SA 2024

Wi-Fi Security Threats - an Integrative Review

Version 1.0
December 17, 2024
Mario Burger, Alice Glaus

1 Abstract

"What are possible threats against Wi-Fi infrastructure?" is the main research question this integrative literature-review wants to find an answer to. Whether those threats are adequately dealt with, and how impactful the real-world implications appeared to be, are additional research goals. It was aimed to classify and illustrate those threats and attacks in an attack tree, demonstrating the possible attack vectors and paths.

The literature-review is based on scientific sources (papers, articles) in the field of vulnerabilities in Wi-Fi security, protocol weaknesses and flaws, and cyber-attacks on Wi-Fi networks. The sources were selected based on a set of criteria and keywords, such as year and type of publication, and publication libraries.

In a first part each selected source is summarized, highlighting their individual focus on the discussed threats or attacks and the resulting findings. The second part, the literature-review, consists of interwove comparisons of the sources' topics and findings, categorized by threats or attack types.

The literature-review concludes that many of the presented threats and attacks are enabled by inherent vulnerabilities in Wi-Fi protocols or implementation flaws. Some vulnerabilities may have been partially addressed in amendments to Wi-Fi standards, while others persist due to backward compatibility requirements.

Regarding improvements and future fields of study, the literature-review recognized the need for more rigorously defined standards in Wi-Fi technology. Implementations should be formally verified in a way to eliminate lacking adherence to standards and to reduce risks of bugs. Testing of Wi-Fi implementations must be expanded to include a broader range of devices, real-world environments, and configurations. This includes vendor-specific features and implementations, which often rely on Wi-Fi standards but due to ambiguous specifications lack security.

2 Management Summary

2.1 Objective

"What are possible threats against Wi-Fi infrastructure?" is the main research question we wanted to answer in an integrative literature-review. Whether those threats are adequately dealt with, and how impactful the real-world implications appeared to be, are additional research goals.

2.2 Approach

The attack tree shows possible attack vectors and paths. With this information we collected papers published in the last seven years (2017-2024) to create an integrative literature-review. In a first part each selected source was summarized, highlighting their individual focus on the discussed threats or attacks and the resulting findings. The second part, the literature-review, consists of interwove comparisons of the sources' topics and findings, categorized by threats or attack types.

Additionally, we conducted an experiment in which a 4-way handshake between an access point and a client was recorded by a third party. This experiment laid the foundation for further tests and implementations, which will be carried out in the Bachelor's thesis in the next semester.

2.3 Conclusion

The literature-review concludes that many of the presented threats and attacks are enabled by inherent vulnerabilities in Wi-Fi protocols or implementation flaws. Some vulnerabilities may have been partially addressed in amendments to Wi-Fi standards, while others persist due to backward compatibility requirements.

Regarding improvements and future fields of study, the literature-review recognized the need for more rigorously defined standards in Wi-Fi technology. Implementations should be formally verified in a way to eliminate lacking adherence to standards and to reduce risks of bugs. Testing of Wi-Fi implementations must be expanded to include a broader range of devices, real-world environments, and configurations. This includes vendor-specific features and implementations, which often rely on Wi-Fi standards but due to ambiguous specifications lack security.

Contents

3 Introduction

The IEEE 802.11 standard, foundational for Wi-Fi, was originally developed in 1997 by the Institute of Institute of Electrical and Electronics Engineers (IEEE) and they still continue to maintain and expand the standard [1]. The Wi-Fi Alliance, founded in 1999, is responsible for promoting and certifying Wi-Fi technology based on the IEEE 802.11 standard. They focus on ensuring interoperability across devices and managing the "Wi-Fi Certified" program, which validates that products meet compatibility, performance, and security standards for the various versions of Wi-Fi technology [2].

Wireless networks are nowadays widely used across various environments, making network security a critical factor to ensure reliable performance and safeguarding user privacy. The terms Wireless Local Area Network (WLAN) and Wi-Fi are often used interchangeably to refer to wireless networks. However, to be precise, 'Wi-Fi CERTIFIED™' indicates that a product meets the industry standards defined by the Wi-Fi Alliance [3].

3.1 History of Wi-Fi Security Standards

The first security standard for Wi-Fi devices, Wired Equivalent Privacy (WEP), was replaced by a new standard, Wi-Fi Protected Access (WPA), in 2003. WEP can be compromised and therefore is no longer considered secure [4].

In 2006 the Wi-Fi Alliance announced that WPA2 is now mandatory after being an optional program since 2004. It is based on the IEEE 802.11i standard and uses Advanced Encryption Standard (AES) to encrypt data [5].

At the time of this writing the latest standard WPA3 was introduced by the Wi-Fi alliance in 2018 to enhance security. They state that WPA3 enhances network protection by simplifying security protocols, strengthening authentication, and increasing cryptographic robustness, particularly for sensitive data. Even though WPA3 is the recommended standard to use by the Wi-Fi Alliance, the support of WPA2 is still mandatory for all Wi-Fi CERTIFIED devices [6].

3.2 Paper Structure

This integrative literature review aims to highlight the most impactful vulnerabilities in Wi-Fi networks. In section 4 we describe our approach to creating the review. We will give an overview of the reviewed papers by summarizing their findings briefly in section 5. Then we connect the attacks described in the papers with each other in section 6 before finalizing our paper with a conclusion in section 7.

4 Methods

4.1 Research Methodology

In this paper our goal is to provide an overview of security vulnerabilities in Wi-Fi networks and standards in the style of an integrative review. We will use an attack tree to group the reviewed papers by the attack vector paths required to compromise a Wi-Fi network.

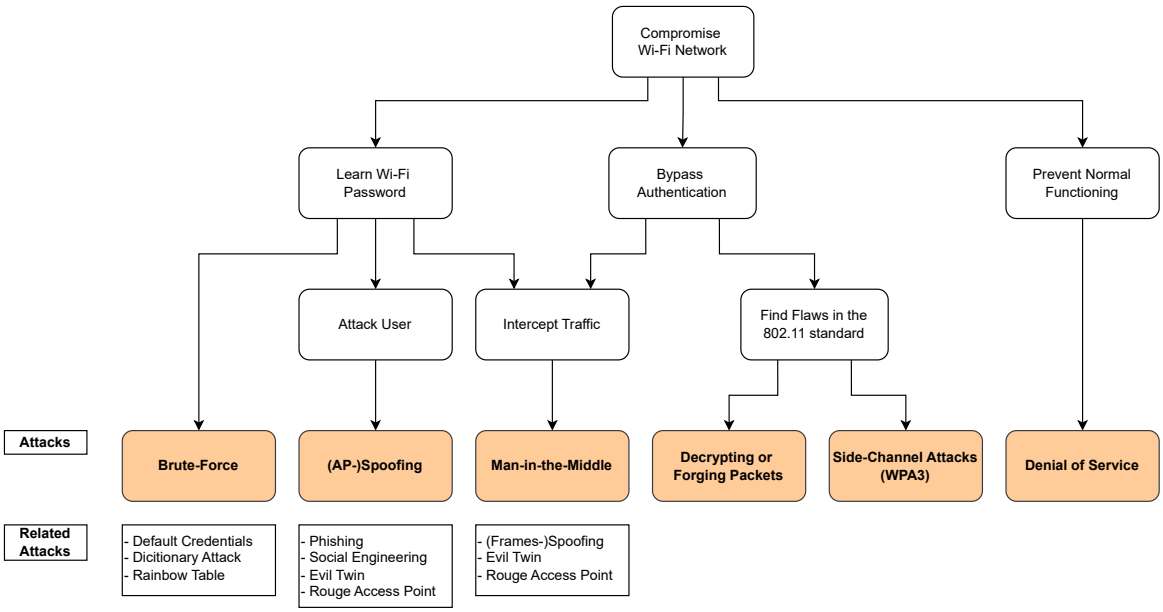

Figure 4.1: Attack Tree

Search and selection criteria for selected papers are based on the research questions, both of which are defined in more detail in the next section 4.2.

The following table 4.1 shows which papers can be associated with which attacks. Note that it is possible that a source introduces multiple attacks and therefore the source appears more than once in the table.

Attack Type	Source Title	Reference
Brute-Force	WLAN Security Protocols and WPA3 Security Approach Measurement through Aircrack-ng Technique	[7]
	Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd	[8]
	From Dragondoom to Dragonstar: Side-channel Attacks and Formally Verified Implementation of WPA3 Dragonfly Handshake	[9]
	A Security Analysis of WPA3-PK: Implementation and Precomputation Attacks	[10]
Spoofing and Evil Twins	WiFi vulnerability caused by SSID forgery in the IEEE 802.11 protocol	[11]
	Preamble Injection and Spoofing Attacks in Wi-Fi Networks	[12]
	Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation	[13]
	Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3	[14]
	Deciphering WEP, WPA, and WPA2 Pre-shared Keys Using Fluxion	[15]
Man-in-the-Middle	Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues.	[16]
	Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems	[17]
	Man-in-the-Middle Attacks without Rogue AP: When WPAs Meet ICMP Redirects	[18]
	Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2	[19]
	Release the Kraken: New KRACKs in the 802.11 Standard	[20]
Decrypting or Forging Packets	Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2	[19]
	Release the Kraken: New KRACKs in the 802.11 Standard	[20]
	Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation	[13]
Side-Channel Attacks in WPA3	Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd	[8]
	From Dragondoom to Dragonstar: Side-channel Attacks and Formally Verified Implementation of WPA3 Dragonfly Handshake	[9]
Denial of Service	Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd	[8]
	Preamble Injection and Spoofing Attacks in Wi-Fi Networks	[12]
	Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation	[13]
	Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3	[14]
	Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems	[17]
	Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues.	[16]

Table 4.1: Sources grouped by attack type

4.2 Source Search Strategy

We defined the following search criteria to limit the sources which we want to include in this review:

- Keywords used for search: Wi-Fi, Security, Vulnerability, Attack, WPA2, WPA3, Evil Twin, Rogue Access Point
- Years 2017 to now (Oct. 2024)
- Publication types included: Journal publications, articles, papers
- Publication libraries: USENIX, IEEE Xplore, ACM Digital Library, Springer

The year of publication was limited to be between 2017 and now (2024). As stated in section 3.1 WPA3 is the latest security standard, while the support for WPA2 is still mandatory, but even in 2024 WPA2 is still the most widely used protocol [21]. We start with the year 2017 because in that year the paper on Key Reinstallation Attacks [19] was published, revealing a serious vulnerability in WPA2 that affected everyone [22].

4.3 Research Question

The primary objective of this work is to review the research on cybersecurity vulnerabilities in Wi-Fi networks. The main research question we aim to answer in chapter 6 is as follows:

Main research question: What are possible threats to Wi-Fi networks?

General Structure Used to Review Vulnerabilities: The (minimum) questions we want to answer for each identified vulnerability are as follows:

- What is the focus of the vulnerability? (Scope and Focus)
- What methods were used, or which experiments were conducted to prove the found vulnerabilities? (Methods and Experiments)
- What conclusion can be drawn from the sources? What questions are left open? (Conclusion)

5 Paper Summary

This chapter provides a summary of each selected source. They are sorted primarily in ascending order by year of publication and secondarily alphabetically by the first surname mentioned on the paper.

5.1 Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2

M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in wpa2," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2017, pp. 1313–1328,* ISBN: *9781450349468.* DOI: *10.1145/3133956.3134027*

5.1.1 Attack

The researcher of this paper [19] found a vulnerability in the 4-way handshake of WPA/WPA2 that they named KRACK (Key Reinstallation Attack).

The attack is possible because of two definitions in the 802.11i standard which are: The Access Point (AP) will retransmit message 1 and 3 of the handshake if it did not receive message 2 or 4 respectively as a confirmation. And the client should install the Pairwise Transient Key (PTK) only after processing and replying to message 3. This allows for message 3 to be resent and therefore the PTK to be reinstalled causing parameters such as the incremental transmit packet number (nonce) and received packet number (replay counter) to be reset to their initial value.

5.1.2 Results

Depending on the implementation of the standard 802.11, the Network Interface Card (NIC), and the Operating System (OS) of a device, they are behaving differently towards retransmitted frames. Some allow the retransmitted message 3 frame to be in plaintext (e.g., wpa_supplicant v2.3-2.6) others allow plaintext retransmission only if it happens right after the first message 3 and before the installation of the PTK (e.g., Android depending on the NIC) and some implementations require for the retransmitted frame to be encrypted (OS X 10.9.5, macOS Sierra 10.12, OpenBSD 6.1 (iwn)). Because the implementation of Windows (7 and 10) and iOS (10.3.1) violate the 802.11 standard they are not vulnerable to KRACK.

They also found that KRACK is applicable on the PeerKey (used when two clients want to communicate with each other directly), group key (used for encryption of broadcast and multicast packets), and Fast BSS Transition (FT) handshake (used when roaming between AP in the same network).

All KRACK attacks beside the one targeting the FT handshake require a Man-in-the-Middle (MitM) position.

5.1.3 Conclusion

The researches suggest that first, the entity implementing the data-confidentiality protocol should not reset associated nonces and replay counter if a key is installed again that is already in use. And second, a particular key should only be installed once during a handshake execution. This

could be solved by a boolean value which only allows the installation of the key if the boolean is set to true otherwise the installation will be skipped.

5.2 Release the Kraken: New KRACKs in the 802.11 Standard

M. Vanhoef and F. Piessens, "Release the kraken: New kracks in the 802.11 standard," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2018, pp. 299–314,* ISBN: *9781450356930.* DOI: *10.1145/3243734.3243807*

5.2.1 Attack

This paper [20] is a continuation of [19] in which the researchers analyze more handshakes and introduce an improved version of the KRACK attack.

They improve their technique on how to execute KRACK if the client no longer accepts plaintext handshake messages after the initial handshake was completed. The new method presented does not depend on hard-to-win race conditions but abuses the power-save functionality in the AP. The authors cause the AP to buffer and later transmit message 3 of the 4-way handshake after the PTK has been installed. The client will then receive this encrypted message 3 and reinstall the key.

They also introduce a more practical way to establish the required MitM position to carry out the attack by using Channel Switch Announcement (CSA) to trick the victim client into switching to a rogue channel controlled by the attacker.

Further the researchers systematically investigated all 802.11 features that also might be affected by KRACK and discover new vulnerabilities in the Fast Initial Link Setup (FILS) handshake and Tunneled Direct-Link Setup (TDLS) PeerKey (TPK) handshake:

FILS: FILS is used to securely connect to an AP and initialize higher-layer protocols simultaneously. The FILS handshake is vulnerable to key reinstallations because the AP may reinstall the PTK when it receives a retransmitted (re)association request. Since the FILS handshake does not have replay counters, an attacker can easily replay a (re)association request to trigger the key reinstallation.

TDLS: With TDLS a direct secure tunnel between two clients can be established. This allows devices to communicate directly without the overhead of passing all traffic through the AP. The TPK handshake is also vulnerable to key reinstallations due to the way implementations handle retransmissions of handshake messages. For example, an attacker can trigger a key reinstallation by forwarding the first two messages of the handshake but blocking the third message. This causes the responder to retransmit the second message, leading the initiator to reinstall the TPK.

The paper then shows how to perform group key reinstallations by manipulating Wireless Network Management (WNM) power-save features. In particular, the authors demonstrate how an attacker can use WNM-Sleep response frames to make the client reinstall the group key.

The authors also bypass the official countermeasure against key reinstallation attacks specified in the updated 802.11 standard. The countermeasure attempts to prevent key reinstallations by specifying that a key's associated parameters should not be reset if it is being reinstalled. The authors bypass this countermeasure by exploiting the interaction between EAPOL-Key and WNM-Sleep frames to temporarily make the victim install a different key before reinstalling the old key.

Finally, the paper presents several implementation-specific key reinstallation vulnerabilities including:

- Implementations that reuse the SNonce or ANonce when refreshing the session key
- APs that accept replayed message 4's of the 4-way handshake
- Devices that always install the group key with an all-zero replay counter

5.2.2 Conclusion

The authors conclude that preventing key reinstallation attacks in 802.11 is more challenging than initially thought and suggest having high-level descriptions or formal models of the standard to help reason about its design and security and make it easier to test the correctness of implementations.

5.3 WiFi vulnerability caused by SSID forgery in the IEEE 802.11 protocol

K. Juhász, V. Póser, M. Kozlovszky, et al., *"Wifi vulnerability caused by ssid forgery in the ieee 802.11 protocol," in* 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), *IEEE, 2019, pp. 333–338.* DOI: *10.1109/SAMI.2019.8782775*

This paper [11] examines a vulnerability in the IEEE 802.11 protocol that allows an attacker to forge SSIDs and create fake Wi-Fi networks. It is possible for attackers to create networks with the same SSID as trusted networks because the protocol identifies Wi-Fi networks solely by their SSIDs. When devices search for nearby Wi-Fi networks, they compare the list of received network SSIDs to their database of known networks and automatically send a connection request to the AP with the matching SSID. This automatic connection feature poses significant risk as devices might connect to the attacker's network without user intervention. In the paper several problems associated with this vulnerability are highlighted by the authors.

5.3.1 Attack

The researchers successfully demonstrated attacks by spoofing or forging the SSID of an existing legitimate Wi-Fi network with their own hardware. For the test setup they focused on broadly available consumer AP for the legitimate network and laptops and mobile phones for clients. A Raspberry Pi took on the role of an evil twin and spoofed the exact SSID of the legit network. The tests were conducted on a password-protected network to further illustrate that not only open, unprotected networks alone pose danger.

When a user device is already connected to the trusted Wi-Fi network, the attacker can force the device to disconnect by forging frames and sending them to the AP on behalf of the victim's device. Forging those frames can be accomplished by techniques discussed further in [14]. Further configuration for the malicious AP may include reducing the frequency of beacon frames being sent out on the air. This increases the chance of clients connecting to the malicious AP because it announces itself more often compared to the legit AP. Once the device is connected to the malicious AP it is possible for the attacker to intercept, alter and inject frames.

5.3.2 Impacts

The impacts of this vulnerability are enormous. Although mitigations are possible, as discussed below, a great number of Wi-Fi networks belonging to ordinary users, such as home Wi-Fi, are not even considered a risk by their owners. This is the case for many other Wi-Fi vulnerabilities that affect most privately owned APs, because to understand the vulnerability a high technical understanding is required. It highlights that vulnerabilities like this one are incredibly common.

5.3.3 Mitigation

To avoid this vulnerability it is necessary to use WPA2-Enterprise authentication methods with the EAP-TLS protocol. With this it is possible to verify clients at an authentication server with

the use of certificates. This setup requires a properly configured RADIUS server as well as issuing and installing certificates on clients.

Considering this, it is highly unlikely that those measures are applicable outside of business or organizational situations. The authors of the paper therefor rightfully point out, that the only viable option for average users is to disable the automatic connection feature for all their devices. Even this seems to be a chore on some mobile devices (depending on vendor implementations in their operating systems), and it is likely that most users will not bother anyway. Although this prevents automatic connections to potentially malicious Wi-Fi networks by the devices without user interaction, it is still possible for users to manually connect to such networks by accident.

Alternatively, detection of fake access points is theoretically possible by deploying dedicated receiver units. Other methods require exact knowledge of the physical locations of trusted APs. The authors point out such solutions and also highlight the additional challenges posed.

5.4 Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd

M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the dragonfly handshake of wpa3 and eap-pwd," in 2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 517–533.* DOI: *10.1109/SP40000.2020.00031*

5.4.1 Attack

The authors of this paper [8] aimed to systematically evaluate the security of the Dragonfly handshake and its implementations within WPA3 and EAP-pwd. Their goal was to identify vulnerabilities and weaknesses in both the protocol design and practical implementations.

The authors conducted tests and audits across a range of devices, vendors, and software, covering both open-source and proprietary implementations.

5.4.2 Implementation-Specific Flaws:

Authentication Bypasses and Reflection Attacks: Their black-box testing revealed authentication bypasses and reflection attacks in multiple implementations of EAP-pwd and WPA3. For instance, they discovered that none of the tested EAP-pwd implementations validated the received scalar or element in the commit frame, enabling an "invalid curve attack" that allowed bypassing authentication. They also found reflection attack vulnerabilities in several server-side EAP-pwd implementations and in certain versions of wpa_supplicant for SAE, allowing an attacker to authenticate as the victim without obtaining the session key.

Insecure Random Number Generation: Code audits and reverse engineering revealed that some implementations, like Aruba's EAP-pwd client for Windows, used predictable sources of randomness, such as the system time, for generating critical cryptographic values. This vulnerability enabled an attacker to predict those values and potentially recover the password.

Insufficient Side-Channel Defenses: The authors found inconsistencies in the implementation of side-channel defenses, particularly related to the number of iterations performed in the hash-to-curve method. Some versions of hostapd and wpa_supplicant used a lower number of iterations than recommended, leaving them vulnerable to timing attacks. EAP-pwd implementations in FreeRADIUS, Radiator, Aruba, and older versions of hostapd and wpa_supplicant didn't perform any extra iterations, increasing their susceptibility to timing attacks.

Information Leaks from Error Handling: The researchers discovered that FreeRADIUS's EAP-pwd implementation leaked information through its error handling. If the hash-to-curve algorithm required more than 10 iterations, the handshake would fail, revealing information that could be exploited in a brute-force attack.

5.4.3 Protocol Design Weaknesses:

Downgrade and Dictionary Attacks against WPA3 Transition Mode: While WPA3's transition mode aims to provide backward compatibility with WPA2 devices, the authors found that an attacker could exploit this mode to perform downgrade and dictionary attacks. By forcing a client to connect using WPA2, an attacker could capture enough information to conduct a dictionary attack against the password, even if the network ultimately supported WPA3.

Group Downgrade Attack against Simultaneous Authentication of Equals (SAE): They discovered a vulnerability in SAE's group negotiation mechanism that allowed an attacker to force a client into using a weaker cryptographic group during the handshake. This attack exploited the lack of cryptographic validation in the group negotiation process, enabling an attacker to intercept and manipulate messages to downgrade the security level of the connection.

Denial-of-Service Vulnerability Due to High Overhead: The authors found that the defenses against known timing side-channels in Dragonfly introduced significant computational overhead. This overhead could be abused in a Denial-of-service attack to overload the CPU of an access point, preventing legitimate clients from connecting. They demonstrated the practicality of this attack, showing that a low-powered device could effectively cripple a modern access point by spoofing commit frames and exploiting the handshake's computational complexity.

5.4.4 Novel Side-Channel Attacks:

Timing Attacks against Hash-to-Group and Hash-to-Curve: Further they identified novel timing vulnerabilities in both the hash-to-group (used with MODP groups) and the hash-to-curve (used with elliptic curve groups) methods. They showed how variations in execution time, caused by factors like the number of iterations and the processing of Key Derivation Function (KDF) outputs, could leak information about the password. These timing differences, even when subtle, could be measured and analyzed by an attacker to gain insights into the password's structure and eventually recover it through brute-force techniques.

Cache-Based Attacks on ECC Groups: The authors discovered that implementations of the hash-to-curve algorithm using elliptic curve groups were susceptible to cache-based side-channel attacks. They demonstrated how an attacker could monitor cache access patterns to infer the result of the quadratic residue test in the hash-to-curve method. This leaked information, combined with other side-channel data, could significantly reduce the complexity of brute-forcing the password.

5.4.5 Conclusion

To mitigate the attacks the researchers propose to remove the peer's MAC addresses (identities) from the hash-to-group and hash-to-curve algorithms, so that the password element can be precomputed offline. This would reduce the impact of side-channel leaks.

They conclude that because of their findings WPA3 and EAP-pwd do not meet the standards of modern security protocols. They further comment that an open design process could have avoided the weaknesses they found.

5.5 Deciphering WEP, WPA, and WPA2 Pre-shared Keys Using Fluxion

S. Atluri and R. Rallabandi, "Deciphering wep, wpa, and wpa2 pre-shared keys using fluxion," in Smart Computing Techniques and Applications*, Springer Singapore, Jul. 2021, pp. 377–385,* ISBN: *978-981-16-0878-0.* DOI: *10.1007/978-981-16-0878-0_37*

5.5.1 Attack

In this paper [15] the authors show how to use a tool called Fluxion to attack WEP, WPA and WPA2 networks.

Fluxion sets up a fake login page that mimics the router's login page. It then disconnects the user from the legitimate network and lures them to connect to the fake access point, which has the same name as the legitimate one. When the user tries to connect to the internet, they are presented with the fake login page that prompts them to enter their Wi-Fi password. In this process, Fluxion automates several tasks involved in the attack, such as capturing the handshake, performing de-authentication attacks, setting up a fake access point, and jamming the legitimate network. It also provides a user-friendly interface that guides the user through the attack process.

In comparison to other tools like Aircrack-ng that try to crack the Wi-Fi passwords with brute-force attacks, Fluxion leverages social engineering to obtain the password.

5.5.2 Conclusion

Fluxion is effective for WEP and WPA/WPA2-Personal, it is not designed to work directly with WPA/WPA2-Enterprise networks due to the more secure and complex authentication mechanisms used in enterprise environments. The attacker would need to intercept the authentication process and crack the captured handshake, which is significantly more challenging because of the additional encryption and authentication layers present in enterprise setups.

Fluxion prevents an attacker from having to learn and memorize commands like it would be necessary when using Aircrack-ng. But because Fluxion relies on social engineering it success depends on the users awareness of the fake login page.

5.6 WLAN Security Protocols and WPA3 Security Approach Measurement through Aircrack-ng Technique

E. Baray and N. Kumar Ojha, "Wlan security protocols and wpa3 security approach measurement through aircrack-ng technique," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), *2021, pp. 23–30.* DOI: *10.1109/ICCMC51019.2021.9418230*

5.6.1 Attack

The researcher in this paper [7] tests the security of WLAN networks and executes a downgrade attack on WPA3, followed by password cracking using Aircrack-ng.

5.6.2 Attack Execution

When a device attempts to connect to a Wi-Fi network, both the client and the access point exchange their supported cipher suites within the Robust Security Network Element (RSNE). This element is included in beacon frames but WPA2 also verifies the authenticity of the RSNEs exchanged during the four-way handshake. If the handshake process reveals a mismatch between the expected and received RSNEs, indicating potential manipulation by an attacker, the handshake is immediately terminated.

The basic idea to create a fake access point and send beacon frames to the client in which only WPA2 support gets advertised. Because the four-way handshake gets initialize by the access point and the first message is unauthenticated the verification can be evaded. After receiving the second message of the four-way handshake from the client the attackers can start a dictionary attack using Aircrack-ng to crack the password which is the same in WPA2 and WPA3.

In [7, Section 4] they explain in eleven stages how to execute the attack.

5.6.3 Conclusion

The research concludes that while WPA3 offers enhanced security, it remains vulnerable to attacks, particularly when downgraded, emphasizing the ongoing need for improved Wi-Fi security measures.

5.7 Tracking a Rogue Mobile Access Point

L. Qawasmeh and F. Awad, "Tracking a mobile rouge access point," in 2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 522–526.* DOI: *10.1109/ICIT52682.2021.9491684*

This research paper [23] explores the problem of tracking Mobile Rogue Access Points (MRAPs). A classic Rogue Access Point will be stationary and according to the researchers of this paper, that problem has already been successfully addressed. MRAPs on the other hand, being portable wireless access points disguised as legitimate ones, are much more feasible than before and can be found in various mobile devices.

The authors propose a novel approach to localize and track these MRAPs using received data of Received Signal Strength Indicator from beacon frames sent out by the MRAPs. By employing trilateration and analyzing RSSI data from reference nodes, the algorithm estimates the MRAP's location and predicts its future movement. Trilateration is a method for determining a location by measuring distances from at least three known reference points and finding the intersection of the resulting circles (2D) or spheres (3D).

5.7.1 Objective

The related attack to this research are RAPs. Instead of demonstrating the attack, the researchers aims to facilitate the real-time tracking and eventual arrest of individuals using these RAPs, contributing to improved wireless network security. Therefor the localization of a RAP in real-time and with sufficiently high accuracy is an effective countermeasure.

5.7.2 Methods

The approach is to track MRAP using the RSSI associated with the beacon frames that RAPs send out to announce their presence. The RSSI measurements from three reference wireless devices are used to estimate the distances to the RAP. The distance set is then used as input to a trilateration algorithm to estimate the RAP's location. The paper also uses a number of consecutive estimated locations to extract the speed and direction of mobility of the mobile RAP. This information is then used to predict the next location of the RAP.

5.7.3 Practical implementation

The proposed algorithm to perform the tracking is relying on beacon frames of legitimate access points with their position known to estimate the signal characteristics of any given environment. Then the algorithm uses signals from three stationary WLAN stations at known locations as reference nodes. The signals of the reference nodes are centrally collected on the localization server. The Lognormal Shadowing Path Loss model is used to estimate the distance between the MRAP and each reference node. To reduce the impact of noise and fluctuations on the distance estimation accuracy, the average of consecutive RSSI values at each reference node is used. Linear regression is performed on the last consecutive estimated locations to estimate the MRAP's current direction and speed of movement.

5.7.4 Conclusion

The performance of the algorithm was evaluated through computer simulations using metrics such as Mean Localization Error (MLE) and Mean Next Location Error (MNLE). MLE refers to the Euclidean distance between the estimated and actual locations of the MRAP. MNLE is the Euclidean distance between the predicted next location and the corresponding actual location of the MRAP. The performance evaluation results indicate that the proposed algorithm can localize and track mobile RAPs with sufficient accuracy. The average localization error achieved ranged from 1.2m to 2.2m, while the average next location error was between 2m to 3.5m.

As earlier noted, this paper is not on an attack or vulnerability, instead it showcases a solution to the problem of RAPs. The proposed method allows for accurate tracking of moving and stationary RAPs in a given environment, according to their evaluation. This can be especially important in situations where a given Wi-Fi network range should be closely monitored and there is the possibility of localising a RAP if it is detected.

5.8 Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation

M. Vanhoef, "Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation," in 30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 161–178,* ISBN*: 978-1-939133-24-3. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/vanhoef*

This research [13] shows the discovery of three design and implementation flaws in the underlying standard for Wi-Fi, the 802.11 standard. One flaw is related to frame aggregation while the two other flaws are related to frame fragmentation.

5.8.1 Attacks

These flaws can be exploited by an adversary to exfiltrate sensitive data by forging encrypted frames. The author highlights the impact of these vulnerabilities on all protected Wi-Fi networks, spanning from older WEP to modern WPA3 Wi-Fi networks.

Firstly, a flaw in the 802.11 frame aggregation enables attackers to inject arbitrary frames. By Manipulating a normal 802.11 frame to be processed as an Aggregate MAC Service Data Unit (A-MSDU) frame, an attacker can make one of the subframes correspond to the packet they want to inject. This vulnerability allows port scans to be performed and tricks a victim into using a malicious DNS server.

Secondly, two flaws related to frame fragmentation revolve around the encryption of and the handling of decrypted frames in memory. It is possible for an attacker to forge frame fragments with a different encryption key. Although all fragments of a frame usually are encrypted with the same key, by the 802.11 standard it is not required to be checked nor enforced whether this is the case [13, Sec. 4.1].

5.8.2 Impact

The 802.11 standard does not state when received and decrypted fragments should be removed from the receiver's memory. This allows an adversary to inject fragments into a victim's fragment cache. Then the attacker can combine injected fragments with legitimate ones and proceed to inject packets or exfiltrate decrypted fragments. Attackers can use these exploits to exfiltrate data [13, Sec. 5.2] or inject packets [13, Sec. 5.3]. By injecting packets that force the client to use a malicious DNS server, the attacker can intercept virtually all client traffic.

5.9 Preamble Injection and Spoofing Attacks in Wi-Fi Networks

Z. Zhang and M. Krunz, "Preamble injection and spoofing attacks in wi-fi networks," in 2021 IEEE Global Communications Conference (GLOBECOM), *2021, pp. 1–6.* DOI: *10.1109/GLOBECOM46510.2021.9685461*

In this paper [12] the authors focus on vulnerabilities of the preamble, which is a crucial component in Wi-Fi frames. It is used to support frame detection, synchronization and channel estimation. The researchers emphasize that the preamble is particularly vulnerable to Preamble Injection and Spoofing (PrInS) attacks.

5.9.1 Attack

By injecting forged preambles, these attacks are disrupting and potentially causing substantial problems. The study examines different PrInS attack scenarios, including frame detection attacks, channel silencing, and data alteration attacks. Software-defined Radios (SDRs) are used to demonstrate the efficacy of these attacks. Experimental results of these demonstrations are presented.

5.9.2 Mitigation

To mitigate risks of the PrInS attacks, the paper proposes a backward-compatible scheme for preamble authentication as a countermeasure.

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

N. Hoque, H. Rahbari, and C. Rezendes, "Systematically analyzing vulnerabilities in the connection establishment phase of wi-fi systems," in 2022 IEEE Conference on Communications and Network Security (CNS), *2022, pp. 64–72.* DOI: *10.1109/CNS56114.2022.9947252*

5.10.1 Attack

The researchers in this paper [17] formally modeled and analyzed the connection establishment phase of Wi-Fi systems using the symbolic Model Checker (MC) NuSMV, with the goal to discover vulnerabilities. They provided the code of the MC [24]. They found three new variants of known MitM attacks and a new DoS vulnerability.

5.10.2 Findings

The main property they checked was that a station and an AP would always eventually connect over the same channel. They analyzed different scenarios how an adversary could violate this property and found the following:

- MitM: An adversary could send fake CSA to the station to deceive it to switch its channel.
- MitM: An adversary could send fake CSA to the AP to deceive it to switch its channel.
- MitM: An adversary could send fake CSA to the station and the AP to deceive them both to switch to two different channels.
- OCV: Could prevent CSA-based MitM attacks but was vulnerable to other types of MitM attacks. E.g., relay attacks where an attacker sets up a rogue AP on the same channel as a legitimate AP, but in a location where the AP and the station can't directly communicate.

- DoS: By selectively jamming one of the pre-authentication frames they can cause the retransmission limit to be repeatedly exhausted, then the station would be going back to the disconnected/probing state.

5.10.3 Experiment

They used an iPhone 6s running version 13.3.1 of iOS with Aruba AP and used Wireshark to record the frame arrival times to test the performance of the Operating Channel Validation (OCV) technique. To test the DoS attack, they used a virtual machine running Ubuntu 20.04.3 LTS 64-bit with the Wi-Fi framework that is integrated with hostapd and wpa_supplicant. They note that they were unable to test the battery drain effect of repeatedly forcing a connection attempt because of the limitations of the virtual testing environment.

5.10.4 Conclusion

To mitigate this DoS vulnerability, the researchers suggest randomizing the pre-authentication frame's timeout window and retry count values. The researchers note that this technique makes the attack less stealthy, as continuous jamming is more likely to be detected.

They also mention that the 802.11 standard is quite ambiguous in its language or fails to specify certain parameters completely, which can lead to different interpretations and potentially insecure implementations. For example, the standard doesn't define the retransmission limit for frames, leading to different values being used in different implementations. The authors found this limit to be 7 in the ns3 simulator documentation, but only 3 in the hostapd and wpa_supplicant implementations. The standard also leaves some MAC header fields as "Vendor Specific" or allows for "implementation decisions" without providing clear guidelines or warnings about potential security risks. This lack of specificity can lead to vulnerabilities if developers aren't careful about their implementation choices.

5.11 Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3

K. Lounis, S. H. Ding, and M. Zulkernine, "Cut it: Deauthentication attacks on protected management frames in wpa2 and wpa3," in Foundations and Practice of Security, *Springer International Publishing, 2022, pp. 235–252,* ISBN: *978-3-031-08147-7.* DOI: *10.1007/978-3-031-08147-7_16*

This paper [14] examines how deauthentication attacks can be carried out on Wi-Fi networks, specifically those that employ WPA2-PSK or WPA3-PSK with Protected Management Frames (PMF) enabled. The authors highlight that the vulnerability works despite the security enhancements introduced by the IEEE 802.11w amendment, which mandates the use of PMF. Through empirical testing, the authors reveal that even PMF-enabled networks remain vulnerable to these attacks, pointing to specific weaknesses in some vendor-specific implementations of the standard. The findings underscore that the protections intended by PMF are insufficient against certain attack vectors, exposing devices and networks to potential exploitation. The authors also examine possible countermeasures to mitigate these vulnerabilities and emphasize the need for device manufacturers to rigorously validate their implementations, ensuring that they adhere to the highest security standards to protect against such deauthentication threats.

5.11.1 Attacks

The researchers successfully demonstrated attacks in three scenarios. They used the Aircrack-ng suite to generate deauthentication frames, capture Wi-Fi traffic, and fake authentication sessions.

Scapy was used to create custom programs allowing them to adjust specific parameters and test different attack scenarios.

Firstly, flooding the client and AP with large amounts of spoofed, unprotected, unicast deauthentication and disassociation frames could cause the AP to disassociate the client. By sending frames in both directions the Security Association query (SA-query) procedure on both the AP and client was initiated. Surprisingly, they also found that sending these frames only to the AP on behalf of the client could also cause disconnection, but sending them to the client on behalf of the access point did not.

Secondly, an implementation flaw in certain Apple devices was found by the researchers. When a fake authentication session using the IEEE 802.11 open system mode is initiated, the AP rejected the association with reason "Association Request Rejected temporarily; Try Again Later (Code 30)", which has made the AP exchange protected action frames to check the legitimacy of the new association. The researchers discovered however, that the client (specifically for Apple devices) did not react as described by the standard, which requires it to wait for the SA-query request and then to properly respond to it. If the standard is followed, the client would respond to the SA-query in a way that conveys to the AP that the request was not made by the respective client. All this results in disconnection of the client by disassociation by the access point. However, this vulnerability is fixed by Apple with macOS Monterey v12.0 Beta and iOS 15 Beta.

Lastly, the researchers made the hypothesis, that overwhelming devices with a flood of spoofed frames might impact their ability to process and respond to SA-query requests and responses in time. Further they suspect this to stem from vulnerabilities caused by how devices implement the SA-query procedure, which in turn means various devices of different vendors can be affected by this vulnerability. They propose that device manufacturers should focus on strengthening the SA-query procedure to enhance its resilience against attacks that exploit it. A robust SA-query procedure should effectively verify the authenticity of management frames and prevent spoofing attacks even under stress from high volumes of traffic or malicious activity.

5.11.2 Impacts

The SA-query procedure is designed to verify authenticity of management frames exchanged by the client and AP. Clients taking too much time or getting overwhelmed by too many of those procedures are ultimately disconnected from the AP. This effectively results in Denial-of-service. Furthermore, disconnected devices face severe difficulties when trying to rejoin the network. Especially if the attack is still ongoing. The continuous flood of spoofed frames hinders the clients ability to process legitimate ones and successful reconnection is very challenging if not impossible.

5.12 From Dragondoom to Dragonstar: Side-channel Attacks and Formally Verified Implementation of WPA3 Dragonfly Handshake

D. De Almeida Braga, N. Kulatova, M. Sabt, et al., *"From dragondoom to dragonstar: Side-channel attacks and formally verified implementation of wpa3 dragonfly handshake," in* 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), *2023, pp. 707–723.* DOI: *10.1109/EuroSP57164.2023.00048*

5.12.1 Attack

The paper [9] examines the security of the Dragonfly handshake used in WPA3, the latest Wi-Fi security protocol. The authors uncover a collection of vulnerabilities, collectively termed "Dragondoom," that arise from the interaction of the Dragonfly protocol with external cryptographic

libraries. These vulnerabilities can lead to side-channel attacks, enabling an attacker to recover a user's Wi-Fi password.

5.12.2 Leakage Vectors

The paper focuses on two specific leakage vectors:

Point Decompression: This vulnerability affects the "hunting-and-pecking" password conversion method used in the SAE protocol, a variant of Dragonfly used in WPA3. The leakage occurs during the decompression of an elliptic curve point, where the compression format is dependent on the user's password. Attackers can exploit timing variations in the point decompression algorithm to deduce information about the password.

Binary to Big Number Conversion: This vulnerability impacts both "hunting-and-pecking" (SAE) and the newer SSWU (Simplified Shallue-Woestijne-Ulas) method (SAE-PT). The issue stems from an optimization in the conversion routine that skips leading zero bytes in a secret value. By observing timing variations, attackers can infer the number of leading zeros, leaking information about the secret value, which is derived from the password.

The researchers analyze the implementations of these routines in popular cryptographic libraries like OpenSSL, WolfSSL, and ell, which are commonly used in Wi-Fi daemons such as hostapd, FreeRadius, and iwd. They find that all of these libraries are vulnerable to at least one of the identified leakage vectors.

5.12.3 Practical Testing

To demonstrate the practical implications of these vulnerabilities, the authors conduct a real-world attack against wpa_supplicant (a Wi-Fi client) using OpenSSL, a common configuration in Linux systems. They use a technique called Flush+Reload, a type of cache-based side-channel attack, to monitor the execution of specific instructions and extract information about the password. Notably, they introduce a novel Flush+Reload gadget to overcome the spatial limitations of traditional Flush+Reload attacks, achieving more precise and reliable measurements. The paper compares the efficiency of their attack with previous attacks against WPA3's Dragonfly handshake. Their results show that "Dragondoom" is significantly more efficient, requiring fewer measurements to extract the same amount of information. This improvement is attributed to the precise measurements achieved through their Flush+Reload gadget and the exploitation of a previously overlooked leakage vector.

5.12.4 Mitigation

To mitigate these vulnerabilities, the researchers propose "Dragonstar" a formally verified implementation of the Dragonfly handshake that utilizes the HACL* cryptographic library. HACL* offers mathematically proven guarantees of secret independence for all cryptographic operations, effectively eliminating the leakage vectors exploited by "Dragondoom". They demonstrate that Dragonstar can be integrated into hostapd as a drop-in replacement for OpenSSL with comparable performance.

5.12.5 Conclusion

The paper concludes by highlighting the importance of considering the security implications of interactions between cryptographic protocols and external libraries, even when those libraries are widely used and trusted. Formally verified implementations like Dragonstar, which provide strong security guarantees at the level of individual cryptographic operations, are presented as a robust solution for mitigating the risks of side-channel attacks in complex systems like WPA3.

5.13 Man-in-the-Middle Attacks without Rogue AP: When WPAs Meet ICMP Redirects

X. Feng, Q. Li, K. Sun, et al., *"Man-in-the-middle attacks without rogue ap: When wpas meet icmp redirects," in* 2023 IEEE Symposium on Security and Privacy (SP), *IEEE, 2023, pp. 3162–3177.* DOI: *10.1109/SP46215.2023.10179441*

This research paper [18] describes a new type of Man-in-the-Middle (MitM) attack that can hijack traffic on Wi-Fi networks without requiring a rogue access point. The attack exploits a vulnerability in the interaction between the ICMP and WPA protocols, allowing attackers to spoof the legitimate AP and redirect a victim's traffic. The authors demonstrate their attack's effectiveness through an extensive evaluation on 122 real-world Wi-Fi networks, finding that 89% are vulnerable. They propose two countermeasures: Enhancing supplicant security checks and modifying AP routers to filter spoofed ICMP messages.

5.13.1 Attack

For a successful attack some prerequisites are necessary according to the paper:

- ICMP redirects must be enabled in the Wi-Fi network. If the AP issues an ICMP redirect message, the supplicant optimizes its routing. Many supplicants and networks support this mechanism.
- Supplicants (e.g., victim and attacker) must be able to communicate with each other to allow the attacker to receive the victim's traffic.
- The attacker needs the IP addresses of the victim and the server the victim is communicating with. In IPv4, network probing easily reveals this, in IPv6, probing within the network is required, and existing methods improve efficiency. Popular targets include DNS, search engines, and social media.
- The attacker must identify open UDP ports on the victim. Many systems have open UDP ports by default, probing for these within the network is hard to block with firewalls or middle-boxes.
- The attacker must send spoofed packets using the AP's source IP address. None of the 55 tested APs could block spoofed packets sent to the victim.

The attack works by poisoning the victim's routing table, redirecting their traffic to the attacker, who can then decrypt and observe the communication. This is a summary of the three-step process required:

1. **Probing the victim:** The attacker needs to gain access to the Wi-Fi network, since they need to be connected to the same network as their victim. This is relatively easy for networks in personal mode by obtaining the pre-shared key, enterprise mode might require stealing user credentials or exploiting further vulnerabilities in the authentication process. Next, the IP of the victim and a server the victim is communicating to need to be identified. This can be achieved through network scanning techniques and a commonly used service, such as DNS, to identify a server. Lastly, an active UDP port needs to be discovered on the victim, which can be helped with consulting commonly used ports by operating systems. Port probing is difficult to block in Wi-Fi networks due to the absence of firewalls or middle-boxes.

2. **Exploiting ICMP to poison the victim's routing:** An attacker crafts a forged ICMP redirect message, spoofing the AP's IP to instruct the victim's device to route traffic through the attacker's IP. To bypass modern OS legitimacy checks, the attacker embeds a fake UDP header with an active UDP port identified earlier, making the message seem genuine. A

vulnerability in many AP routers, where the Network Processing Unit (NPU) prioritizes performance over security, allows these forged messages to pass unchecked to the victim without allow higher-layer security mechanisms to block it.

3. **Hijacking traffic through cross-layer deception:** Once the victim's routing table is poisoned, their traffic is redirected through the attacker. While the victim sends traffic to the AP, unaware of the manipulation, the AP decrypts it using the victim's session key, re-encrypts it with the attacker's session key, and forwards it. The attacker then decrypts the traffic, bypassing WPA security and accessing the victim's plaintext communication.

See [18, Fig. 2] for a visual overview of the three attack steps.

The attacker leverages the fact that WPA encryption operates at the link layer, while routing decisions happen at the IP layer. By manipulating the routing using ICMP, they can force the AP to unknowingly decrypt and re-encrypt traffic, handing it over to the attacker. Proposed countermeasures include stricter checks on ICMP messages by devices and enhanced AP filtering for forged traffic.

5.13.2 Impacts

The paper highlights a significant security vulnerability in Wi-Fi networks, demonstrating the real-world feasibility of Man-in-the-Middle attacks using ICMP redirects. Through experiments with common hardware in public Wi-Fi settings, researchers revealed that 89% of the 122 networks tested were vulnerable. This vulnerability allows attackers to intercept and decrypt victim communications, exposing sensitive data like passwords, private messages, and more. Beyond eavesdropping, attackers can manipulate data, injecting malicious code or altering transactions in consecutive attacks downstream. The attack compromises privacy, reinforces caution towards public Wi-Fi providers, and risks reputational harm to organizations offering such networks.

Its stealthy nature, requiring no rogue access point and exploiting cross-layer vulnerabilities, makes it hard to detect using traditional methods. Widely used Wi-Fi protocols like WPA2 and WPA3 are affected, putting many devices and networks at risk. Those most impacted include individuals using public Wi-Fi, organizations offering these networks, AP router manufacturers, and operating system developers. The researchers emphasize the need for action, including stricter ICMP checks, security patches, and enhanced router and device protections, to mitigate this pervasive threat.

5.13.3 Mitigation

They propose two countermeasures:

The first proposed countermeasure to mitigate the vulnerability involves enhancing the security of individual supplicants by validating cross-layer interactions. This approach focuses on detecting inconsistencies between the IP and link layer information in received ICMP redirect messages. Legitimate messages from an AP maintain consistent source IP and MAC addresses, while forged messages from attackers spoof the AP's IP address but have a mismatched source MAC address. To address this, the researchers recommend modifying the operating system kernel to check for such discrepancies, with a proof-of-concept implementation provided for the Linux 4.18 kernel. This solution does not depend on changes from AP manufacturers or network operators, making it a flexible option for users to protect their devices. However, it requires system-level updates, which may not be available for all devices, and depends on users' willingness to update or patch their systems.

The second proposed countermeasure targets the network infrastructure by enhancing APs to filter and block forged ICMP redirect messages. APs can identify forgeries by recognizing discrepancies where the source IP in the message matches the AP's IP but originates from a

different MAC address. This requires APs to implement stricter filtering rules to scrutinize and discard such packets, even those forwarded internally. By doing so, this measure protects all connected devices, regardless of their operating systems or individual configurations. However, it relies on AP manufacturers to update firmware and fix vulnerabilities in their NPUs, a process that can be slow and met with resistance from some manufacturers.

Together, these two countermeasures offer complementary protection, addressing the issue at both the device and network levels for a more robust defense against the attack.

5.14 Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues

D. Schepers, A. Ranganathan, and M. Vanhoef, "Framing frames: Bypassing wi-fi encryption by manipulating transmit queues," in 32nd USENIX Security Symposium (USENIX Security 23)*, USENIX Association, Aug. 2023, pp. 53–68,* ISBN*: 978-1-939133-37-3. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/schepers*

The research in this paper [16] reveals vulnerabilities caused by the mismanagement of security contexts within Wi-Fi transmit queues across various devices and operating systems. The researchers outline a general attack strategy in their paper that forces an AP to queue frames and then manipulates the security context to leak the queued frames.

5.14.1 Attacks

An attacker tricks the AP into believing the victim client is entering power-save mode, causing the AP to queue frames in plaintext. Then, the attacker manipulates the AP's security context, removing the victim client's encryption keys. Finally, the attacker triggers the AP to transmit the queued frames, which are now either sent in plaintext or encrypted using a weaker key. This weaker key can be an all-zero key or the group key.

This attack was observed to be working with different open-source network stacks like FreeBSD and Linux, and also in hardware-dongles with open-source firmware. The specific vulnerabilities and patterns of how the information will be leaked differs with distinct implementations. Some FreeBSD drivers for example leak frames encrypted with the group key, while others leak frames in plaintext or with WEP encryption using an all-zero key. The observed behavior reflects a situation where the transmit queues of APs are not adequately dequeued or purged when the security context changes.

In Linux systems two attack variations utilizing race conditions were discovered [16, Sec. 3.5.2] when the encryption operations are offloaded to the hardware. In the transmission mechanism frames are leaked in plaintext when retransmission is attempted a number of times. An attacker might jam a victims' connection in order to force the AP to retransmit frames and in combination with sending an authentication request without the sleep-bit set to remove the pairwise key and wake up the client. This results in the last retransmission happening in plaintext.

5.14.2 Impacts

The learnings coming out of the authors' work present vulnerabilities with significant practical impact [16, Sec. 6.1]. The security context override attack allows an adversary to compromise frames that are part of a TCP connection. The attacker can acknowledge these incoming packets and receive all pending TCP packets from the server. It also allows to hijack the TCP session because of the revealed sequence numbers and an adversary can inject off-path TCP packets with a spoofed sender address. This in turn could be used to send malicious JavaScript to exploit

vulnerabilities in the victims' browser. Intercepting client traffic by identifying the port and transaction identifier that a client is using for DNS requests and spoofing DNS responses allows the attacker to redirect and intercept most traffic sent by the victim. By intercepting any packet sent to the client and attacker can steal cookies from plaintext HTTP websites, and learn the IP addresses of servers the victim is connecting to, even if TLS is used. Based on this information an attacker can determine the websites and services the victim is visiting, allowing further insights in more possible attack-vectors.

5.14.3 Countermeasures

As the standard does not define how an access point should behave if the security context changes, the researches propose the following:

- Before deleting a pairwise encryption key, the transmit queue should be dequeued, regardless of the receiving client's power-save status. This would involve the access point making a final attempt to send any frames in the transmit queue.
- Before deleting a pairwise encryption key, the transmit queue should be purged. This would involve the access point dropping all frames stored in the transmit queue.

Further they propose to implement defenses against attacks that target hotspot-like networks by temporarily preventing clients from connecting if they are using a MAC address that was recently connected. This prevents an attacker from spoofing a MAC address to intercept pending or queued frames. To securely recognize recently-connected users, an AP can store a mapping between the client's MAC address and their cached security associations. This allows a client to immediately reconnect using a recently used MAC address by proving they possess the cached security association. Another method to recognize recently connected users is based on the EAP identity they used during 802.1X authentication. The AP securely learns the EAP identity from the RADIUS server that authenticated the client and keeps a mapping of recently connected MAC addresses with their corresponding EAP identity. If the same MAC address attempts to connect under a different EAP identity, the client is forced to wait before connecting.

5.14.4 Conclusion

The authors of the paper conclude that the root cause of the security issues they investigated is that the 802.11 standard is vague and does not provide explicit guidance on how to manage security context changes in situations involving transmit queues. As a result, they discovered that modern devices and operating systems fail to securely manage security context in these scenarios, creating opportunities for attackers to intercept frames or perform denial-of-service attacks.

They also emphasize the importance of considering queuing mechanisms in relation to a changing security context when developing and implementing the 802.11 standard. They note that formal models of WPA2 should take into account transmission queues and security context updates to improve the security of future implementations.

The paper specifically highlights that while higher-layer security mechanisms like HTTPS and TLS can help mitigate some risks associated with leaked Wi-Fi frames, these measures alone are insufficient. Attackers can still exploit vulnerabilities to obtain sensitive information, such as the IP addresses a client communicates with. This information can potentially be used to identify the websites visited by the victim, underscoring the need for robust security mechanisms at all layers of the network stack.

5.15 A Security Analysis of WPA3-PK: Implementation and Precomputation Attacks

M. Vanhoef and J. Robben, "A security analysis of wpa3-pk: Implementation and precomputation attacks," in

Applied Cryptography and Network Security, *Springer Nature Switzerland, 2024, pp. 217–240,* ISBN: *978-3-031-54773-7.* DOI: *10.1007/978-3-031-54773-7_9*

5.15.1 Attack

This paper [10] is a comprehensive analysis of SAE-PK protocol (referred to as WPA3-PK in this paper). The authors investigate a range of attack vectors, including implementation related weaknesses and vulnerabilities at the network layer. They also conduct an in-depth examination of time-memory trade-off attacks and introduce the application of rainbow tables to enhance the effectiveness of these attacks.

5.15.2 Implementation and Network-Based Attacks

Bad Randomness: The authors investigated the security implications of flawed random number generator implementations in WPA3-PK. They analyzed three different open-source implementations of the PKHash algorithm (used for generating WPA3-PK passwords): Hostap's sae-pk-gen, an OpenSSL-based tool and a Python3 implementation. They found that the OpenSSL-based implementation, used in a dd-wrt fork, initialized the modifier value using the router's MAC address, which is predictable. The Python3 implementation simply initialized the modifier to zero. This predictability makes it possible for an attacker to deduce the WPA3-PK password if they know the hotspot's public key.

Client-to-Client Attacks: Even though WPA3-PK prevents the setup of rogue APs, the authors tested the feasibility of standard network-layer attacks. Using ARP poisoning, they successfully intercepted traffic between clients connected to a WPA3-PK network.

Group Key Abuse: They also tested the exploitation of shared group keys, which are used for broadcast and multicast traffic in WPA3-PK networks. They successfully injected both broadcast and unicast traffic towards a victim client, even when client-to-client communication was disabled, by obtaining the group key.

5.15.3 Time-Memory Trade-off Attacks

Baseline Time-Memory Trade-off Attack: The authors evaluated a pre-existing time-memory trade-off attack against WPA3-PK. This baseline attack involves precomputing tables of distinguished points (fingerprints with a specific number of leading zeros). They conducted simulations to assess the performance of the attack under different parameters, including the number of tables and the number of starting points per table. Their findings confirmed that this type of attack could effectively recover a WPA3-PK password in a practical timeframe, especially when using the weakest allowed security settings.

Rainbow Table Attack: To enhance the efficiency and success rate of the time-memory trade-off attack, they introduced the concept of rainbow tables, which use multiple reduction functions during the table generation process to reduce chain collisions. The authors implemented a proof-of-concept tool and ran simulations to evaluate the performance of this enhanced attack. Their results showed that using rainbow tables significantly improved the probability of finding a matching WPA3-PK password.

5.15.4 Password Collision Attacks

Multi-Network Password Collisions: The authors devised a technique for creating password collisions across multiple SSIDs. This involves exploiting flexibility in the parsing of public keys, allowing them to craft a public key that can be interpreted in different ways depending on the SSID length. This vulnerability enables the construction of a single precomputed table capable

of attacking numerous networks using the same WPA3-PK password, even with different SSIDs. They validated the practicality of this attack by creating a modified AP that advertised a crafted public key and testing it against a vulnerable client.

5.15.5 Conclusion

The authors conclude that while WPA3-PK offers significant security improvements over its predecessors, it is crucial to consider implementation details and potential network-layer vulnerabilities. They highlight that relying solely on the protocol's theoretical strength is insufficient to guarantee robust security.

Further they recommend not to use the weakest allowed WPA3-PK password as time-memory trade-off attacks are on the verge of practicality.

6 Literature Review

The literature-review is structured in sections by threats:

1. Brute-Force
2. Spoofing (and Evil Twin)
3. Man-in-the-Middle (MitM)
4. Decrypting or Forging Packets
5. Side-Channel Attacks in WPA3
6. Denial-of-service (DoS)

A source may be mentioned in more than one section due to its scope and relevance in different fields.

6.1 Brute-Force

Papers relevant for this section:

- [8] Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd
- [9] From Dragondoom to Dragonstar: Side-channel Attacks and Formally Verified Implementation of WPA3 Dragonfly Handshake
- [7] WLAN Security Protocols and WPA3 Security Approach Measurement through Aircrack-ng Technique
- [10] A Security Analysis of WPA3-PK: Implementation and Precomputation Attacks

6.1.1 Scope and Focus

In WPA2 the Pairwise Master Key (PMK) was directly derived from the Pre-shared Key (PSK) which allowed an attacker who sniffed the four-way handshake, to crack the password in an offline brute-force attack as described in [7]. WPA3 addresses this problem by requiring the client to authenticate using the SAE protocol.

But brute-force attacks are also possible in WPA3. As described in section 6.5, the sources [8] and [9] introduce side-channel attacks. By observing subtle variations in execution time or memory access patterns during the SAE handshake, attackers can gather information about the password and narrow down the search space.

Source [10] focuses on the SAE-PK and explores the use of precomputation attacks and rainbow tables to improve time-memory trade-off attacks. The idea is to precompute a massive table of possible password hashes, enabling faster attacks by reducing the computational effort required during the attack phase. Additionally, they introduce multi-network password collisions, a technique that aims to create multiple networks with different SSIDs but the same SAE-PK password. This allows attackers to use a single precomputed table to attack multiple networks, making attacks more efficient.

In source [7] the authors explain how to use Aircrack-ng to execute a downgrade attack and brute-force the password. The feasibility of this attack is also mentioned in [8].

6.1.2 Methods and Experiments

All sources analyze a topic regarding WPA3 and conduct experiments to proof their claims.

The researchers in [8] conducted a black-box analysis of various implementations of WPA3 and EAP-pwd to test their handling of edge cases. They also performed side-channel analysis using a tool called MicroWalk to detect timing and cache leaks. The researchers also created a tool that spoofs commit frames and measures the AP's response times. Sources [8] and [9] used the "FLUSH+RELOAD" technique to test their found cache-based side-channel attacks. For more details refer to section 6.5.

Source [10] details simulations and analysis of time-memory trade-off attacks against WPA3-PK. The researchers evaluated the performance of baseline attacks and explored improvements using rainbow tables. They implemented and evaluated a baseline time-memory trade-off attack, measuring its success rate and computational cost. They found that breaking WPA3-PK under its lowest security settings would require an amortized cost of less than 12 days with a nearly 50% success rate. Further they implemented a proof-of-concept of a rainbow table attack, comparing different parameters to evaluate the performance improvement over the baseline attack. They found that using multiple sub-tables with unique reduction functions significantly improved the success rate of password lookups.

6.1.3 Conclusion

The sources collectively demonstrate that brute-force attacks against WPA3 are a practical threat, particularly when combined with side-channel information. The need for constant-time implementations, careful consideration of cryptographic library choices, and the adoption of formal verification techniques are crucial for strengthening WPA3 security and mitigating the risks posed by brute-force attacks.

6.2 Spoofing and Evil Twins

Papers relevant for this section:

- [11] WiFi vulnerability caused by SSID forgery in the IEEE 802.11 protocol
- [12] Preamble Injection and Spoofing Attacks in Wi-Fi Networks
- [13] Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation
- [14] Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3
- [15] Deciphering WEP, WPA, and WPA2 Pre-shared Keys Using Fluxion

6.2.1 Scope and Focus

Thoughts on Evil Twins: Evil Twins often operate in ways that reflect all the characteristics of spoofing, which is why they are being discussed within the Spoofing section of this work. Spoofing is used in cyber-attacks and always includes someone or something pretending to be a different, trustworthy party or entity. Similarly, Evil Twins deceive victims and trick them into connecting to malicious access points controlled by an adversary. SSID spoofing, or as described in [11] as SSID forgery, leads to automatic and or involuntary connections to a malicious AP.

The sources in this section discuss vulnerabilities of the Wi-Fi protocol and potential security risks for users.

[14] revisits the deauthentication attacks on Wi-Fi networks, highlighting their continued feasibility even in PMF-enabled WPA2-PSK and WPA3-PSK networks. It emphasizes the vulnerabilities in Protected Management Frames (PMF), which were introduced as part of the IEEE

802.11w amendment to address spoofing-related attacks in management frames. They demonstrate how DoS attacks can be carried out by exploiting these vulnerabilities, as further discussed in 6.6.

Source [12] focuses on Preamble Injection and Spoofing (PrInS) attacks, which exploit the fact that the preamble of a Wi-Fi frame is weakly protected. An adversary can inject forged preambles without any payload, disrupting legitimate receptions or forcing legitimate users to defer their transmissions, ultimately silencing the channel. Among other things, this leads to DoS, as further described in section 6.6.

[13] displays how spoofing can be utilized to inject arbitrary frames, which in turn allows an attacker to obtain further attack vectors, such as DoS opportunities. By spoofing aggregated frames (manipulating a normal 802.11 frame), an attacker can trick a receiver into processing it as an A-MSDU frame. This can further be exploited for actions such as port scanning or redirecting a victim to a malicious DNS server.

Source [11] examines vulnerabilities related to SSID forgery. This vulnerability arises because devices often automatically connect to known networks based on SSID alone, without verifying the network's authenticity. An attacker can exploit this by creating a network with the same SSID as a trusted network, allowing them to intercept and potentially manipulate a victim's traffic.

While all sources address Wi-Fi security issues, [12] examines attacks on the physical layer. [15] on the other hand demonstrates that with the help of the tool Fluxion social engineering attacks on Wi-Fi credentials are also possible. The sources highlight the potential for malicious actors to exploit weaknesses in the Wi-Fi protocol to compromise user security and privacy.

6.2.2 Methods and Experiments

The research in [12] employs a combination of theoretical analysis and experimental evaluation to investigate the vulnerabilities of Wi-Fi preambles and the effectiveness of PrInS attacks. The authors analyze the IEEE 802.11 protocols to identify weaknesses in the preamble's protection mechanisms. They then develop theoretical models of PrInS attacks, considering various attack timings and power levels. To validate the practicality and impact of PrInS attacks, the authors conduct experiments using Software-defined Radios (SDRs). They set up a realistic indoor lab environment to evaluate the performance of PrInS attacks under different scenarios, measuring metrics such as throughput ratio and packet error rate.

The research of [11] relies on protocol analysis and vulnerability testing to demonstrate the security risks associated with SSID forgery. The authors examine the IEEE 802.11 standard to highlight the lack of message authenticity verification. They point out that devices solely rely on the SSID to identify and connect to networks, creating an opportunity for attackers to spoof SSIDs. To illustrate the vulnerability, the authors set up a test environment with two access points (one trusted and one malicious) and client devices. They demonstrate how an attacker can use a Raspberry Pi to create a rogue access point ([23] demonstrates tracking of mobile rogue access points, also see section 5.7) with the same SSID as a legitimate network. They then simulate a deauthentication attack to force a client device to connect to the attacker's network.

6.2.3 Conclusion

[12] takes a more technical and quantitative approach, analyzing the protocol's inner workings and using experimental data to demonstrate the impact of PrInS attacks. The authors conducted their experiments in a controlled lab environment primarily using SDRs. While SDRs offer great flexibility, they might not fully represent the behavior of commercial Wi-Fi chipsets. Also, real-world Wi-Fi networks often experience more complex interference patterns, especially in public or high-density residential areas. These factors likely affect the success rates and effectiveness of

PrInS attacks. The authors mainly focused on 802.11a/ac networks. While other generations of the Wi-Fi standard might be vulnerable as well, the specific details may vary and more targeted research would be needed. To answer the primary question in our review work, real-world application of the presented attacks and the respective evaluation of threats posed by the vulnerability in such a dynamic environment, are answered by [12].

[11] adopts a more practical and illustrative approach, using a testbed to showcase the steps involved in exploiting the SSID forgery vulnerability. The researchers successfully identified and tested the vulnerabilities with real-world conditions and reasonable setups. They demonstrated how to set up an evil twin AP, force the victim's device to disconnect from the trusted AP and connect to the malicious one. Concluding their tests with the design and implementation of an application used to monitor the network information in the background, with the goal to detect signs of deauthentication attacks and to alert the user. This nicely shows possible means to counteract threats posed by the described vulnerability.

These methodological differences are aligned with the distinct research aims of each source. [12] seeks to identify and characterize a novel attack vector, while [11] aims to raise awareness of a known vulnerability and its potential consequences.

6.3 Man-in-the-Middle

Papers relevant for this section:

- [16] Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues.
- [17] Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems
- [18] Man-in-the-Middle Attacks without Rogue AP: When WPAs Meet ICMP Redirects
- [19] Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2
- [20] Release the Kraken: New KRACKs in the 802.11 Standard

6.3.1 Scope and Focus

The sources demonstrate how vulnerabilities in Wi-Fi networks can be exploited to perform MitM attacks.

[16] reveals that by manipulating transmit queues, attackers can force an AP to send frames using a compromised or predictable security context, such as an all-zero key or the group key. This essentially strips away the intended encryption, allowing the attacker to intercept and decrypt the frames. This can lead to hijacking TCP connections or intercepting sensitive information like client and web traffic.

[17] describes how to abuse the Channel Switch Announcement (CSA) element execute MitM attacks. They differ in who is getting targeted but all of them rely on CSA beacon frames being sent to the target to deceive it to switch to the attackers AP on a different channel.

[18] focuses on a different approach: Exploiting the interaction between Wi-Fi security (WPA/WP A2/WPA3) and ICMP redirects. By spoofing the legitimate AP, attackers can send forged ICMP redirect messages to a victim client.

The studies underline the necessity of addressing these vulnerabilities through a combination of improved security standards, robust implementations, and layered security measures.

6.3.2 Methods and Experiments

The sources employ a combination of theoretical analysis and practical experimentation to demonstrate the vulnerabilities and their impact.

In [16] the authors analyze the 802.11 standard, focusing on its guidance for managing the security context of buffered frames. The authors develop attacks to demonstrate how power-save

features can be exploited to leak frames. They trigger these leaks by manipulating the security context, forcing the AP to transmit frames in plaintext, with the group key, or an all-zero key. They evaluate these attacks against a range of devices and operating systems, showcasing their widespread impact. The authors demonstrate the practical consequences of these vulnerabilities by showcasing how they can be used to hijack TCP connections, intercept client traffic, and even compromise web traffic.

To prove their claims the researcher in [17] use a symbolic Model Checker (MC) which is used to verify if a given system model satisfies a desired property. In this case the authors of the source utilize NuSMV to implement and verify their model of Wi-Fi's connection establishment process. They use their created MC to test different scenarios in which they force the station or the access point to switch the channel.

The researchers in [18] identify a vulnerability in the Network Processing Unit (NPU) used in many AP routers. This vulnerability prevents these routers from blocking forged ICMP redirect messages, even when they are spoofed to appear as if they originated from the AP itself. The authors develop a technique to craft ICMP redirect messages that can evade legitimacy checks in a wide range of operating systems. This involves embedding a fake UDP header with an active source port to circumvent the checks that modern operating systems employ. The authors test 55 popular wireless routers from 10 prominent vendors, finding that none could block the crafted ICMP redirect messages due to the identified NPU vulnerability. They also test their attack against 122 real-world Wi-Fi networks, encompassing different security modes (WPA2 and WPA3), both IPv4 and IPv6 environments, and diverse locations. The attack achieved a high success rate of 89%, demonstrating its practical effectiveness

6.3.3 Conclusion

[16] and [18] move beyond theoretical vulnerabilities and demonstrate practical attacks that exploit the identified weaknesses. The research on the MitM attack in Wi-Fi networks presents a well-executed analysis of a critical security vulnerability. They conduct experiments in real-world settings, evaluating the impact of the attacks against actual devices and networks. This approach provides strong evidence of the vulnerabilities' significance and potential for harm. While focusing on different attack vectors, both sources highlight the importance of properly managing security contexts in Wi-Fi networks.

While the formal analysis in [17] provides extensive coverage of potential attack scenarios, the empirical validation is limited especially the found MitM attacks. Future work could involve expanding empirical testing to validate the identified attacks.

The researchers' proactive approach in [18] to responsible disclosure shows best practices in security research. By notifying affected parties before publication, they allowed vendors and network operators to address the vulnerability. Their experiments were conducted ethically, with consent from network operators and safeguards to avoid affecting uninvolved users.

These works are a strong contribution to Wi-Fi security research, effectively communicating the vulnerability's significance and offering actionable solutions.

KRACK Attack Requires a MitM Position

To be able to execute the KRACK attack that the sources [19] and [20] mention a MitM position is required. First in [19], the authors employed channel-based MitM attack techniques to manipulate handshake messages and trigger key reinstallations. This attack method relies on cloning the legitimate AP on a different channel and forcing the client to connect to the rogue access point, allowing the attacker to intercept and manipulate traffic. Later in [20], they propose using forged CSAs as a more practical method to establish the MitM position needed for the attack (as

described in [17]). Forging CSAs tricks clients into switching to a rogue channel controlled by the attacker, eliminating the need for specialized jamming equipment.

As these two papers about KRACK (2017 and 2018) were published before [17] (2022), [16] (2023), and [18] (2023), they might give new insights in how to reach a MitM position.

6.4 Decrypting or Forging Packets

Papers relevant for this section:

- [19] Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2
- [20] Release the Kraken: New KRACKs in the 802.11 Standard
- [13] Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation

6.4.1 Scope and Focus

The sources are all about attacks which cause an attacker to be able to decrypt or forge packets. They all are related to flaws in the 802.11 standard.

[19] and [20] describe the KRACK attack, which exploits the vulnerability of key reinstallation in WPA2. This attack can be used to decrypt, replay, and potentially forge packets.

When a key is reinstalled, the associated nonce and replay counter are reset. This allows an attacker to replay previously sent packets. Since all three data-confidentiality protocols (TKIP, CCMP, and GCMP) use stream ciphers to encrypt frames, reusing a nonce means reusing the keystream, allowing an attacker to decrypt the packets as well.

While replaying packets is possible with all three protocols, forging packets depends on the specific protocol and handshake targeted. With GCMP, the attacker can forge packets in both directions because the authentication key can be recovered and is used to protect both communication directions while with TKIP the attacker can forge packets in only one direction. The direction of forgery depends on the handshake targeted. For example, attacking the 4-way handshake allows forging packets from the client to the AP, while attacking the Fast BSS Transition (FT) handshake allows forging packets from the AP to the client.

In the source [13] the frame fragmentation is attacked. A mixed key attack exploits the fact that 802.11 does not require receivers to check if all fragments of a frame are encrypted under the same key. This allows an attacker to forge frames by mixing fragments of frames that were encrypted under different keys. An adversary can then exfiltrate data, potentially recovering sensitive info sent over plaintext HTTP. It is also possible to poison the fragment cache by exploiting the fact that the 802.11 standard does not specify when receivers should remove decrypted fragments from memory (the fragment cache). An attacker can inject fragments into a victim's fragment cache and combine them with legitimate fragments to forge packets or exfiltrate data.

6.4.2 Methods and Experiments

All sources tested their attack methods against a range of devices, operating systems, and wireless network cards. This practical evaluation validates their findings and highlights the widespread impact of the found vulnerabilities.

The sources [19] and [20] do not explicitly state what setup and tools they used to conduct their experiments. But on their website [25] they link to their GitHub repository [26] where they made scripts available which help to detect whether an implementation is vulnerable to KRACK.

In source [13] the researchers analyzed the code of leaked and open-source network stacks to uncover implementation flaws related to aggregation and fragmentation. This approach helped identify specific code segments responsible for vulnerabilities and understand the underlying causes of insecure behavior. Additionally, they developed a custom tool capable of testing both client devices and APs for vulnerabilities. This tool encompasses over 45 test cases and can evaluate

the security of both home and enterprise networks employing various authentication methods like PEAP-MSCHAPv2 and Extensible Authentication Protocol - Transport Layer Security (EAP-TLS).

6.4.3 Conclusion

The sources highlight the importance of formally modeling and verifying the security properties of 802.11 protocols. This includes modeling key installation procedures, interactions between different features, and potential race conditions to ensure robust security. Rigorous testing of implementations is essential to identify and eliminate vulnerabilities like race conditions and insecure handling of handshake messages. This includes testing under various network conditions and adversarial scenarios. They also emphasize the complex and evolving nature of cybersecurity, urging continued research, vigilance, and adaptation to address emerging threats.

6.5 Side-Channel Attacks in WPA3

Papers relevant for this section:

- [8] Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd
- [9] From Dragondoom to Dragonstar: Side-channel Attacks and Formally Verified Implementation of WPA3 Dragonfly Handshake

6.5.1 Scope and Focus

The papers [8] and [9] examine the security of the Dragonfly handshake, a critical component of WPA3 and EAP-pwd protocols designed to enhance the security of Wi-Fi networks. Both papers offer complementary perspectives on the security challenges inherent in Dragonfly implementations. [8] focuses on the implementation of the Dragonfly handshake and possible side-channel attacks, while [9] takes a closer look at external libraries used in the Dragonfly handshake. Both papers converge on the significant threat posed by side-channel attacks, specifically timing and cache attacks.

6.5.2 Methods and Experiments

Timing Attacks: Timing attacks exploit variations in the time required to complete specific operations, correlating these variations with secret information.

Both papers discuss timing attacks that leverage variations in the number of iterations needed in the "hunting-and-pecking" password conversion algorithm. This algorithm iteratively searches for a valid point on an elliptic curve. The time taken to find this point can leak information about the password, especially in implementations that do not employ constant-time techniques.

Source [8] also introduces a novel timing attack that specifically targets implementations using Brainpool curves. Even though the hash-to-curve method incorporates defenses against known timing leaks, these defenses are less effective with Brainpool curves. This vulnerability arises because the prime number used in Brainpool curves is not close to a power of two, leading to secret-dependent variations in execution time when checking if the output of a Key Derivation Function (KDF) is smaller than the prime.

Cache Attacks: Cache attacks exploit the processor's cache, a fast memory that stores frequently accessed data, to infer information about secret data.

Source [9] demonstrates that commonly used cryptographic libraries like OpenSSL, WolfSSL, and ell, while generally secure, can introduce vulnerabilities when used within the Dragonfly handshake. These libraries may contain functions, such as point decompression and binary-to-bignum conversion, that are not implemented in a constant-time manner. When these functions

are called with secret-dependent values, their interaction with the processor's cache can leak information about the secrets.

Both sources employ the "Flush+Reload" technique to carry out these cache attacks. This technique involves repeatedly flushing a specific memory location from the cache and then measuring the time it takes to reload that location. If the target process accesses the flushed location during the measurement period, the reload time will be short, indicating a cache hit. By observing the pattern of cache hits and misses, attackers can infer information about the target's memory access patterns, including secret data.

In the source [9] a novel "Flush+Reload-gadget" is introduced that enhances the effectiveness of cache attacks by exploiting the processor's instruction prefetching mechanism. This technique overcomes the spatial limitations of traditional Flush+Reload attacks, enabling more precise measurements and increased information leakage.

6.5.3 Conclusion

While both sources examine the security of the Dragonfly handshake, they differ in their perspectives. One focuses more on the implementation of the Dragonfly handshake itself, while the other one takes a close look at external libraries used in the Dragonfly handshake.
Their analysis of the Dragonfly handshake and the examination of the found side-channel attacks seem to be conducted with care and attention to detail, but there might still be more side-channel attacks which have not been discovered yet.
In light of these vulnerabilities, the sources advocate for secure implementation practices and the use of formal verification techniques to ensure the robustness of Dragonfly implementations.

6.6 Denial-of-Service

Papers relevant for this section:

- [8] Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd
- [12] Preamble Injection and Spoofing Attacks in Wi-Fi Networks
- [13] Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation
- [14] Cut It: Deauthentication Attacks on Protected Management Frames in WPA2 and WPA3
- [17] Systematically Analyzing Vulnerabilities in the Connection Establishment Phase of Wi-Fi Systems
- [16] Framing Frames: Bypassing Wi-Fi Encryption by Manipulating Transmit Queues.

6.6.1 Scope and Focus

Many cyber-attacks can directly or indirectly lead to DoS. Although the primary research goals of the cited sources may not exactly aim to achieve DoS, some techniques could potentially be used to do just that. DoS is not very stealthy and generally easy to detect once started. The researchers therefore often not only point out possible DoS vulnerabilities in the standards but also construct more advanced experiments based on said vulnerabilities.

[8] evaluates the security of the Dragonfly handshake, a mechanism used in WPA3 and EAP-pwd to secure Wi-Fi networks by providing forward secrecy and resistance to dictionary attacks. The work additionally shows that DoS attacks are also possible in WPA3.

[12] addresses vulnerabilities in the preamble structure of Wi-Fi networks, emphasizing its lack of guarantees for authenticity or confidentiality. It introduces novel PrInS attacks, which exploit these vulnerabilities to disrupt legitimate communications across different Wi-Fi versions. The

work highlights the impact of these attacks on network reliability and performance, as well as DoS potential.

[13] identifies design flaws, specifically in the frame aggregation and fragmentation functionalities, which allow adversaries to forge encrypted frames. Whilst not using the vulnerabilities to conduct DoS attacks, the source is still being mentioned in this section because it could very well be used in a DoS. The study emphasizes that these flaws have existed since Wi-Fi's inception in 1997 and affect all Wi-Fi security protocols, including WEP, WPA, WPA2, and WPA3.

[14] revisits the deauthentication attacks on Wi-Fi networks, highlighting their continued feasibility even in PMF-enabled WPA2-PSK and WPA3-PSK networks. It emphasizes the vulnerabilities in PMF, which were introduced as part of the IEEE 802.11w amendment to address spoofing-related attacks in management frames. The research focuses on how the launch of deauthentication attacks leads to the disconnection of legitimate devices from the Wi-Fi network, effectively resulting in DoS.

[17] analyzes the connection establishment phase in Wi-Fi networks, during which unprotected management frames are exchanged before authentication and session protection begin. It identifies a previously unknown DoS vulnerability and new variants of a known MitM attack (see section 6.3), highlighting security risks in the latest IEEE 802.11 standard.

[16] examines how Wi-Fi access points manage the security contexts of queued frames, particularly in relation to power-save features. It identifies fundamental design flaws in the 802.11 standard, including the unprotected nature of the power-save bit, enabling adversaries to leak frames in plaintext, manipulate encryption contexts, and execute DoS attacks. The work emphasizes the widespread impact on devices and operating systems such as Linux, FreeBSD, iOS, and Android.

6.6.2 Methods and Experiments

The researchers in [8] found that the anti-clogging mechanism in SAE, intended to prevent flooding attacks, is ineffective because MAC addresses are easily spoofed, and secret cookies can be captured and replayed. This allows an attacker to overload the AP's CPU even with defenses in place. They conducted experiments using a Raspberry Pi as the attacker and a professional AP as the target. They successfully demonstrated that spoofing a relatively small number of commit exchanges per second could push the AP's CPU usage to 100%, effectively denying service to legitimate clients.

In [12] the researchers validated the PrInS attacks experimentally using SDRs. They demonstrated that adversaries could nearly silence the communication channel, reducing legitimate users' throughput to 2% of its normal level. Even with a 30 dB lower signal power, the attack caused an 87% reduction in throughput, proving its efficacy under various conditions.

The researchers of [13] analyzed the 802.11 standard and implementations to uncover design and common implementation flaws related to aggregation and fragmentation. Experiments showed that all tested devices were vulnerable to at least one attack, demonstrating the widespread impact of these flaws across all Wi-Fi devices. They discuss driver/firmware patches that are needed to ensure success of the attack [13, Sec. 4.6]. The attack could effectively be used for DoS by injecting packets. They also highlight the usage of spoofed MAC addresses of a trusted network to advertise the SSID of an untrusted network (rogue access point) [13, Sec. 5.3].

[14] simulated various attack scenarios to demonstrate how deauthentication attacks could bypass the protections offered by PMF-enabled networks. Through these experiments, they identified gaps in the implementation of IEEE 802.11w and WPA3's mechanisms, providing interpretations to explain why these attacks remain effective. They used a number of different devices and created real-world conditions in their experiments, showing real impacts of the vulnerabilities discussed. The authors explore different attack scenarios (including code snippets in [14, Table

3]), analyze the reasons behind their success, and effectively demonstrate the impacts of when applied in DoS.

In [17] formal modeling was used to analyze the connection establishment phase and validate the identified vulnerabilities through experiments. The researchers tested the performance of the Operating Channel Validation (OCV) technique using an iPhone 6s (with iOS 13.3.1) and Aruba AP, capturing frame arrival times with Wireshark. For DoS attack testing, they used a virtual machine with Ubuntu 20.04.3 and integrated Wi-Fi frameworks but could not assess battery drain effects due to virtual environment limitations. Testing against the latest WPA-supplicant daemon demonstrates that an adversary can prevent a station from connecting to a preferred access point for up to 90 minutes or more, proving the feasibility of the DoS attack.

[16] The researchers exploit power-save features and design flaws in hotspot-like networks to force access points into leaking frames or encrypting them with weak or adversary-controlled keys. By abusing the power-save bit it is possible to interrupt connections of legitimate clients connected to the AP. When repeating this procedure the client is effectively unable to adequately communicate with the network, resulting in DoS. They validate these vulnerabilities through attacks that demonstrate hijacking TCP connections, intercepting client traffic, and bypassing Wi-Fi encryption. Their tests involve various open-source network stacks and highlight inconsistent security practices across layers.

6.6.3 Conclusion

All sources discuss potential solutions to mitigate the identified vulnerabilities.
The researchers of [8] suggest multiple strategies to fix this vulnerability like making the password elements independent of the peers' identities. They also suggest using a more efficient hash-to-curve method.

To mitigate the PrInS attacks, the paper [12] proposes a backward-compatible preamble authentication scheme. This solution aims to enhance the integrity and security of Wi-Fi preambles while maintaining compatibility with existing devices and protocols.

[13] introduces a testing tool to help identify devices affected by these vulnerabilities and discusses potential countermeasures to mitigate the attacks. These countermeasures include changes to the standard and implementation practices improving frame aggregation and fragmentation security.

[14] suggests ensuring robust handling of SA-query requests and responses by Wi-Fi devices. The source highlights that the attacks' success often stems from the access point concluding the session due to a lack of timely SA-query responses. This points to a potential weakness in how devices handle a high volume of SA-query requests, especially when interfered with by malicious frames. The authors recommend manufacturers rigorously test their 802.11w implementations to ensure they can withstand such situations.

[17] proposes a mitigation approach to address the identified DoS vulnerability, emphasizing the need for more robust protections during the unprotected connection establishment phase. Additionally, it evaluates the optional operating channel validation technique in the IEEE 802.11 standard, showing that it effectively protects only against multi-channel MitM attacks.

[16] calls for improved transparency in handling security contexts across network stack layers and stresses the need for explicit guidance in the 802.11 standards. It highlights the challenges of managing security contexts for queued frames and advocates for design revisions to address these vulnerabilities, including stronger protection for the power-save bit and stricter encryption mechanisms.

The sources contribute significantly to understanding the ongoing security challenges in Wi-Fi technology, providing valuable insights for both researchers and those responsible for securing Wi-Fi networks. This demonstrates that despite continuous efforts to improve Wi-Fi security,

vulnerabilities persist and require ongoing attention and mitigation strategies. Some suggest countermeasures while also offering insights into attack execution and interpretation.

Even though WPA3 tried to prevent DoS attacks by implementing anti-clogging mechanism, [8] showed that this prevention can be easily avoided and DoS is still feasible in WPA3.

While all sources target Wi-Fi security, they examine different aspects of the 802.11 standard and its implementations.

In relation to WPA2, [14] zooms in on the SA-query procedure and deauthentication attacks, while [13] uncovers fundamental design flaws in frame handling mechanisms that can further be used to facilitate DoS attacks.

Despite their distinct focuses, the sources are comparable in several ways. It is demonstrated that even with continuous security improvements like WPA3 and PMF, Wi-Fi remains susceptible to attacks. This underscores the need for continuous vigilance and ongoing research in Wi-Fi security. The sources call for a multipronged approach to addressing Wi-Fi vulnerabilities, emphasizing the importance of robust implementations, rigorous testing, and prompt security updates. They advocate for formal analysis of security mechanisms to enhance confidence in proposed defenses.

7 Conclusion

7.1 Shortcomings in the 802.11 Standard

Despite significant progress in Wi-Fi security, the research reveals recurring vulnerabilities that stem from both design flaws in the standards and implementation-specific weaknesses. Key areas for improvement include standardization processes, formal verification of implementations, mitigation of side-channel attacks, and transparent handling of security contexts.

7.2 Shortcomings in the reviewed sources

Many of the presented attacks exploit design flaws in fundamental elements of the Wi-Fi standard, such as frame management, e.g., aggregation and fragmentation [13], or key reinstallation [19], [20]. Even though some issues, like management frame spoofing, have been partially addressed in a new standard (IEEE 802.11w with Protected Management Frames), these improvements also introduce surfaces for new vulnerabilities [14].

Another recurring topic is the presence of vulnerabilities due to flawed implementations by vendors. For instance, improper handling of cryptographic handshakes, such as Dragonfly, lead to side-channel attacks, timing leaks, and predictable behaviors [8]. This reveals a gap in testing and verifying implementations by some vendors. While some countermeasures and protocol fixes are proposed (e.g., Dragonstar leveraging formally verified cryptographic libraries [9]), the challenge of fully resolving vulnerabilities without overhauling existing protocols gets acknowledged as well [16], [20].

Even though many of the attacks were validated experimentally (e.g., using specific devices or testbeds), the scope of real-world testing in some were limited [17], [7], [12]. This leaves uncertainty about how well these vulnerabilities translate into practical exploits across diverse Wi-Fi ecosystems.

7.3 Improvements and Further Research

The success of formally verified cryptographic libraries, as demonstrated by Dragonstar [9], underscores the importance of applying formal methods to eliminate leakage vectors and implementation bugs. Future research should focus on creating formally verified implementations for all critical components of Wi-Fi protocols especially the authentication and key exchange part (handshakes). Even though protocols like WPA3-PK attempt to improve user authentication they remain vulnerable to insider threats and password brute-forcing [10].

Additionally, several vulnerabilities arise from poorly managed security contexts (e.g., frame queuing [16] and key management [19], [20], [13]). Researchers should investigate transparent mechanisms for handling security contexts across all layers of the Wi-Fi stack, emphasizing clear guidelines for developers.

Another issue comes with the need for backward compatibility of security standards. Downgrade attacks (e.g., WPA3 to WPA2) [7], [8] can result in old vulnerabilities even though a newer standard like WPA3 would provide more security. It is understandable that older standards still have to be supported, but researchers should make sure, that this kind of attacks are not possible by testing downgrade attacks.

Acronyms

A-MSDU Aggregate MAC Service Data Unit. 12, 25

AES Advanced Encryption Standard. 1

AP Access Point. 5, 6, 7, 8, 13, 14, 15, 17, 18, 19, 20, 21, 22, 24, 26, 27, 28, 31, 32

ARP Address Resolution Protocol. 21

CCMP Counter Mode Cipher Block Chaining Message Authentication Code Protocol. 28

CPU Central Processing Unit. 9, 31

CSA Channel Switch Announcement. 6, 13, 26, 27, 28

DNS Domain Name System. 12, 17, 20, 25

DoS Denial-of-service. 9, 13, 14, 15, 23, 25, 30, 31, 32, 33

EAP Extensible Authentication Protocol. 37

EAP-TLS Extensible Authentication Protocol - Transport Layer Security. 29

EAPOL Extensible Authentication Protocol over LAN. 6

FILS Fast Initial Link Setup. 6

FT Fast BSS Transition. 5, 28

GCMP Galois Counter Mode Protocol. 28

HACL High-Assurance Cryptographic Library. 16

HTTP Hypertext Transfer Protocol. 20, 28

HTTPS Hypertext Transfer Protocol Secure. 20

ICMP Internet Control Message Protocol. 17, 18, 26, 27

IEEE Institute of Electrical and Electronics Engineers. 1, 7, 14, 15, 24, 25, 31, 32, 34

IP Internet Protocol. 17, 18, 20, 27

KDF Key Derivation Function. 9, 29

KRACK Key Reinstallation Attack. 5, 6, 27, 28

MAC Medium Access Control. 9, 14, 18, 19, 20, 21, 31

MC Model Checker. 13, 27

MitM Man-in-the-Middle. 5, 6, 13, 17, 18, 23, 26, 27, 28, 31, 32

MLE Mean Localization Error. 12

MNLE Mean Next Location Error. 12

MRAP Mobile Rogue Access Point. 11, 12

NIC Network Interface Card. 5

NPU Network Processing Unit. 18, 19, 27

OCV Operating Channel Validation. 13, 14, 32

OS Operating System. 5, 17

PMF Protected Management Frames. 14, 24, 31, 33, 34

PMK Pairwise Master Key. 23

PrInS Preamble Injection and Spoofing. 13, 25, 26, 30, 31, 32

PSK Pre-shared Key. 23

PTK Pairwise Transient Key. 5, 6

RAP Rogue Access Point. 11, 12

RSNE Robust Security Network Element. 10

RSSI Received Signal Strength Indicator. 11

SAE Simultaneous Authentication of Equals. 9, 16, 23, 31, 38

SDR Software-defined Radio. 13, 25, 31

SSID Service Set Identifier. 7, 21, 22, 23, 24, 25, 26, 31

SSWU Simplified Shallue-Woestijne-Ulas. 16

TCP Transmission Control Protocol. 19, 26, 27, 32

TDLS Tunneled Direct-Link Setup. 6

TKIP Temporal Key Integrity Protocol. 28

TLS Transport Layer Security. 20

TPK Tunneled Direct-Link Setup (TDLS) PeerKey. 6

UDP User Datagram Protocol. 17, 27

WEP Wired Equivalent Privacy. 1, 10, 12, 19, 31

WLAN Wireless Local Area Network. 1, 10

WNM Wireless Network Management. 6

WPA Wi-Fi Protected Access. 1, 4, 5, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 20, 23, 24, 26, 27, 28, 29, 30, 31, 33, 34

Glossary

EAP-pwd Is an EAP authentication method that uses a shared password for authentication. It addresses the problem of password-based authenticated key exchange using a possibly weak password for authentication to derive an authenticated and cryptographically strong shared secret. . 8, 9, 24, 29, 30

EAP-TLS Extensible Authentication Protocol - Transport Layer Security is the authentication protocol most commonly deployed on WPA2-Enterprise networks to enable the use of X.509 digital certificates for authentication. It's a mutual authentication method, where both the client and the server need certificates for successful authentication. Once those certificates are identified, the Extensible Authentication Protocol - Transport Layer Security will create session-based keys that each party can use to complete the login. . 7

hash-to-curve Is a cryptographic method that deterministically maps arbitrary data to a point on an elliptic curve, ensuring uniform distribution for use in elliptic curve cryptography. . 9

hash-to-group Is a process that maps arbitrary data to an element in a mathematical group (not limited to elliptic curves), ensuring the output is uniformly distributed and secure for cryptographic protocols. . 9

MODP Refers to pre-defined groups of prime numbers and generators used in cryptographic protocols, like Diffie-Hellman key exchange, to ensure secure key generation and exchange processes. . 9

nonce An arbitrary number used only once in a cryptographic communication. . 5, 6

RADIUS Remote Authentication Dial-In User Service is a networking protocol that provides centralized authentication, authorization, and accounting (AAA) management for users who connect and use a network service. . 8, 20

SAE-PK Simultaneous Authentication of Equals - Public Key is an extension of the SAE protocol in Wi-Fi networks, which enhances security by enabling authentication without relying on a shared password. Instead, it uses a public/private key pair embedded in the network to authenticate devices securely, protecting against attacks like dictionary attacks and rogue network impersonation. . 21, 23

SAE-PT Simultaneous Authentication of Equals - Password Transmission is a variant of the SAE authentication protocol in Wi-Fi, specifically designed to securely transmit passwords between devices during the handshake process, while still maintaining resistance to offline dictionary attacks and providing strong security. . 16

SA-query Is a mechanism used in networking, particularly in protocols like IEEE 802.11 (Wi-Fi), to check the status or validate the existence of a Security Association (SA) between two devices, ensuring the continuation or re-establishment of secure communication. . 15, 32, 33

SAE Simultaneous Authentication of Equals is based on the Dragonfly handshake protocol and enables the secure exchange of keys of password-based authentication methods. In WPA3, Simultaneous Authentication of Equals replaces the previous methods of negotiating session keys using PSK. . 8, 16

WPA2-PSK WPA2-Personal mode, which is used to protect network access and data transmission by using an AES (Advanced Encryption Standard) or TKIP (Temporal Key Integrity Protocol) encryption method. . 14, 24, 31

WPA3-PK See term SAE-PK. Used in source [10] to refere to the term SAE-PK. . 21, 22, 24, 34

WPA3-PSK WPA3-Personal mode, which is used to protect network access and data transmission by using Simultaneous Authentication of Equals (SAE), often called WPA3-SAE. . 14, 24, 31

References

[1] S. McCann, *Official ieee 802.11 working group project timelines - 2024-11-21*, IEEE Standards Association, Accessed: Dec. 08, 2024. [Online]. Available: https://www.ieee802.org/11/Reports/802.11_Timelines.htm.

[2] Wi-Fi Alliance, *Wi-fi alliance® celebrates 25 years of wi-fi® innovation and impact*, Wi-Fi Alliance, Accessed: Dec. 08, 2024. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-celebrates-25-years-of-wi-fi-innovation-and-impact.

[3] Z. Xia, *What is wlan?* Huawei Technologies Co., Ltd., Accessed: Dec. 08, 2024. [Online]. Available: https://info.support.huawei.com/info-finder/encyclopedia/en/WLAN.html.

[4] Wi-Fi Alliance, *Wi-fi protected access security sees strong adoption*, Wi-Fi Alliance, Accessed: Nov. 10, 2024. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-protected-access-security-sees-strong-adoption.

[5] Wi-Fi Alliance, *Wpa2™ security now mandatory for wi-fi certified™ products*, Wi-Fi Alliance, Accessed: Nov. 10, 2024. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wpa2-security-now-mandatory-for-wi-fi-certified-products.

[6] Wi-Fi Alliance, *Wi-fi alliance® introduces wi-fi certified wpa3™ security*, Wi-Fi Alliance, Accessed: Nov. 10, 2024. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-introduces-wi-fi-certified-wpa3-security.

[7] E. Baray and N. Kumar Ojha, "Wlan security protocols and wpa3 security approach measurement through aircrack-ng technique," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 23–30. DOI: 10.1109/ICCMC51019.2021.9418230.

[8] M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the dragonfly handshake of wpa3 and eap-pwd," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 517–533. DOI: 10.1109/SP40000.2020.00031.

[9] D. De Almeida Braga, N. Kulatova, M. Sabt, P.-A. Fouque, and K. Bhargavan, "From dragondoom to dragonstar: Side-channel attacks and formally verified implementation of wpa3 dragonfly handshake," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, 2023, pp. 707–723. DOI: 10.1109/EuroSP57164.2023.00048.

[10] M. Vanhoef and J. Robben, "A security analysis of wpa3-pk: Implementation and precomputation attacks," in *Applied Cryptography and Network Security*, Springer Nature Switzerland, 2024, pp. 217–240, ISBN: 978-3-031-54773-7. DOI: 10.1007/978-3-031-54773-7_9.

[11] K. Juhász, V. Póser, M. Kozlovszky, and A. Bánáti, "Wifi vulnerability caused by ssid forgery in the ieee 802.11 protocol," in *2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, 2019, pp. 333–338. DOI: 10.1109/SAMI.2019.8782775.

[12] Z. Zhang and M. Krunz, "Preamble injection and spoofing attacks in wi-fi networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6. DOI: 10.1109/GLOBECOM46510.2021.9685461.

[13] M. Vanhoef, "Fragment and forge: Breaking Wi-Fi through frame aggregation and fragmentation," in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 161–178, ISBN: 978-1-939133-24-3. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/vanhoef.

[14] K. Lounis, S. H. Ding, and M. Zulkernine, "Cut it: Deauthentication attacks on protected management frames in wpa2 and wpa3," in *Foundations and Practice of Security*, Springer International Publishing, 2022, pp. 235–252, ISBN: 978-3-031-08147-7. DOI: 10.1007/978-3-031-08147-7_16.

[15] S. Atluri and R. Rallabandi, "Deciphering wep, wpa, and wpa2 pre-shared keys using fluxion," in *Smart Computing Techniques and Applications*, Springer Singapore, Jul. 2021, pp. 377–385, ISBN: 978-981-16-0878-0. DOI: 10.1007/978-981-16-0878-0_37.

[16] D. Schepers, A. Ranganathan, and M. Vanhoef, "Framing frames: Bypassing wi-fi encryption by manipulating transmit queues," in *32nd USENIX Security Symposium (USENIX Security 23)*, USENIX Association, Aug. 2023, pp. 53–68, ISBN: 978-1-939133-37-3. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/schepers.

[17] N. Hoque, H. Rahbari, and C. Rezendes, "Systematically analyzing vulnerabilities in the connection establishment phase of wi-fi systems," in *2022 IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 64–72. DOI: 10.1109/CNS56114.2022.9947252.

[18] X. Feng, Q. Li, K. Sun, Y. Yang, and K. Xu, "Man-in-the-middle attacks without rogue ap: When wpas meet icmp redirects," in *2023 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2023, pp. 3162–3177. DOI: 10.1109/SP46215.2023.10179441.

[19] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in wpa2," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2017, pp. 1313–1328, ISBN: 9781450349468. DOI: 10.1145/3133956.3134027.

[20] M. Vanhoef and F. Piessens, "Release the kraken: New kracks in the 802.11 standard," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Association for Computing Machinery, 2018, pp. 299–314, ISBN: 9781450356930. DOI: 10.1145/3243734.3243807.

[21] M. Higgins, *What should you look for in the most secure routers of 2024*, NordVPN, Accessed: Dec. 15, 2024. [Online]. Available: https://nordvpn.com/blog/most-secure-router/.

[22] Kaspersky, *What is krack attack and how to defend against it?* Kaspersky, Accessed: Dec. 15, 2024. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/krack.

[23] L. Qawasmeh and F. Awad, "Tracking a mobile rouge access point," in *2021 International Conference on Information Technology (ICIT)*, IEEE, 2021, pp. 522–526. DOI: 10.1109/ICIT52682.2021.9491684.

[24] hoquenaureen, *Wifi-preauthvul-analyze*, GitHub, Accessed: Nov. 29, 2024. [Online]. Available: https://github.com/hoquenaureen/wifi-preauthvul-analyze/tree/d8d97cc82be32c1d9215851ce4e44d5634b41554.

[25] M. Vanhoef, *Key reinstallation attacks breaking wpa2 by forcing nonce reuse*, Accessed: Nov. 29, 2024. [Online]. Available: https://www.krackattacks.com/.

[26] M. Vanhoef, *Krackattacks-scripts*, GitHub, Accessed: Nov. 29, 2024. [Online]. Available: https://github.com/vanhoefm/krackattacks-scripts/tree/f80d0005086cd6b0a32ff1fd236d90026587ed38.

List of Figures

List of Tables