021.

[27] "hostapd and wpa_supplicant," accessed: June 10, 2022. [Online]. Available: http://w1.fi/

[28] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "NuSMV: A new symbolic model verifier," *Int. J. on Softw. Tools for Tech. Transfer*, vol. 2, pp. 410–425, 2000.

[29] "Operating system market share worldwide," Apr. 2022. [Online]. Available: https://gs.statcounter.com/os-market-share#monthly-202204-202204-bar

[30] M. Eian and S. F. Mjølsnes, "A formal analysis of ieee 802.11w deadlock vulnerabilities," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Oct. 2012.

[31] "*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements*, IEEE 802.11i," 2004.

[32] D. Schepers, A. Ranganathan, and M. Vanhoef, "On the robustness of Wi-Fi deauthentication countermeasures," in *Proc. ACM Conf. Secur. and Privacy in Wireless and Mobile Netw. (WiSec)*, San Antonio, TX, USA, May 2022, pp. 245–256.