electronics

MDPI

Article

A Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3

Christopher P. Kohlios [†,‡] **and Thaier Hayajneh** [*,†,‡]

Fordham Center for Cybersecurity, Fordham University, New York, NY 10023, USA; ckohlios@fordham.edu
* Correspondence: thayajneh@fordham.edu; Tel.: +1-212-636-7785
† These authors contributed equally to this work.
‡ Current address: 113 W 60th St, Room 616A, New York, NY 10023, USA.

Received: 24 September 2018; Accepted: 26 October 2018; Published: 30 October 2018

Abstract: The presence of wireless communication grows undeniably more prevalent each year. Since the introduction of the IEEE 802.11 standard for Wireless Local Area Networks (WLAN) in 1997, technologies have progressed to provide wireless accessibility to industries and consumers with growing ease and convenience. As the usage of personal devices, such as phones and watches, that connect to the Internet through Wi-Fi increases, wireless attacks on users are becoming more critical. This paper provides a novel attack model to offer an organized and comprehensive view of the possible attacks on Wi-Fi latest security standards. All existing attacks will be investigated, with emphasis on more recent attacks, such as the KRACK and PMKID Dictionary attacks. The main contribution of this paper is to analyze the technology offered in the new Wi-Fi Protected Access III (WPA3) security scheme and provide the first comprehensive security analysis and discussion to determine whether it has addressed the vulnerabilities of its predecessor. An interesting finding of this paper is that WPA3 still lacks in addressing all the issues existing in WPA2 and exploring other mitigations for future research.

Keywords: WPA3; Wi-Fi; attack flow; security analysis; WLAN

1. Introduction

In 1997, a standard was released by the Institute of Electrical and Electronics Engineers (IEEE) that set guidelines for creating a network in which devices could connect to each other wirelessly, known as Wireless Local Area Network (WLAN). The standard is referred to as IEEE 802.11 and has gone through a few revisions since its inception [1]. Wireless connectivity is highly advantageous over wired, but the absence of proper security could cause significant damage. If no security measures are implemented in a WLAN system, then there is nothing stopping an attacker from joining a network and capturing traffic or injecting his/her own malicious traffic. To counteract this problem, security protocols have been developed to ensure confidentiality, integrity and authentication.

At the time of writing, three main security protocols have been implemented for IEEE 802.11: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2). A fourth protocol, Wi-Fi Protected Access III (WPA3), was recently released to the public by the Wi-Fi Alliance on 25 June 2018. The benefits and limitations of each of these protocols will be discussed in this paper. Even though WEP is outdated and no longer permitted to be implemented in new routers and devices, it is still in service in legacy devices. Each protocol will be defined and analyzed for their benefits and limitations. Following will be a discussion on WPA3 and the problems it seeks to address in the current state of Wi-Fi security.

Aside from attacking encryption schemes, several backdoors exist that enable hackers to penetrate a network and conduct malicious activities. This paper will survey all available attacks on a Wi-Fi network using WPA2 in an organized manner based on timing. A novel categorization model will be provided to assist future researchers in studying, visualizing and categorizing attacks on Wi-Fi networks. Each attack will be described in detail and analyzed, including the new Key Re-installation Attack (KRACK) exploit released in 2017 [2] and the recent PMKID attack released in 2018 [3]. An analysis of the WPA3 security protocol will be given for each attack to determine whether or not the vulnerabilities have been addressed. Finally, we will propose defenses users can adhere to in order to prevent attacks on their networks and secure private information.

Limited existing research has surveyed the current security schemes from WEP to WPA2 [4–7]. This paper, however, seeks to create a comprehensive survey that reiterates key points of previous literature to provide the reader with enough information needed for understanding the encryption schemes.

Furthermore, several papers have described in detail different attacks that can be performed on a Wi-Fi network [2,3,8–10]. This paper aggregates these attacks into one comprehensive reference model of all Wi-Fi network attacks. There have been papers surveying attacks on mobile networks [11,12], sensor networks [13–16] and mesh networks [17], but a comprehensive attack survey on Wi-Fi networks is yet to be seen. This paper serves this purpose by clearly identifying available attacks on the current Wi-Fi security protocol, WPA2, before the introduction of WPA3. Since WPA3 shares much of its structure with its predecessor, it is therefore necessary to provide a detailed discussion of WPA2.

Likewise, there is no literature that offers an attack flow diagram to clearly display the process an attacker would take from the beginning of an attack to reach certain outcomes. The main contribution of this paper is to provide the first analysis of the security provided by WPA3 to a Wi-Fi network and identify which attacks must be addressed in future research in Sections 4 and 5.

The remainder of this paper is outlined as follows: A description of each of the current security protocols up to WPA2, along with their limitations, is given in Section 2. Section 3 provides the attack flow diagram, going into detail explaining each attack that can be performed on a WPA2-protected Wi-Fi network. Section 4 gives an overview of the features of WPA3 and provides a security analysis on the attacks described in this paper. Section 5 discusses the benefits provided by WPA3 given the security analysis with respect to the attack flow diagram from Section 2. Lastly, other mitigation methods for the remaining issues not addressed by WPA3 are given in Section 6, and 7 concludes the paper with closing remarks and future research.

2. Wi-Fi Security Protocols and Limitations

Security protocols were implemented to give security to Wi-Fi networks in the form of authentication and encryption, as opposed to just providing a wireless medium to the Internet. At the writing of this paper, WPA2 is the most used security protocol due to its high level of security and time in the market. The release of WPA3 is still new and has not gained enough popularity yet, but nevertheless has the highest level of security to date, which we will look into in detail. Even though WEP is no longer accepted as a reliable security protocol and is not implemented in new devices, it is still possible to see each protocol in some devices in today's world [18].

Along with the benefits of each of these protocols and methods of encryptions, there are also many limitations and vulnerabilities. The newest devices have the most updated security measures and are capable of supporting all of the protocols discussed. However, since older devices still exist and are being used in the world, it is still important to be aware of these limitations.

2.1. Wireless Equivalency Protocol

WEP was the first protocol used to secure wireless networks. It was introduced as part of the IEEE 802.11 security standard in September 1999 [18]. It was created to provide a similar degree of security found in wired networks. It uses the Rivest Cipher 4 (RC4) stream cipher for encryption to increase the overall speed of communication, compared to slower encryption schemes such as DES [19]. The RC4 uses a 40-bit shared key used with a 24-bit Initialization Vector (IV). The shared key and the IV are concatenated to create a 64-bit key. The 64-bit key is then a seed value for a Pseudo Random

Number Generator (PRNG) [18]. The plaintext is then sent to an integrity check algorithm called Cyclic Redundancy Check 32 (CRC-32), where the product is the Integrity Check Value (ICV), which is used to compare to the plaintext for integrity. The key sequence generated by the PRNG is then XORed with the plaintext concatenated to the ICV to produce the ciphertext. The IV is concatenated to the ciphertext to use for decryption by the receiving party [19]. This same process is done in reverse to obtain a valid plaintext message.

As previously stated, WEP was the first protocol used in securing wireless networks. However, WEP has been proven to be easily broken [8–10]. One of the main vulnerabilities in WEP is the ability to broadcast fake data packets. Due to the fact that WEP is using shared key authentication, it makes it easy for an attacker to forge an authentication message. In shared key authentication, knowledge of a shared WEP key is demonstrated by encrypting a challenge. An attacker can observe the challenge and the encrypted response to determine the RC4 stream used for encryption. The attacker can use that same stream in the future [19]. Another shortcoming of the WEP protocol is the reuse of the initialization vector. Different cryptanalysis methods could then be used to decrypt the data [18].

Key management is also a major vulnerability of WEP. Key distribution is not specified by the standard. A field in each message is used to identify the key that is used. In a wireless network, only one key is used, so if more than one user is using the key, there is an increased chance for key decryption [18]. Along with a lack of key management, the small size of the keys is also a weakness to this protocol. A 40-bit key is used, making a brute force attack likely to decrypt the key [18].

2.2. Wi-Fi Protected Access

2.2.1. Protocol Details

WPA was created in 2003 by the Wi-Fi Alliance in order to overcome flaws in WEP. Version 1 was designed as an intermediate solution intended to correct the WEP cryptographic deficiencies without requiring new hardware and uses the Temporal Key Integrity Protocol (TKIP) for encryption. The 128-bit per packet key is dynamically generated for every packet. The Pre-Shared Key (PSK) is a static key used to initiate communication between two parties. A 256-bit key is used to authenticate the wireless devices, which is never transmitted over the air. The Message Integrity Code (MIC) key and encryption key are derived from the PSK.

TKIP uses an RC4 device (implemented in the hardware of a wireless network adapter) to alter the way the shared key is used. WEP uses a shared key in encryption, while TKIP uses a shared key to generate other keys. TKIP made four improvements to WEP: (1) it encrypted the MIC to prevent falsifications; (2) used a strict IV sequence to prevent replay attacks; (3) used improved key generation and (4) refreshed keys to prevent key repetition attacks [18].

TKIP keys are used after a client is authenticated and associated. A four-way handshake, demonstrated in Figure 1, is performed using the TKIP keys, resulting in a 512-bit key that is shared between the client and the access point. A 128-bit temporal key and two 64-bit MIC keys are derived from this 512-bit key. One MIC key is for the Access Point (AP) to client communication and the other for client to AP communication. The sender of a TKIP frame calculates the MIC value of each data packet using an algorithm, called the Michael algorithm, which takes the MIC and a secret key.

The data packet concatenated with the MIC is then encapsulated using WEP so it can be implemented on old WEP hardware. An ICV is appended, then the packet is encrypted using RC4 and a key that uses the function that combines the temporal key, transmitter MAC address and the TKIP Sequence Counter (TSC). The receiver will check to see if the TSC is in order and the ICV is correct. If either of these checks are not valid, the frame will be dropped. The original data packet is reassembled, and the MIC value is verified. If it is accepted, the TSC replay counter is updated [20].

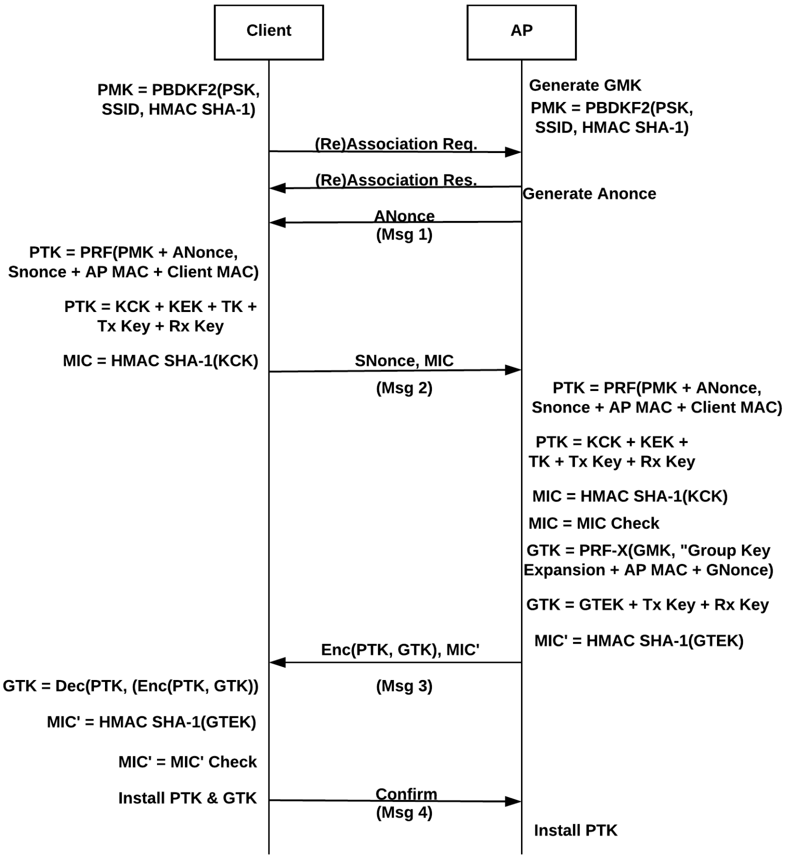
Client
AP
PMK = PBDKF2(PSK, SSID, HMAC SHA-1)
Generate GMK
PMK = PBDKF2(PSK, SSID, HMAC SHA-1)
(Re)Association Req.
(Re)Association Res.
Generate Anonce
ANonce (Msg 1)
PTK = PRF(PMK + ANonce, Snonce + AP MAC + Client MAC)
PTK = KCK + KEK + TK + Tx Key + Rx Key
MIC = HMAC SHA-1(KCK)
SNonce, MIC (Msg 2)
PTK = PRF(PMK + ANonce, Snonce + AP MAC + Client MAC)
PTK = KCK + KEK + TK + Tx Key + Rx Key
MIC = HMAC SHA-1(KCK)
MIC = MIC Check
GTK = PRF-X(GMK, "Group Key Expansion + AP MAC + GNonce)
GTK = GTEK + Tx Key + Rx Key
MIC' = HMAC SHA-1(GTEK)
Enc(PTK, GTK), MIC' (Msg 3)
GTK = Dec(PTK, (Enc(PTK, GTK))
MIC' = HMAC SHA-1(GTEK)
MIC' = MIC' Check
Install PTK & GTK
Confirm (Msg 4)
Install PTK

Figure 1. A detailed diagram of the four-way handshake. Msg, Message.

2.2.2. Limitations

There are a number of major shortcomings with the WPA protocol. The first is the usage of the RC4 algorithm over the more advanced AES algorithm [18]. As previously stated, having two or more RC4 keys computed under the same IV makes it easy for an attacker to compute the Temporal Key (TK).

The next shortcoming is in WPA-PSK mode. It is vulnerable to brute force attacks if a poor password is used. A dictionary attack can be used if the password is less than 20 characters. Another shortcoming of WPA is that there is a greater performance overhead than WEP [18]. According to research done by Tripathi and Damani [21], there is lower average throughput and greater overhead when using WPA-TKIP when compared to the throughput and overhead when using WEP.

The main vulnerability of WPA is in TKIP. This is due to hash collisions when using hash functions for TKIP key mixing [18]. It is easy for an attacker to compute the TK and decrypt any packet if two or more RC4 keys are computed under the same IV [22]. This makes WPA susceptible to threats related to hash collisions while using hash functions in TKIP key mixing. A per-packet key mixing function exists to de-correlate the IVs from weak keys. A re-keying mechanism provides new encryption and integrity keys. This function, called the temporal key hash, produces a 128-bit RC4 encryption key. If an attacker collects a few RC4 keys calculated under the same IV, they will be able to recover the TK and the MIC key, which is used to detect forged packets [23]. Most new equipment being released today does not support a TKIP only option. In 2014, TKIP was scheduled to be disallowed entirely. However, there is still legacy equipment in the field today that supports and is using TKIP [20].

2.3. Wi-Fi Protected Access Version 2

2.3.1. Protocol Details

WPA2 guarantees that all equipment with it installed can support 802.11i, which is a standard to provide enhanced security in the Medium Access Control (MAC) layer [4]. This introduced Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). It uses the Advanced Encryption Standard (AES) block cipher for data encryption. TKIP is also available for backwards compatibility with existing hardware. Additionally, WPA2 has PSK and enterprise modes [18]. Due to the nature of AES, WPA2 requires replacing older hardware because AES has extensive processing demands [24]. In order to generate keys in WPA2, a four-way handshake is needed to get a Pairwise Transient Key (PTK) and a Group Temporal Key (GTK), as well as a group key handshake for GTK renewal or host dissociation [18].

In the beginning of the handshake, as depicted in Figure 1, both the client and AP have a Pairwise Master Key (PMK), which is a PBDKF2 function of the PSK, the Service Set Identifier (SSID), or name, of the AP and a Hash Message Authentication Protocol (HMAC) function. After the client sends a request to connect and the AP acknowledges the request, the AP will generate a nonce (Anonce) and send it to the client. A nonce is a random value that is known by the sender to test that the receiver knows a certain piece of information. The client is tested by using the nonce along with some other information to create a new value that the AP can test. To create the PTK, the client will generate its own nonce (Snonce) and concatenate that with the Anonce, the PMK and the MAC address of both the AP and client. Part of this key is used to derive the MIC, to ensure that the Snonce sent in plaintext was not altered in transmission. Once the AP receives the Snonce and the MIC, it will derive the PTK using the same information as the client and confirm that the MIC match. The PTK is derived through the two random nonces exchanged, which will be different every session, making the PTK fresh every session.

CCMP is based on the Counter mode (CTR) with Cipher-Block Chaining (CBC) message authentication code of AES. CTR is used for data confidentiality, and CBC message authentication code is used for authentication and integrity [25]. As Figure 2 shows, the CCMP encryption takes in the PTK or GTK (if the message is unicast or broadcast, respectively) encryption key and runs it through an AES encryption algorithm along with the 802.11 headers and flags, MAC address of the transmitter, the packet number of the message and some counters that are required for counter mode in AES. AES is a block cipher algorithm that supports 128–256 keys in sequences of 32 bits. The length of the key and the length of the block are chosen independently. The value of these blocks is changed after each round is completed. The key is enlarged into 44 32-bit words, with each word equaling four bytes. This creates 11 keys to be used in 10 rounds, the first of which is used for the initialization of the encryption and the last used for initialization of the decryption. An increased number of rounds are used with an increased key size. Each round consists of one permutation and three substitutions. This algorithm is considered secure due to the complexity of the key extension, as well as the complexity of the transformations, which, as stated above, consist of a combination of permutations and substitutions in each round [4].

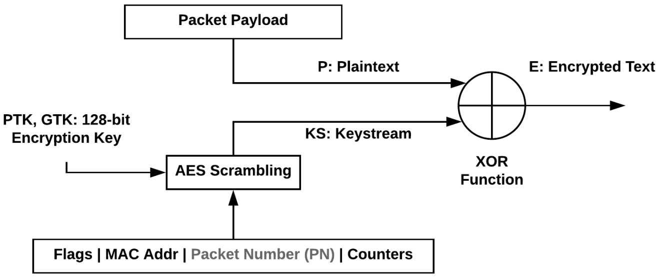

Figure 2. CCMP encryption diagram.

2.3.2. Limitations

One limitation of WPA2 is the need for upgraded hardware to deploy it. This is due to the fact that a CCMP and AES implementation requires a change to existing hardware. All new hardware being released today can support WPA2. WPA2 is supported in all Wi-Fi devices certified since 2006. However, for networks that have already been deployed, it can be expensive to replace all hardware with new hardware that supports CCMP and AES [18].

It has also been demonstrated that WPA2 can be exploited by a method known as KRACK, for which we will go into further depth in Section 4.2.7. This process exploits the four-way handshake that wireless security protocols use to authenticate their users when connecting to the network. For this attack, the attacker sets the counters to their initial values and can then replay messages and decrypt them [2]. The vulnerability is that WPA2 allows reinitialization of keys, which a secure system should not.

WPA2 also allows system information, known as management frames, to be sent in plaintext packets from the client to the AP. With this vulnerability, an adversary can spoof the packets to make it look like they are coming from the target client and preform attacks such as deauthentication. The problem lies with a lack of encryption and authentication to maintain authenticity of the messages.

3. Attack Flow

In this section, we will describe the main attacks an adversary can perform against a victim client on a Wi-Fi network using WPA2-PSK security. To clearly identify all weaknesses in the design of current Wi-Fi networks, we have created a flowchart that walks the reader through the steps taken by the attacker to achieve the desired outcomes, shown in Figure 3. The diagram is broken up into three categories: states, attacks and outcomes. A state is the position the attacker is in with the ability to perform an attack or achieve a desired outcome. Going from one state to the next is usually accomplished by an attack, but can also be done directly. An attack is an action preformed against the victim or AP by the adversary to move to another state or achieve a desired outcome. An outcome is the malicious goal of the attacker; in other words, what he/she plans to accomplish. The diagram is then further broken down into four parts: Phase 1, Phase 2, Phase 3 and Phase 4. The four phases are used to separate the types of attacks based on a given set of states at that portion of the attack flow. Some states and attacks need to happen before another attack can be performed to makes sense chronologically. The phases are used to illustrate this distinction. The rest of this section will be broken up into the four phases, giving a description of each state in that phase, followed by each attack. The attacks and states will give references to the states or outcomes to which they lead.

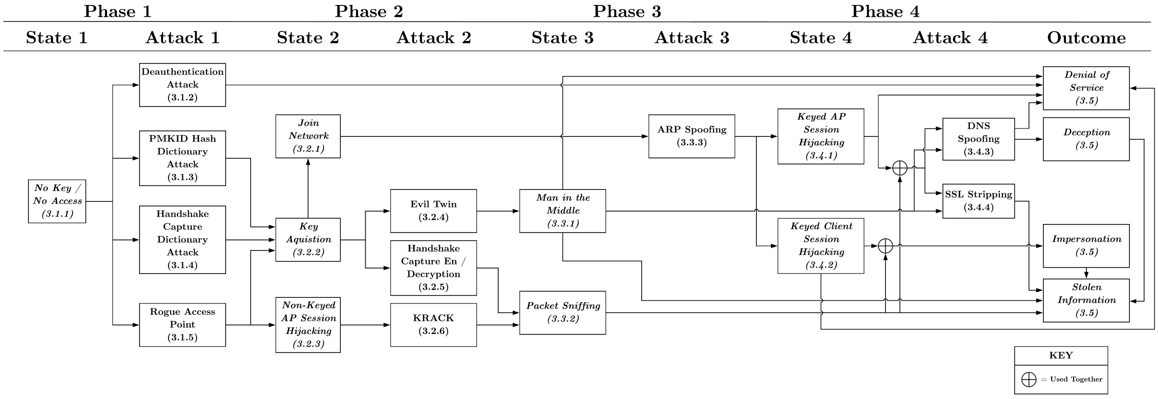

Figure 3. Attack flow diagram with corresponding section numbers.

3.1. Phase 1

3.1.1. State-No Key/No Access

This is the beginning state of an adversary initializing his/her attack on a Wi-Fi network, assuming he/she has no advantages, such as the Wi-Fi passphrase or backdoor network access. In this state, the adversary can only perform attacks visualized in Figure 4 to advance to a more advantageous state or reach a desired outcome.

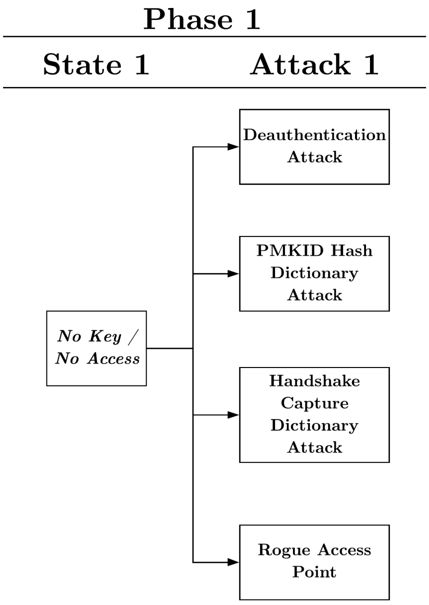

Figure 4. Phase 1.

3.1.2. Attack-De-Authentication Attack

The de-authentication attack is a straightforward attack that creates a Denial of Service (DoS) for one or many users. When a client wants to connect with an AP, it must first identify and authenticate itself to the AP. The AP will then send a response back to the client acknowledging the authentication. The client will then request an association with the AP and await the response for connection. Once those initial steps are completed, the client and AP will perform a four-way handshake to prove knowledge of the PSK and use it to derive keys for encryption. From that point on, the two devices can send encrypted data to each other. However, just as they can send authentication frames to each other, they can also send de-authentication frames to tell the other device to cut communication. De-authentication frames fall under the management frames category. Management frames are important system data packets sent from the AP to the client and vice versa. Unfortunately, management frames are sent through plaintext with no authentication protocol. The de-authentication attack exploits this vulnerability by spoofing the MAC address of the devices, pretending to be the client or AP, and sending de-authentication frames between them. The devices, thinking the flags are coming from one another, will then cut the connection with each other [26]. Multiple tools exist to perform this attack, the most common being Aireplay-ng.

3.1.3. Attack-Handshake Capture Dictionary Attack

For a client to connect to an AP, the AP must first trust that the client is allowed to join the network and give it the key that will be used in encryption of data. This trust is created and authenticated using the four-way handshake.

In this protocol, the client and the AP will communicate certain information to each other so that the other can create several keys individually to arrive at an agreed upon key, the PTK, which will be the fresh session key used for safe encrypted data transmission for that particular connection. For each new connection made between the client and AP, a new PTK will be created for encryption. This prevents a one-time derivation of the PTK by an adversary for decryption of future traffic.

To perform the off-line dictionary attack, the attacker will passively monitor the air for packets going from a client to an AP. Being that Wi-Fi connection uses frequencies and send information through the air, an adversary can eavesdrop the packets destined for a specific AP and capture them. The only components of this exchange making the connection and PTK fresh is the random nonces in the handshake. By capturing the handshake, the attacker will have enough information to test to see if a possible passphrase is correct. Referring back to Figure 1, the candidate passphrase is used to derive the PMK, which is a Password-Based Key Derivation Function 2 (PBKDF2) function of the PSK, derived from the passphrase, the SSID of the AP and an HMAC function. A PTK is created using the nonces that were captured, along with all the other information that remains constant. The MIC is then derived from the PTK, which will be compared to the captured MIC. If the MICs match, that means the candidate passphrase was correct. This process is repeated for every word in a word list until the correct passphrase is found [27].

3.1.4. Attack-PMKID Hash Dictionary Attack

A new method of off-line dictionary attacks on a Wi-Fi network was discovered accidentally six years later in August of 2018 by researcher Jens "atom" Steube when attempting to break the WPA3 security scheme. In his post [3], he details a procedure in which an off-line dictionary attack can be performed without needing to capture a handshake between another client and an AP.

The attack exploits the Robust Security Network (RSN) information element of a single Extensible Authentication Protocol (EAP) over LAN (EAPOL) frame. This EAPOL frame is received upon the authentication phase of connection right before the four-way handshake (see Figure 1). After examination of the captured frame using a packet capturing tool (e.g., Wireshark), the RSN Pairwise Master Key Identification (PMKID) can be seen under the WPA key data section as a hash value. The PMKID is calculated as:

$$\mathrm{PMK}_{\mathrm{ID}} = H(\mathrm{PMK}, \mathrm{PMK}_{\mathrm{Name}} | \mathrm{MAC}_{\mathrm{AP}} | \mathrm{MAC}_{\mathrm{STA}}) \tag{1}$$

where the PMK is the key to the function and the data part is a fixed string PMK name, the MAC address of the AP and the MAC address of the device trying to connect. With all this information known, the attacker can just compute a PMK using candidate PSKs computed from a word list of passphrases and check the candidate PMKID hash against the PMKID sent in the EAPOL frame. If the values match, then the passphrase attempted is the correct passphrase.

3.1.5. Attack-Rogue Access Point

A rogue access point is an unauthorized access point connected to a network that acts as a gateway for users. A simple demonstration of this attack is to buy an AP and physically connect it to a port that is connected to a specific network. With a wired connection, users can then access the network wirelessly and interact with it as they please. This is dangerous, as an attacker can set up an access point with a known security key and create an unwanted backdoor into a network. However, this type of attack may be difficult to perform, as gaining physical access to network ports is not always readily available.

Attackers can also use rogue access points to acquire a network key using a phishing technique, as opposed to brute force, as described in Sections 3.1.3 and 3.1.4. This attack begins the same as the evil twin attack, which will be discussed in Section 3.2.4, but the client will not fully connect to the attackers AP. Instead, upon association, the rogue AP will redirect the client to a landing page, prompting the user to re-enter the Wi-Fi passphrase (e.g., for firmware update). The page will then use a previously captured handshake that the legitimate AP used to authenticate a client and compare the MIC to a computed PTK from the entered passphrase that came through in plaintext from the web page. If incorrect, the web page will tell the user to try again until the passphrase is correct. Many may think this is suspicious and not partake, but it only takes one user out of many to enter in the right passphrase.

Finally, an attacker can use a rogue AP to have a user connect to the AP, without knowing the passphrase. This is done by imitating the SSID and MAC, as done with the evil twin attack in Section 3.2.4, but create the AP on a different Radio Frequency (RF) channel. The AP will then send a Channel Switch Announcement (CSA) beacon to prompt the user to switch channels from the genuine APs' channel to the malicious APs' channel. The client will obey because it will think it is an authentic frame from the AP due to the SSID and MAC address being spoofed. From there, you can assume a non-keyed AP session hijacking position to execute the KRACK attack, which will be discussed in Section 3.2.6.

3.2. Phase 2

3.2.1. State-Key Acquisition

From the brute force/dictionary and rogue access point attacks preformed in Section 3.1, the attacker has gained the passphrase to the AP. This will now allow the execution of the evil twin attack, as well as give the ability to legitimately join the network through the normal handshake process, as shown in Figure 5. Likewise, from this position, an adversary could monitor the air medium from packets being sent between clients and APs and, using the passphrase, can launch the handshake capture decryption attack.

3.2.2. State-Join Network

In this state, the attacker has legitimate access to the network by using the passphrase to join as an authenticated client. The state is reached only after the key acquisition state. From here, an attacker can execute ARP spoofing on the AP and clients on the network.

3.2.3. State-Non-Keyed AP Session Hijacking

This is a unique state reached by preforming a rogue access point attack. The session created between a client and a genuine AP is hijacked, making the client believe he/she is still communicating with the AP, when in reality, he/she is connected to the attacker. This is done through channel switching. The attacker, in this case, does not know the key, but merely redirected the connection. The connection began with a handshake between the client and the genuine AP to create the sessions key. Since the session was hijacked, encryption by the client using the original session key persists. Likewise, the client expects encrypted packets to be decrypted using the original key. Without knowledge of the key, handshake capture decryption is not possible. However, the attacker can perform the KRACK attack to decrypt generated traffic by the client.

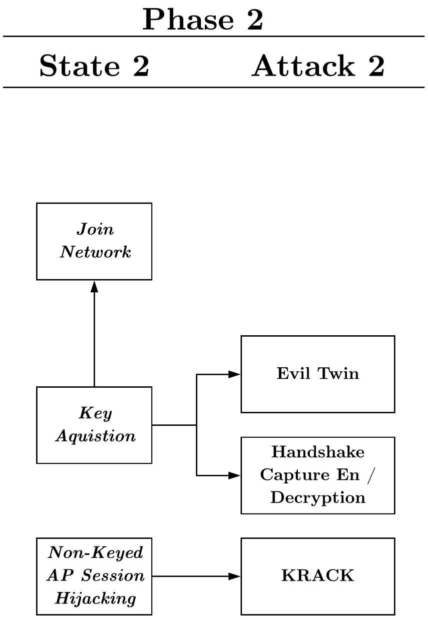
Phase 2
State 2
Attack 2
Join Network
Key Aquistion
Evil Twin
Handshake Capture En / Decryption
Non-Keyed AP Session Hijacking
KRACK

Figure 5. Phase 2.

3.2.4. Attack-Evil Twin Attack

Another common practice is to trick the client into thinking he/she is connecting to a genuine AP, while he/she is actually connecting to a rogue AP. This is a variant of the rogue AP attack known as the evil twin attack. The attacker impersonates a specific AP, in hopes that a user will connect to it. Once the user connects to the malicious AP, the attacker will be a Man-In-The-Middle (MITM) and will be able to decrypt, see and manipulate traffic that the user is receiving and sending from his/her device. The attacker will forward Internet access to the user, so the user will get what he/she wants and not suspect anything, but the attacker acts as a proxy, which views all data first.

This attack is simple in nature, due to the lack of authentication. To perform this attack, an attacker either needs a router or wireless interface adapter on a laptop. This attack is simple in nature, due to the lack of authentication. To perform this attack, an attacker either needs a router or wireless interface adapter on a laptop. Since all that is needed is the SSID, MAC address and security scheme and password to be the same to trick the device, the attacker will configure the rogue AP as such. Then, the attacker will have to de-authenticate the user from the real AP by sending de-authentication flags, as described in Section 3.1.2. Once the device is disconnected, it will begin to look for connection again. When choosing between two AP's with the same SSID, most devices will usually choose the one with the stronger signal. Again, tools such as iwconfig can change the AP's signal strength to make sure yours is higher; however, signal strength beyond a certain point is illegal in certain countries. If the target router, however, contains a passphrase, then the attacker must set up the malicious AP to have the same security protocol and the same passphrase, or else the device will try to use the remembered passphrase for the handshake and get it wrong.

3.2.5. Attack-Handshake Capture Encryption/Decryption

After understanding the authentication process and the four-way handshake, which was explained in detail in Section 2.3 and in Figure 1, we can discuss how an attacker can decrypt encrypted traffic by knowing the network key. Using a wireless adapter set to monitor mode and a traffic sniffing software, such as Wireshark, the attacker is able to view and capture all traffic flowing from a client to an AP and vice versa, being that the messages are sent through the air in a public medium. To a normal user, this traffic is useless because it is encrypted with a different PTK for each client generated by the four-way handshake upon connection. If the attacker, however, was able to capture the handshake upon connection for a particular client and has knowledge of the PSK, he/she would have enough information to derive the PTK for that client. The AP SSID and PSK are already known by the attacker to generate the PMK. The attacker will then capture the two nonces sent by the AP and the client and use them to derive the PTK, as shown in Figure 1. From there, the attacker could use the PTK, along with the CCMP protocol shown in Figure 2, to derive all messages for that session by the victim client and view the information being transmitted, as long as the attacker keeps track of the message counter from the beginning of the connection, which is used in the CCMP encryption.

3.2.6. Attack-KRACK Exploit

Vanhoef and Piessens discovered a vulnerability in the four-way handshake that would give any adversary the ability to decrypt a user's traffic without needing to capture the handshake and have knowledge of the key [2]. The vulnerability occurs in the installation of the PTK given a certain message counter. To understand how this decryption can happen, we need to examine how the keystreams are used in encryption.

The CCMP encryption method is said to be highly secure due to its use of the AES-CTR encryption. As mentioned earlier, this algorithm makes it extremely difficult for any computer to crack, impossible with the technology at the writing of this paper. However, there is another step in this algorithm that creates the vulnerability that KRACK exploits. The encrypted message sent from the client to the AP is simply the plaintext message XORed with the keystream, which is the PTK scrambled with several other parameters using AES, as shown in Figure 2.

The vulnerability in this scheme is present in the last XOR step. There is a fundamental, mathematical property of logical flow that makes the KRACK exploit possible. To create the encrypted text E, the plaintext P is XORed with the keystream KS to yield the formula:

$$E = P \oplus KS \tag{2}$$

If an adversary were to capture two encrypted packets, he/she might be able to use these two packets to decipher them. Given that the keystreams are the same, an adversary could XOR the two encrypted texts together to cancel out the keystreams and leave the two plaintexts.

Given:

$$E_1 = P_1 \oplus KS_1 \tag{3}$$

$$E_2 = P_2 \oplus KS_2 \tag{4}$$

$$KS_1 = KS_2 = KS \tag{5}$$

Then:

$$E_1 \oplus E_2 = (P_1 \oplus KS) \oplus (P_2 \oplus KS) = P_1 \oplus P_2 \tag{6}$$

If the adversary were to be able guess or know P_1, then it is possible to decrypt P_2. This can be done using a default known first message that the AP or client will send upon connection. The WPA2 keystream is designed to change so that this exploit does not happen, but the KRACK researchers found a way around this. The keystream is comprised of mainly static variables, such as the PTK, GTK, flags, MAC addresses and counters. The only variable that changes when encrypting messages is the Packet Number (PN), as shown in gray in Figure 2. With a different packet number, the keystream will be different for each encrypted message, and the XOR cancellation will not be possible.

The KRACK exploit, however, takes advantage of a flaw in the design of the EAPOL handshake to get two packets of the same keystream. After authentication and association, the client and the AP begin to send four messages to each other, known as the four-way handshake, which will give them both the keys they need to construct each keystream and start encrypting data. Once the PTK and GTK are installed on the client side, the client can begin sending data packets and encrypt using the CCMP scheme shown in Figure 2. The first message sent after this key installation will have a packet number of 1. The AP will then know to decrypt the first incoming packet using the PTK and Packet Number 1. They both then increment their packet number for the next packet sent.

This protocol, however, has a function designed to make the system more efficient, but results in a vulnerability. There are occasions where the AP would need to resend Message 3 if there was an issue with Message 4. In this scenario, the AP will resend Message 3 (Msg 3; Figure 1), and the client will respond by reinstalling the PTK and GTK and responding to an acknowledgment (Msg 4). When this happens, the packet number is also reset to 1. Knowing this, an adversary could hijack a session from the AP, replay Message 3 to the client and start capturing packets from the client. This can then be done again and again until you have several messages with an encryption using PN as 1, PN as 2, and so on. XORing the packets with matching PNs and PTKs will then give you an XOR of two plaintexts. If one of the plaintexts is known or guessed, then the adversary can derive all packets being sent. This is especially dangerous as an adversary, in certain cases, can decrypt packets to obtain encryption keys and forge arbitrary messages to inject into the communication. This applies only to the TKIP and GCMP encryption schemes however. It has been shown that this cannot work with CCMP, which WPA2-PSK uses [6].

3.3. Phase 3

3.3.1. State-Man-in-the-Middle

Figure 6 shows us that the MITM state is not connected to any other state or attack in this phase, as the attacker is able to go on its own route form here. The attacker in this state has placed himself/herself between the client and a gateway by means of an evil twin attack. In this position, all traffic that would normally be transmitted from the client to a genuine AP first goes through the attackers machine. What makes this position so dangerous and powerful is the fact that the client reconnects with the attacker's AP and performs a handshake to create a session key between them. This allows for simple packet sniffing without the need for a decryption attack. Packets can also be easily altered going to and from the client to perform the attacks in Section 3.4. Being an MITM also creates a possibility of DoS, as the attacker can drop request and response packets going to and from the client.

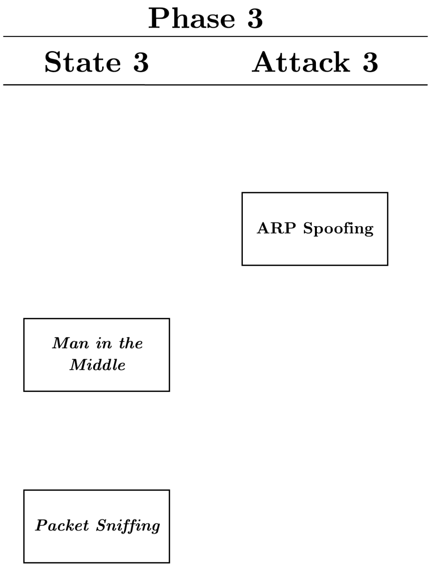

Figure 6. Phase 3.

3.3.2. State-Packet Sniffing

Packet sniffing is the state in which the attacker is able to capture traffic generated by the client and/or AP and decrypt it. Handshake capture decryption can be performed with knowledge of the key, given that the handshake of that client to the AP was captured, while the KRACK attack allows this decryption without knowledge of they key. The packet sniffing state paired with keyed AP session hijacking allows the attacker to perform the Phase 4 attacks on the client. When paired with keyed client session hijacking, the attacker will be able to impersonate the client and request information about the client. Packet sniffing ultimately leads to stolen information due to the decryption of data packets to and from the client.

3.3.3. Attack-ARP Spoofing

ARP (Address Resolution Protocol) is used to map a client's Internet Protocol (IP) address to his/her MAC address in a local network, such as a WLAN. Clients in this protocol have an ARP table that keeps track of all other clients in the network to reference when a packet needs to be sent. When a client joins a network, an ARP packet will be broadcast to all other hosts in the network, requesting them to identify their IP address and MAC address so that the client will be sure to acknowledge with whom he/she is speaking. The other hosts will then send back ARP response packets identifying their IP and MAC addresses. ARP will then form a table in which it will associate all the IP address with the MAC addresses that it learned.

The main drawback of the ARP protocol is that it does not have any authentication procedure before it is accepted into the table. The ARP packet is broadcast in the network; everyone in the network will get the packet; and anyone can reply to that packet. This is called a proxy ARP. Someone else can answer the ARP broadcast, posing as another host. Moreover, a malicious host can send an ARP response irrespective of the requested ARP packets sent or not sent. ARP replies are accepted, and the ARP table will be updated.

ARP spoofing can be used to hijack sessions with the AP and the client. The attacker will send an ARP reply to the client using its own MAC address, while using the IP address of the AP. At the same time, the attacker will send an ARP reply to the AP with his/her own MAC address, using the IP address of the client. This will change the ARP tables of both the AP and the client, thinking he/she has the right MAC addresses for his/her respective IP, but all packets sent for his/her specific IP will be sent to the attacker instead. In this attack, all the traffic flowing between the client and AP will be going through the attacker's machine. This attack can lead to keyed AP session hijacking, keyed AP session hijacking, or both.

3.4. Phase 4

3.4.1. State-Keyed AP Session Hijacking

AP session hijacking from ARP spoofing accomplishes a similar task as the non-keyed AP sessions hijacking explained in Section 3.2.3 in that it fools the client into believing he/she is communicating with a genuine AP, when traffic is being sent to and from an adversary's machine. In this state alone, the adversary, acting as the AP, can choose to not forward the client's requests out, consequently causing a denial of service. ARP messages will constantly be sent to assure that the AP does not try to resolve the problem by sending its own ARP packets to update the client's ARP table. This state can also lead to DNS Spoofing and SSL Stripping, as shown in Figure 7, only under the right conditions. The adversary, at this point in time, has hijacked a session that was in progress between the AP and the client, meaning packets have already been sent and the message counter in the CCMP protocol, shown in Figure 2, would have incremented to an indistinguishable number, making the task of handshake capture decryption and packet sniffing difficult. However, if the state of packet sniffing by handshake capture decryption has already been met, then the adversary would be capturing message packets from the client and AP and know what packet number they are up to. Therefore, when the adversary hijacks the session, he/she will be able to decrypt the client's messages with the correct PTK and PN for CCMP decryption and forge encrypted messages to send to the client to perform the attacks in Section 3.4.

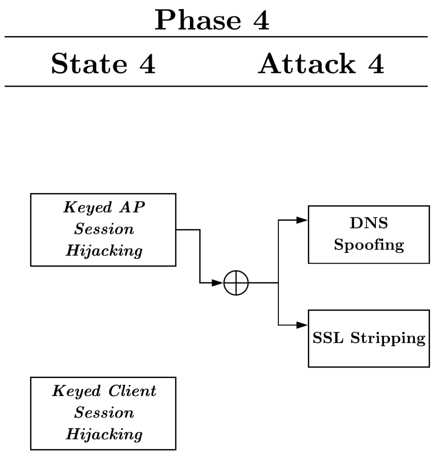

Figure 7. Phase 4.

3.4.2. State-Keyed Client Session Hijacking

This state involves the attacker hijacking a current session that a victim client has with a genuine AP by means of ARP spoofing. The attack is similar to the one discussed in Section 3.4.1 except ARP packets are sent to the AP, instead of a client, to fool the AP into thinking the adversary is another client on the network. Constantly sending ARP packets to the AP to update its ARP table so that the AP sends a victim client's packets to the attackers IP address will cause a DoS on the victim. Likewise, being able to decrypt and forge messages with the previous techniques explained in Section 3.4.1 will allow the attacker to send and receive messages on behalf of the client. This can lead to impersonation and, consequently, stolen information.

3.4.3. Attack-SSL Stripping

When an attacker performs an MITM attack or keyed AP session hijacking with packet sniffing, he/she has access to all traffic between a client and a gateway with the potential to view and manipulate packets. If web traffic is being sent and received using the HyperText Transfer Protocol (HTTP), then the data will be sent in plaintext, and the attacker can capture, read and alter them. However, if the victim uses HTTPS web pages, which is a combination of HTTP and SSL (Secured Socket Layer) protocols, even if the attacker captures the packet, he/she will not be able to read the message because the text is encrypted by the SSL protocol. An attacker can prevent the user, however, from accessing the HTTPS pages and allow the user to access only the HTTP version of the web page with a technique known as SSL stripping.

The client sends an HTTPS request for a web page over the Internet, which is then received by the webserver, who sends back the web page with an encrypted SSL tunnel. If an attacker is proxying all this traffic, however, he/she can alter the request for an HTTP web page, instead of an HTTPS web page. The web page stays the same, but the protocol being used lacks that extra layer of encryption between the host and the webserver. Once the client receives the HTTP page, he/she will try to authenticate with the webserver using his/her credentials; those credentials are in plaintext, and they are captured by the attacker. The attacker will initiate a new HTTPS session using these credentials to the HTTPS server. Then, the server will think that this connection is legitimate and accept it. There are two different sessions that are formed; one is the HTTP session formed between the victim and attacker, and the other is between the attacker and webserver. This will lead to leaked unencrypted credentials and stolen information.

3.4.4. Attack-DNS Spoofing

DNS (Domain Name Server) spoofing is an attack that can be accomplished after an MITM position or a keyed session hijacking with packet sniffing. When a client sends a request in a web browser using a domain name, a DNS request is sent to a DNS server asking for the IP address of the requested domain name. The DNS server will then send the desired IP address back to the client so that the client can send his/her HTTP request to the correct address. DNS spoofing works by intercepting this DNS request, coming as a UDP packet from port 53, and checking the request against a homemade text file with mappings of domain names and IP addresses. For example, when a user is attempting to go to www.example.com, instead of getting the actual IP address of the webserver of that domain name, an attacker can map it to his/her own IP address and host a fake website on his/her webserver. An attack like this can cause such damage as people can create web pages so similar to real web pages, that the user will be deceived into disclosing credentials or personal information. They can also redirect them to a "not found" or "under maintenance" page to cause a DoS.

3.5. Outcomes

As the attacker completes each attack and traverses through the attack flow, he/she will eventually reach an outcome, some paths being quicker than others. Outcomes are the effects of an adversary on a user within an exploited Wi-Fi network. For clarity, each outcome is defined as follows:

Denial of service: preventing a user from accessing the Internet or other specific services through the gateway.
Deception: tricking the user into believing he/she is communicating with one host or gateway, but is actually communicating with the adversary.
Impersonation: taking the users identity and making a remote host or gateway believe the adversary is the user.
Stolen information: taking sensitive information from the user, such as credentials or PII.

4. Wi-Fi Protected Access Version 3

4.1. Overview

Released in June of 2018, WPA3 is the latest security scheme designed to strengthen security in existing Wi-Fi networks and solve the problems the previous versions encountered. WPA3 uses the password-based Simultaneous Authentication of Equals (SAE) technique to authenticate the client to the AP [28]. SAE was a protocol first introduced for use in WLAN mesh networks (IEEE 802.11s) by Dan Harkins in 2008 [29], which was later proved to be vulnerable to passive and active attacks, as well as off-line dictionary attacks, which it claimed to protect against [30]. After a revision of the RFC 7764 standard in 2015 [31], the improved protocol was shown to offer the protection promised [32]. This resistance is achieved using a dragonfly handshake to leverage discrete logarithmic and elliptic curve cryptography. The result of the handshake generates a PMK, which is then used in the standard four-way handshake used in the WPA2 scheme.

The SAE protocol only uses the shared password for authentication, not for deriving the PMK. In the dragonfly protocol, a password element PE is used instead of the password for computing keys. The PE is determined at the time of the session, using an agreed upon set of elliptic curve parameters p, which is a large prime number used to determine the prime field for the elliptic curve, and q, which is another large prime number in the order of a group G, agreed upon by the client and AP using discrete logarithmic computation and a hunting-and-pecking technique with the password as a seed value, described in [31].

A detailed diagram of the dragonfly handshake is provided in Figure 8, demonstrating a peer-to-peer communication between two parties A and B. After elliptic curve parameters are shared and the PE is derived, both parties will then generate a private r and mask m, which are randomly chosen large numbers in the range $\{1...q\}$. They will then use those values to calculate a scalar s and, along with the PE, calculate an element E using the given Equations (7) and (8):

$$s_A = r_A + m_A \mod q \tag{7}$$

$$E_A = (m_A \times PE)^{-1} \tag{8}$$

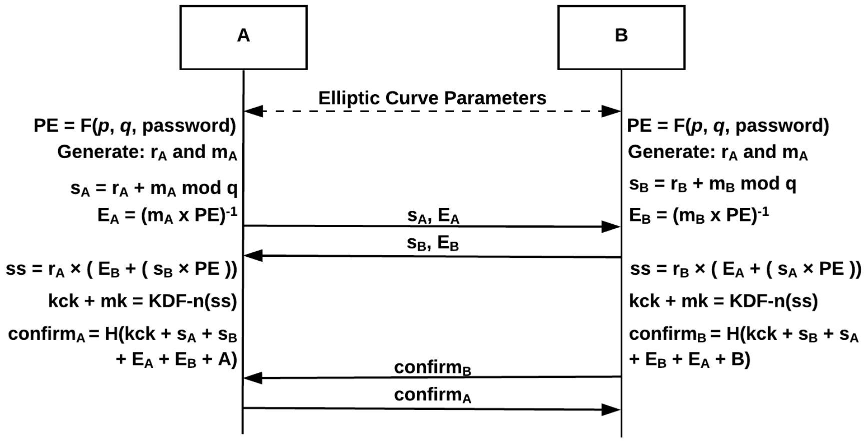
A
B
Elliptic Curve Parameters
PE = F(p, q, password)
Generate: r_A and m_A
$s_A = r_A + m_A$ mod q
$E_A = (m_A$ x PE$)^{-1}$
PE = F(p, q, password)
Generate: r_A and m_A
$s_B = r_B + m_B$ mod q
$E_B = (m_B$ x PE$)^{-1}$
s_A, E_A
s_B, E_B
ss = r_A × (E_B + (s_B × PE))
kck + mk = KDF-n(ss)
confirm$_A$ = H(kck + s_A + s_B + E_A + E_B + A)
ss = r_B × (E_A + (s_A × PE))
kck + mk = KDF-n(ss)
confirm$_B$ = H(kck + s_B + s_A + E_B + E_A + B)
confirm$_B$
confirm$_A$

Figure 8. Dragonfly handshake diagram.

Both parties then send these two calculated values to each other in the first two messages. A will then calculate the shared secret ss by using the information sent by B, such that:

$$ss = r_A \times (E_B + (s_B \times PE)) \tag{9}$$

This can then be further simplified by canceling out operations.

$$ss = r_A \times ((m_B \times PE)^{-1} + ((r_B + m_B) \times PE)) \tag{10}$$

$$ss = r_A \cdot ((m_B \times PE)^{-1} + (r_B \times PE) + (m_B \times PE)) \tag{11}$$

$$ss = r_A \times r_B \times PE = r_B \times r_A \times PE \tag{12}$$

The ss calculated will then be used to derive the key confirmation key kck and the master key mk. The kck will be put into a hash function, concatenated with the sender's scalar, the receiver's scalar, the sender's element, the receiver's element and the identity (in this case, the MAC address) of the sender to confirm that the sender has calculated the correct ss and therefore has knowledge of the password.

$$confirm_A = H(kck + s_A + s_B + E_A + E_B + A) \tag{13}$$

This confirmation message will be calculated on both sides in Messages 3 and 4 with corresponding variables for the sender. The order of concatenation and inclusion of corresponding identity adds authenticity to the message to avoid replay of the other party's message. Finally, mk will be used as the PMK in the 802.11i four-way handshake that follows.

The security of this protocol lies in the intractable nature of the dot product operation in discrete logarithmic computation. By knowing E_A and PE, it is computationally intractable to find m_A [33].

That way, even if an adversary were to compromise the password, he/she cannot use it to derive the PMK himself/herself and decrypt past messages. This provides an element of forward secrecy to the system. Since users need to interact with the AP to derive the fresh PMK each time, attackers can only attempt to obtain the shared password by trying one password at a time, receiving correct or incorrect, then trying again. This level of security strength allows for the user to have a less complicated password for ease of use [28].

The Wi-Fi CERTIFIED Enhanced Open program [34] implemented by the Wi-Fi Alliance applies an extra layer of encryption to each message transmitted between the client and the AP, which allows for private connection in open Wi-Fi networks with no password. This is done by the elliptic curve key exchange explained above, simply without the four-way handshake that follows. Protection Management Frames (PMF), introduced in IEEE 802.11w, are also incorporated to encrypt system management information between the client and AP, so that an adversary cannot spoof management packets (such as de-authentication requests) [35]. A Security Association (SA) mechanism is used to protect the user and AP in the event of an unencrypted management frame. The SA query works by prompting the sender to try the request at a later time within the designated time frame. The AP then sends an encrypted SA request to the sender and waits for an encrypted response. If the sender is already in the network, he/she will be able to send back an encrypted response within the given time. Otherwise, any management frame will be ignored and dropped. 802.11w protects the following management frames [36]:

- Spectrum management
- QoS (Quality of Service)
- DLS (Direct Link Setup)
- Block Ack (Acknowledgment)
- Radio measurement
- Fast BSS (Base Service Station) transition
- SA query
- Protected dual of public action
- Vendor-specific protected

4.2. Security Evaluation and Analysis

As presented in this paper, there are many vulnerabilities in current security measures for WLAN that attackers can leverage to cause all sorts of damage or gain undesired control. Researchers at the Wi-Fi Alliance attempted to update the latest WPA2 system that was in place for 14 years, keeping these vulnerabilities in mind. The release of WPA3 attempted to address these issues and enhance the current state of security. This section will discuss the mitigation techniques for each attack mentioned and whether WPA3 can offer a solution. Table 1 will outline the answer to the question, and more detailed analysis will follow.

Table 1. Attacks against Wi-Fi networks are listed and whether or not WPA3 addresses these attacks.

Attack	Solved by WPA3
Deauthentication	Yes
Handshake Capture Dictionary Attack	Yes
PMKID Hash Dictionary Attack	Yes
Rouge Access Point	Partially
Evil Twin Attack	No
Handshake Capture En/Decryption	Yes
KRACK Exploit	Yes
ARP Spoofing	Partially
SSL Stripping	No
DNS Spoofing	No

In this section, we will go through each attack in an analytical fashion and determine whether WPA3 provides a solution to these vulnerabilities. The format will be as follows: a brief introduction to the attack will be given along with what features could be used to pose a defense, the assumption of the attack and attacker and the proof, which will either support or reject the assumption.

4.2.1. De-Authentication

Here, we will show how WPA3 provides protection to de-authentication attacks with the addition of PMF and SA query. Two cases will be given, followed by a security proof of the resistance.

Case 1

An adversary would be able to send a de-authentication frame to the AP spoofing the MAC address of the client to de-authenticate the client and cut connection with the AP.

Proof. When an AP receives an unencrypted de-authentication or dissociation frame from a client who is already in session, the AP will trigger the SA mechanism and return an error response for the client to try again later given a certain comeback time. The AP will then send an encrypted SA query request to the client and await an SA query response within the response time. The adversary would not be able to send back an encrypted response without the encryption key. Therefore, preforming a de-authentication attack is unfeasible. □

Case 2

An adversary would be able to send a de-authentication frame to one or more clients spoofing the MAC address of the AP to de-authenticate the client and cut connection with the AP.

Proof. When a client receives an unencrypted de-authentication or dissociation frame from the AP who is already in session, the client will send an encrypted SA query request to the AP and await an SA query response within the response time. The real AP will be able to answer with a protected SA query response and ignore any de-authentication frame coming in. Therefore, preforming a de-authentication attack is unfeasible. □

4.2.2. Handshake Capture Dictionary Attack

For off-line dictionary attacks, WPA3 uses the SAE protocol as a defense. The protocol claims to be resistant to passive, active and off-line dictionary attacks.

An adversary would not be able to be able to go through a word list and compute a PMK that comes from the dragonfly handshake to test the MIC of a PTK off-line without interacting with the AP.

Proof. The adversary will first try to capture messages from the dragonfly handshake, where he/she will only obtain E_A, E_B, s_A and s_B, as shown in Figure 8 and defined in Equations (7) and (8). To obtain the PMK used in the four-way handshake, the adversary must compute the shared secret ss, defined in Equation (12), which requires knowledge of r_A and r_B, along with the password element PE. A candidate PE can be derived by brute forcing the password against a word list, which will be used as a seed in a known function given the captured elliptic curve parameters p and q. However, from Equation (8), it is computationally intractable to obtain m_A given E_A and PE. Hence, the adversary would not be able to derive a PMK and PTK to compare to a captured MIC and find the correct password that exists within the word list. Therefore, preforming this off-line dictionary attack is unfeasible. □

4.2.3. PMKID Hash Dictionary Attack

As with the handshake capture dictionary attack, the SAE protocol will defend against this form of off-line dictionary attack.

An adversary would not be able to able to go through a word list and compute a PMKID that comes from the dragonfly handshake to test and compare against a candidate PMKID to derive the passphrase without actively going through the dragonfly handshake.

Proof. The AP does not have a static PMK derived from the PSK. Instead, the PMK comes from the dragonfly handshake, which requires client interaction. Therefore, the PMKID would not be available until after a valid execution of the dragonfly handshake. This is not feasible, as shown in the handshake capture dictionary attack proof. Therefore, preforming this off-line dictionary attack is unfeasible. □

4.2.4. Rogue Access Point

Rouge APs are set up to deceive the user to connect to a false router that mimics a genuine one. With the use of PMF, some protection is given, but this problem persists. We break down the rouge AP analysis into two sections: key acquisition and AP session hijacking. The first will describe the scenario where an adversary attempts to obtain a key using a malicious AP given two cases where the client is either already connected or not connected yet. The second will demonstrate how an adversary attempts to hijack the session from the AP, making the client think he/she is talking to a genuine AP and not a malicious AP using two techniques.

Key Acquisition

Case 1: Client Connected

An adversary would be able to set up a malicious AP that impersonates the genuine APs SSID and MAC address, as well as the correct security protocol with the wrong passphrase in an attempt to have the user input the passphrase. The adversary will then not be able to de-authenticate an already connected client and have them reconnect to the malicious AP instead.

Proof. The adversary will identify the target AP and record its SSID, MAC address and security protocol. The adversary will wait until the client connects to the genuine AP to capture the handshake. The adversary will then set up a rouge AP that matches the configurations of the target AP by spoofing the SSID and MAC address. The adversary will then send de-authentication packets to the target client to cut the connection with the genuine AP. As demonstrated in the de-authentication proof, WPA3 will not allow this to happen. The adversary is then forced to wait or abort the attack. □

Case 2: Client Not Connected

Continuing from the scenario in Case 1, the adversary will then wait for the client to try to connect to the genuine AP and have them reconnect to the malicious AP instead by offering a stronger signal.

Proof. The adversary will identify the target AP and record its SSID, MAC address and security protocol. The adversary will wait until the client connects to the genuine AP to capture the handshake. The adversary will then set up a rouge AP that matches the configurations of the target AP by spoofing the SSID and MAC address. The adversary will strengthen the APs broadcast signal and wait for the client to connect. Once the connection is made, before the handshake, the AP will redirect the user to a landing page asking him/her to confirm the passphrase. The adversary will then take the plaintext entries, calculate the PMK, PTK and MIC to compare to the captured handshake and find the correct key. Therefore, preforming a rogue access point attack to obtain network keys is feasible. □

AP Session Hijacking

Technique 1: Physical Connection and ARP Spoofing

An adversary, assuming having connected a malicious AP physically to a network through wired Ethernet, would not be able to send ARP packets to trick the client into thinking it is the real gateway.

Proof. The adversary will then send an ARP packet to the client spoofing the AP's MAC address with its own IP. The adversary will encrypt the message with the GTK, so the client can decrypt the message with the same GTK, exploiting the Hole 196 vulnerability. A WPA3 router with client isolation turned on, however, will not allow clients in a network to communicate with each other, or know about each other for that matter. Therefore, performing this attack to hijack a client session from a genuine AP is unfeasible. □

Technique 2: Wireless Channel Switching

An adversary would not be able to send a message to the client to switch AP channels to the malicious AP or de-authenticate from the real AP.

Proof. The adversary will set up an evil twin imitating the AP to which the client is connected. The adversary will send a CSA beacon to switch channels from the legitimate AP to the malicious AP. The PMF system should protect against this kind of message as it is under spectrum management [35] and therefore is protected. If it is not, however, the client will try to switch over to the malicious AP. The adversary will then try to de-authenticate the client to the AP to avoid any interference from the AP. We have shown in the de-authentication proof that this cannot happen. Therefore, performing this attack to hijack a client session from a genuine AP is unfeasible. □

4.2.5. Evil Twin Attack

An evil twin is a malicious AP that attempts to trick a user into connecting to it by cloning a genuine AP and offering a better signal in hopes the client will connect to it instead. Once the client is deceived and connects to a malicious AP, the WPA3 protocol is out of its scope of protection. This section gives two cases where the client is either already connected or not connected yet.

Case 1: Client Connected

An adversary would be able to set up a malicious AP that impersonates the genuine AP's SSID and MAC address, as well as the correct security protocol and passphrase being used to create an MITM between the client and the Internet. The adversary will then be able to de-authenticate an already connected client and have him/her reconnect to the malicious AP instead by offering a stronger signal.

Proof. The adversary will use a wireless adapter and set it to monitor mode to observe wireless traffic in the air between clients and APs. The adversary will identify the target AP and record its SSID, MAC address and security protocol. The adversary will then set up a rouge AP that matches the configurations of the target AP by spoofing the SSID and MAC address and setting the passphrase to be the same as the genuine AP. The adversary will then send de-authentication packets to the target client to cut connection with the genuine AP. As demonstrated in the de-authentication proof, WPA3 will not allow this to happen. The adversary is then forced to wait or abort the attack. □

Case 2: Client Not Connected

Continuing from the scenario in Case 1, an adversary will then wait for the client to try to connect to the genuine AP and have him/her reconnect to the malicious AP instead by offering a stronger signal.

Proof. The adversary will use a wireless adapter and set it to monitor mode to observe wireless traffic in the air between clients and APs. The adversary will identify the target AP and record its SSID, MAC address and security protocol. The adversary will then set up a rouge AP that matches the configurations of the target AP by spoofing the SSID and MAC address and setting the passphrase to be the same as the genuine AP. The adversary will strengthen the APs broadcast signal and wait for the client to connect. The client will enter the same passphrase shared with the genuine AP. This will create a trusted connection with the malicious AP where the adversary can decrypt all traffic using the PTK. Once out of the network, the WPA3 protocol no longer protects the client's data. Therefore, preforming a rogue access point attack for creating an MITM is feasible. □

4.2.6. Handshake Capture En/Decryption

In WPA2, an adversary was able to capture the two random nonces generated in the four-way handshake and sent over plaintext and use them, along with the passphrase, to derive the PTK and decrypt traffic. The WPA3 protocol uses the SAE protocol, which utilizes both the dragonfly handshake and the four-way handshake.

An adversary would not be able to capture information from the two handshakes and derive a PTK with knowledge of the password for a specific client to decrypt traffic.

Proof. The adversary will first try to capture messages from the dragonfly handshake, where he/she will only obtain E_A, E_B, s_A and s_B, as shown in Figure 8 and defined in Equations (7) and (8). To obtain the PMK used in the four-way handshake, the adversary must compute the shared secret ss, defined in Equation (12), which requires knowledge of r_A and r_B, along with the password element PE. The PE can be derived by using the known password as a seed in a known function given the captured elliptic curve parameters p and q. However, from Equation (8), it is computationally intractable to obtain m_A given E_A and PE. Hence, the adversary would not be able to derive a PMK and PTK to compare to a captured MIC and find the correct password that exists within word list. Therefore, preforming handshake capture decryption is unfeasible. □

4.2.7. KRACK Exploit

The KRACK exploit leverages the venerability of resending Message 3 in the four-way handshake. Patches to APs and devices have been released to not allow this retransmission.

An adversary would not be able to manipulate messages between the client and AP after hijacking a session from the AP to replay Message 3 of the four-way handshake, reinitialize the keys and reset the keystream.

Proof. Assuming this is not necessary, the attacker will then take over the session and resubmit Message 3 and start capturing packets for decryption. With updated security patches and configurations, a WPA3 router can be set up to not allow the retransmission of Message 3, which is integral to the attack. Therefore, preforming the KRACK attack is unfeasible. □

4.2.8. ARP Spoofing

ARP spoofing can give an adversary the advantage of being an MITM between a client and a gateway, like an AP, or hijacking a session. WPA3 only partially addresses this issue. We break down this proof into two cases: (1) to show that session hijacking is possible and (2) to show that acquiring a MITM position is not possible.

Case 1: Client Session Hijacking

An adversary will be able to send spoofed ARP packets to the AP impersonating the client to hijack the session.

Proof. The adversary will send an ARP packet to the AP spoofing the targeted client's MAC address with its own IP. Since there is no authentication protocol for ARP requests, the AP will accept this and update its ARP table to forward packets for the targeted client to the adversaries IP. Therefore, performing this attack to hijack a session is feasible. □

Case 2: MITM

An adversary will not be able to send spoofed ARP packets to both the AP and client, impersonating both of them to each other, and create an MITM position.

Proof. The adversary will take the same steps as the previous case. In addition, the adversary will then send an ARP packet to the client spoofing the AP's MAC address with its own IP. The adversary will encrypt the message with the GTK, so the client can decrypt the message with the same GTK, exploiting the Hole 196 vulnerability. A WPA3 router with client isolation turned on, however, will not allow clients in a network to communicate with each other, or know about each other for that matter. Therefore, performing this attack to create an MITM position is unfeasible. □

4.2.9. SSL Stripping

The SSL stripping attack deals with data packets being sent over the Internet using the HTTP and HTTPS protocols. This is a layer 7 attack and out of scope for a WPA3 router on layer 3 to provide protection.

An adversary in an MITM position would be able to able to manipulate the HTTPS requests from the client to be HTTP requests and have the server return the HTTP version of the web page. Any information entered by the user is then not protected by the SSL encryption of HTTPS and sent in plaintext.

Proof. By the nature of the attack, the adversary must have already gained access to the network and key to be an active MITM. Therefore, the adversary is also able to decrypt all traffic encrypted using WPA3 encryption. The adversary will capture all HTTPS request coming from the client and decrypt the messages. The adversary will then change the request to be HTTP and forward it through to the router. The router will decrypt and send the request to the server. The server will then respond with an HTTP response page. The router will encrypt the response and send it to the client, which will be caught by the adversary. The adversary will then forward the HTTP page to the client. The client, without realizing, will receive the HTTP page and begin to interact with it. The traffic generated will be in plaintext, after decrypting, and viewed by the adversary. Therefore, an SSL strip attack is still feasible. □

4.2.10. DNS Spoofing

DNS spoofing is a simple attack, but relies on acquiring an MITM position. Once the attacker gains access and places himself/herself between the gateway and the client, there is no further protection WPA3 can offer.

An adversary in an MITM position would be able to manipulate the HTTPS requests from the client to be HTTP requests and have the server return the HTTP version of the web page. Any information entered by the user is then not protected by the SSL encryption of HTTPS and sent in plaintext.

Proof. By the nature of the attack, the adversary must have already gained access to the network and key to be an active MITM. Therefore, the adversary is also able to decrypt all traffic encrypted using WPA3 encryption. The adversary will see that the client made a DNS request for a certain domain name. The adversary will then forge a DNS response and encrypt it with the wrong IP address for the requested domain. The client will have no suspicion not to trust the encrypted DNS response and go to that IP address. Therefore, this attack is still feasible. □

5. Discussion

WPA3 offers a more resilient security scheme than its predecessor, WPA2, by adding features like the dragonfly handshake and protected frame management, among others. We have shown in the previous section how WPA3 was able to address certain attacks that were performed on a Wi-Fi network with WPA2 and how it is still vulnerable to others. Figure 9 shows an updated version of Figure 3, which shows the attack paths that are still possible after implementing WPA3 based on the security analysis provided above.

We have demonstrated that with the addition of PMF, de-authentication attacks are no longer possible and therefore cannot be performed to accomplish a DoS. Brute force attacks, or off-line dictionary attacks, are also shown to be impossible due to the addition of the SAE protocol and dragonfly handshake. This leaves only one option for an attacker against a Wi-Fi network, a rogue access point attack. WPA3 partially addresses this attack in that it prevents the use of unauthorized de-authentication and CSA flags to gain a non-keyed AP session hijacking position by using SA query. However, it does not prevent the attacker from using a rogue access point to phish a user into disclosing the passphrase of a genuine AP (it is important to not that this is just one physical phishing method for key acquisition, and more techniques exist, such as social engineering or careless protection of the passphrase). After a rogue access point attack, the attacker will have acquired the network key and can either genuinely join the network or set up an evil twin.

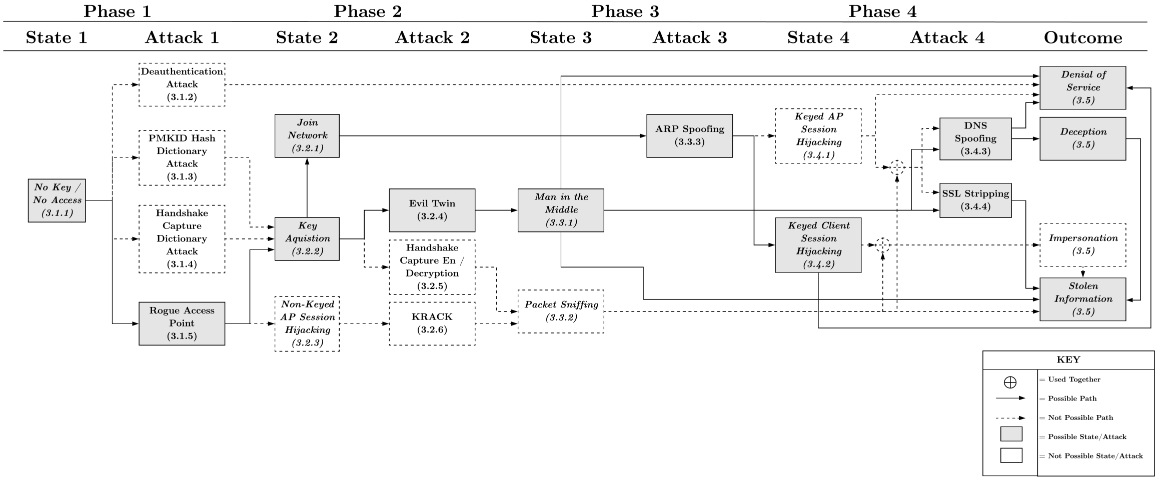

Figure 9. Post attack flow diagram with corresponding section numbers.

Since there are no more known methods to hijack a session from the AP without a key, there would be no more use in performing the KRACK attack to decrypt and encrypt packets for packet sniffing. There have also been many client side patches available for devices that do not allow the resending of Message 3 in the four-way handshake (Figure 1), making most devices resilient to the KRACK attack already. Likewise, we have demonstrated that even with knowledge of the passphrase, the SAE protocol provides forward secrecy for WPA3, and encryption and decryption of packets from a handshake captures are no longer possible. This leaves no paths to packet sniffing, making WPA3 free from unauthorized viewing or tampering of data sent between a client and a genuine AP.

An attacker, however, can still join the network and partially perform ARP spoofing. WPA3 prevents the attacker from exploit Hole 196 and sending messages to other users in the network using the GTK by implementing client isolation. Therefore, an attacker will not be able to hijack a session from the AP with knowledge of the key to cause DoS and launch DNS spoofing and SSL stripping attacks from this path. However, the attacker is still able to send an ARP message to the AP and hijack the session from the client with knowledge of the key. In WPA2, this was a dangerous position in that the attack could use a packet sniffing method to impersonate the client and steal information. Since packet sniffing is no longer possible in WPA3, all the attacker can do is cause a DoS to the user by constantly hijacking the session and not allowing hm/her to communicate with the AP.

As mentioned above, the attacker can also set up an evil twin to trick the user into connecting to his/her router with the same key as the genuine router. Once this happens, the client engages in a handshake with the malicious router and sets up his/her own PTK. This will allow the attacker to have access to all decrypted traffic from the user. There is currently no protection for this issue, and it requires research for future solutions. After executing this attack, the attacker becomes an MITM and is able to deny service or steal information by viewing the packets being sent. The attacker is also able to perform DNS spoofing and SSL stripping by viewing the messages sent by the user and sending forged encrypted messages back. These attacks lead to deception and stolen information.

6. Other Defenses and Mitigations

We have shown and concluded that the WPA3 security scheme addresses many of the vulnerabilities present in its predecessor, WPA2, but not all. Figure 9 shows that there are still paths an attacker can take to achieve the desired outcomes. Aside from the defenses present in WPA3, we will also look at other defenses and mitigations to the vulnerabilities still present after the application of WPA3.

6.1. Rogue Access Point

A distinction needs to made between the on-wire rogue access point attacks and the rogue access points used by attackers on the client side. When used on-wire, attackers find physical access to a network through a port and connect their own routers to the network. Now, they have created a wireless gateway that they can use to access the targeted network. Much research exists on detecting and preventing attack, but this is not the attack that we are dealing with in this paper. On the client side, attackers will use an access point that connects to the Internet by other means outside of a private or public network, then aim to deceive the client to connect to it by manipulating MAC and SSID information on the rogue AP. In this paper, we discuss the implementation of the latter.

Rogue access points after WPA3 protection for now only seem to serve the purpose of phishing a network key out of a client. This attack falls into the same category of other phishing attacks that involve social engineering or physical theft. For protection against this attack, client education is the best answer. Users must become more aware of suspicious web pages of untrusted networks. In this day and age, people sacrifice security for convenience, which can lead to serious punishments. Protection of the network key should be of higher concern when connecting to networks. Other prevention techniques that are shown in research involve comparing gateways to each other and the routes that packets travel, as in [37]. Being that this is the beginning of the remaining attack path, it is crucial that more research be done in this field to prevent the possibility of the attacks that follow.

6.2. Evil Twin

This attack can be very destructive due to its intrusive and controlling nature. Once you have unknowingly connected to the malicious AP, an attacker could monitor your browsing, steal sensitive information such as credit cards, passwords, etc., or even inject packets to cause damage. It is important that users practice secure techniques to protect themselves from malicious adversaries.

The first recommendation is to the user; try to stay away from public networks, both open and secure. As we have seen, attackers are able to break current security protocols and decrypt traffic in WLANs once the passphrase has been cracked. Ensure that you are on a trusted network before browsing on Wi-fi. If Internet access is needed, you can try to connect your machine to your personal phone's Wi-Fi hotspot, if that option is available, to forward Internet and to know you are on a safe network.

Another good defense and safe practice is to use a Virtual Private Network (VPN). VPNs create an encrypted channel between your machine and a network, to allow you to create a persistent, secure connection to a remote network. That way, you will not have to rely on a suspicious WLAN near you and can rest easy knowing your information is being encrypted either way.

6.3. ARP Spoofing

Unfortunately, ARP spoofing continues to be a problem in both wired and wireless networks. The main drawback is in the fact that ARP messages are unauthenticated and can be sent by anyone within the network, prompting an update of a host's ARP table. The first line of defense for this attack is preventing attackers from entering the network, as discussed previously in this paper. However, given that an attacker has entered the network, one can still try to protect one's self from the inside. Client isolation helped to protect the clients within the network, but still leaves the AP vulnerable. There have been a few methods in the literature that were proposed to prevent this attack, including modifying the protocol, using a browser application and implementing a specific network architecture [38–40]. Future research is still required to create a standard for protection.

6.4. SSL Stripping

SSL stripping is an attack that will be performed after the MITM position is assumed. In today's society, companies are becoming more aware of cyber threats and try to protect their customers as much as possible. Most web pages, especially those that deal with highly sensitive data, will only have HTTPS certified pages and not offer an HTTP version. If the site does, however, have an HTTP version of the web page, the web application can include an HTTP Strict Transport Security (HSTS) header, which tells the receiving browser to only use HTTPS and not allow any HTTP requests [41]. However, there are still many sites out there that lack this proper security posture. One should be aware of the sites one is visiting and try to recognize when one is browsing on an insecure version of a site. Once again, a VPN will encrypt all Internet traffic to protect any private credentials that SSLstrip could capture.

6.5. DNS Spoofing

DNS spoofing can also be achieved after an MITM attack is performed. This involves capturing the DNS request, preventing it from going through to the DNS server and giving a custom spoofed DNS response to the client. This is possible because DNS traffic is not encrypted, so that the Internet service provider and DNS server can read and direct your message. You can, however, use a VPN server that has a DNS server within it to make your DNS request and ensure that an MITM cannot read or spoof your requests and responses.

7. Conclusions

Wireless technology has come a long way since 1997 to provide us with efficient means to send and receive data with no physical wired connections. In the beginning, security was not as much of a concern, but as time goes on, attacks on wireless networks are becoming more and more prevalent as more people are educating themselves in this sector. Security schemes need to adapt to stay up to date with new threats to provide as much security to users as possible. Until now, we have seen three security schemes, WEP, WPA and WPA2, and showed that has its own vulnerabilities that attackers can exploit. It is important to understand past, discontinued schemes to be able to create new, more secure schemes, like WPA3. The new scheme implemented fixes many of the issues present in WPA2, including de-authentication, off-line dictionary attacks and the KRACK vulnerability, but falls short of solving some of the major vulnerabilities in Wi-Fi networks. However, there are defenses and safe practices one can take, such as VPN use, to help stay secure even in the face of these threats. This paper hopes to clarify current research by displaying current attacks on Wi-Fi networks in an organized manner. It also hopes to serve as a base for future research and to be added upon as new attacks emerge.

Author Contributions: Formal analysis, C.K. and T.H.; Investigation, T.H.; Methodology, C.K.; Project administration, T.H.; Supervision, T.H.; Writing—original draft, C.K. and T.H.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Ack	Acknowledgment
AES	Advanced Encryption Standard
AP	Access Point
ARP	Address Resolution Protocol
BSS	Base Service Station
CBC	Cipher-Block Chaining
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CRC	Cyclic Redundancy Check
CSA	Channel Switch Announcement
CTR	Counter Mode
DLS	Direct Link Setup
DNS	Domain Name Server
DoS	Denial of Service
EAP	Extensible Authentication Protocol
EAPOL	EAP over LAN
GTK	Group Temporal Key
HMAC	Hash Message Authentication Protocol
HTTP	HyperText Transfer Protocol
HSTS	HTTP Strict Transport Security
ICV	Integrity Check Value
IEEE	Electrical and Electronics Engineers
IP	Internet Protocol
IV	Initialization Vector
KRACK	Key Re-installation Attack
MAC	Medium Access Control
MIC	Message Integrity Code
MITM	Man-in-the-Middle
PBKDF2	Password-Based Key Derivation Function 2
PMF	Protected Management Frames
PMK	Pairwise Master Key
PMKID	Pairwise Master Key Identification
PN	Packet Number
PRNG	Pseudo Random Number Generator
PSK	Pre-Shared Key
PTK	Pairwise Transient Key
QoS	Quality of Service
RC4	Rivest Cipher 4
RF	Radio Frequency
RSN	Robust Security Network Information Element
SA	Security Association
SAE	Simultaneous Authentication of Equals
SSID	Service Set Identifier
SSL	Secured Socket Layer
TK	Temporal Key
TKIP	Temporal Key Integrity Protocol
TSC	TKIP Sequence Counter
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Networks
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access II
WPA3	Wi-Fi Protected Access III

References

1. Gast, M. *802.11 Wireless Networks: The Definitive Guide*; O'Reilly Media, Inc.: Newton, MA, USA, 2005.
2. Vanhoef, M.; Piessens, F. Key reinstallation attacks: Forcing nonce reuse in WPA2. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1313–1328.
3. Steube, J. New Attack on WPA/WPA2 Using PMKID. Available online: https://hashcat.net/forum/thread-7717.html (accessed on 30 October 2018).
4. Simic, D.; Prodanovic, R. A survey of wireless security. *J. Comput. Inf. Technol.* **2007**, *15*, 237–255.
5. Zou, Y.; Zhu, J.; Wang, X.; Hanzo, L. A survey on wireless security: Technical challenges, recent advances, and future trends. *Proc. IEEE* **2016**, 1–39. [CrossRef]
6. Fouque, P.A.; Martinet, G.; Valette, F.; Zimmer, S. On the Security of the CCM Encryption Mode and of a Slight Variant. In Proceedings of the International Conference on Applied Cryptography and Network Security, New York, NY, USA, 3–6 June 2008; pp. 411–428.
7. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). In Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 8–11 August 2009; pp. 48–52.
8. Stubblefield, A.; Ioannidis, J.; Rubin, A.D. A key recovery attack on the 802.11 b wired equivalent privacy protocol (WEP). *ACM Trans. Inf. Syst. Secur.* **2004**, *7*, 319–332. [CrossRef]
9. Tews, E.; Weinmann, R.P.; Pyshkin, A. Breaking 104 bit WEP in less than 60 seconds. In *International Workshop on Information Security Applications*; Springer: Berlin, Germany, 2007; pp. 188–202.
10. Tews, E.; Beck, M. Practical attacks against WEP and WPA. In Proceedings of the Second ACM Conference on Wireless Network Security, Zurich, Switzerland, 16–19 March 2009; pp. 79–86.
11. Mavoungou, S.; Kaddoum, G.; Taha, M.; Matar, G. Survey on threats and attacks on mobile networks. *IEEE Access* **2016**, *4*, 4543–4572. [CrossRef]
12. Tekade, P.S.; Shelke, C. A Survey on different Attacks on Mobile Devices and its Security. *Int. J. Appl. Innov. Eng. Manag.* **2014**, *3*, 247–251.
13. Sen, J. A survey on wireless sensor network security. *arXiv* **2010**, arXiv:1011.1529.
14. Wang, Y.; Attebury, G.; Ramamurthy, B. *A Survey of Security Issues in Wireless Sensor Networks*; IEEE Press: Piscataway, NJ, USA, 2006.
15. Walters, J.P.; Liang, Z.; Shi, W.; Chaudhary, V. Wireless sensor network security: A survey. *Secur. Distrib. Grid Mob. Perv. Comput.* **2007**, *1*, 367.
16. Christin, D.; Mogre, P.S.; Hollick, M. Survey on wireless sensor network technologies for industrial automation: The security and quality of service perspectives. *Future Internet* **2010**, *2*, 96–125. [CrossRef]
17. Akyildiz, I.F.; Wang, X. A survey on wireless mesh networks. *IEEE Commun. Mag.* **2005**, *43*, S23–S30. [CrossRef]
18. Sukhija, S.; Gupta, S. Wireless network security protocols a comparative study. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 357–364.
19. Juwaini, M.; Alsaqour, R.; Abdelhaq, M.; Alsukour, O. A review on WEP wireless security protocol. *J. Theor. Appl. Inf. Technol.* **2012**, *40*, 39–43.
20. Vanhoef, M.; Piessens, F. Practical verification of WPA-TKIP vulnerabilities. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, Hangzhou, China, 8–10 May 2013; pp. 427–436.
21. Tripathi, A.; Damani, O.P. Relative encryption overhead in 802.11 g network. In Proceedings of the International Symposium on Telecommunications, Tehran, Iran, 27–28 Auguet 2008; pp. 420–423.
22. Ferreira, R.A. A Probability Problem Arising from the Security of the Temporal Key Hash of WPA. *Wirel. Pers. Commun.* **2013**, *70*, 1235–1241. [CrossRef]
23. Han, W.; Zheng, D.; Chen, K.f. Some remarks on the TKIP key mixing function of IEEE 802.11 i. *J. Shanghai Jiaotong Univ. (Sci.)* **2009**, *14*, 81–85. [CrossRef]
24. Sheldon, F.T.; Weber, J.M.; Yoo, S.M.; Pan, W.D. The insecurity of wireless networks. *IEEE Secur. Priv.* **2012**, *10*, 54–61. [CrossRef]

25. Cebula, S.L.; Ahmad, A.; Wahsheh, L.A.; Graham, J.M.; DeLoatch, S.L.; Williams, A.T. How secure is WiFi MAC layer in comparison with IPsec for classified environments? In Proceedings of the 14th Communications and Networking Symposium, Society for Computer Simulation International, Boston, MA, USA, 3–7 April 2011; pp. 109–116.
26. Bellardo, J.; Savage, S. *802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions*; Usenix Security Symposium: Washington, DC, USA, 2003; Volume 12, p. 2.
27. Kumkar, V.; Tiwari, A.; Tiwari, P.; Gupta, A.; Shrawne, S. Vulnerabilities of Wireless Security protocols (WEP and WPA2). *Int. J. Adv. Res. Comput. Eng. Technol.* **2012**, *1*, 34.
28. Wi-Fi Certified WPA3 Technology Overview. Available online: https://www.wi-fi.org/downloads-registered-guest (accessed on 30 October 2018)
29. Harkins, D. Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks. In Proceedings of the Second International Conference on Sensor Technologies and Applications, Cap Esterel, France, 25–31 August 2008; pp. 839–844.
30. Clarke, D.; Hao, F. Cryptanalysis of the dragonfly key exchange protocol. *IET Inf. Secur.* **2014**, *8*, 283–289. [CrossRef]
31. Harkins, D. Dragonfly Key Exchange; *Aruba Networks*. Available online: https://tools.ietf.org/html/rfc7664 (accessed 30 October 2018).
32. Lancrenon, J.; Škrobot, M. On the Provable Security of the Dragonfly protocol. In Proceedings of the International Information Security Conference, Trondheim, Norway, 19–21 September 2015; pp. 244–261.
33. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
34. Wi-Fi Certified Enhanced Open Technology Overview. Available online: https://www.wi-fi.org/beacon/dan-harkins/wi-fi-certified-enhanced-open-transparent-wi-fi-protections-without-complexity (accessed on 30 October 2018).
35. Ahmad, M.S.; Tadakamadla, S. Short paper: Security evaluation of IEEE 802.11 w specification. In Proceedings of the fourth ACM conference on Wireless network security, Hamburg, Germany, 14–17 June 2011; pp. 53–58.
36. Cisco. *802.11w Protected Management Frames*; Cisco: San Jose, CA, USA, November 2017.
37. Nikbakhsh, S.; Manaf, A.B.A.; Zamani, M.; Janbeglou, M. A novel approach for rogue access point detection on the client-side. In Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Fukuoka, Japan, 26–29 March 2012; pp. 684–687.
38. Agrawal, N.; Pradeepkumar, B.; Tapaswi, S. Preventing ARP spoofing in WLAN using SHA-512. In Proceedings of the Computational Intelligence and Computing Research (ICCIC), Enathi, India, 26–28 December 2013; pp. 1–5.
39. Behboodian, N.; Razak, S.A. ARP Poisoning Attack Detection and Protection in WLAN via Client Web Browser. In Proceedings of the International Conference on Emerging Trends in Computer and Image, Bangkok, Thailand, 23–24 December 2011; p. 20.
40. Cisco. *Wireless and Network Security Integration Solution Design Guide*; Cisco: San Jose, CA, USA, December 2013.
41. Clark, J.; van Oorschot, P.C. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP), Berkeley, CA, USA, 19–22 May 2013; pp. 511–525.