

UNIVERSITY OF BAHRAIN

College of Information Technology



Mitigation of Wi-Fi Security Threat Using Public Key Cryptography Implementation

A Thesis Submitted in Partial Fulfilment of the Requirements for the
Master of Science Degree in Cyber Security

Submitted By

Mohamed Alhamry

20033218

Supervised By

Dr. Alauddin Alomary

(Associate Professor)

University of Bahrain

Kingdom of Bahrain

June 2023



Dissertation Deposit Form

Author Name:	Mohamed Ali Mohamed Husain Alhamry		
Email:	20033218@stu.uob.edu.bh	Contact No:	33139163
College:	Information Technology		
Department:	Information Systems		
Thesis Title:	Mitigation of Wi-Fi Security Threat Using Public Key Cryptography Implementation		
Year:	2023		

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:	Mohamed Ali Mohamed Husain Alhamry	Date:	20-Jun-23
--------------------------	------------------------------------	-------	-----------

DECLARATION

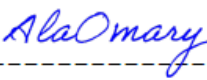
I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed:  Date: 20/06/2023
Mohamed Ali Mohamed Husain Alhamry (Candidate)

The Thesis defense committee considers the thesis titled:
Mitigation of Wi-Fi Security Threat Using Public Key Cryptography Implementation

Submitted by Mohamed Ali Mohamed Husain Alhamry to the
College of Information Technology is satisfactory and
acceptable for the Master of Science in Cyber Security.

The Defense Committee:



Dr. Alauddin Al-Omary Department of Computer Engineering
(Supervisor) University of Bahrain



Prof. Nik Bessis Edge Hill University
(External Examiner) United Kingdom



Dr. Wael Elmedany Department of Computer Engineering
(Internal Examiner) University of Bahrain

Abstract

Wi-Fi networks have become an integral part of our daily lives, with billions of devices connected to them. However, the security of Wi-Fi networks has always been a major concern due to vulnerabilities in security protocols. The current WPA3 security protocol is vulnerable to various types of attacks, including Dragonblood vulnerabilities and Evil Twin attacks, which can compromise sensitive user data. Furthermore, the public key cryptography utilized in WPA3 is susceptible to quantum attacks, which could potentially become viable in the upcoming years. To address these security issues, an effective method is needed to enhance Wi-Fi security. This thesis proposes a novel solution to address Wi-Fi protocol security by integrating two approaches: the Kyber-based Authenticated Key Exchange (KAKE) handshake and the Trust On First Use (TOFU) approach. The KAKE handshake provides security against vulnerabilities in both WPA2 and WPA3 protocols and offers post-quantum protection. Meanwhile, the TOFU approach addresses vulnerabilities in WPA3 by providing additional security against Evil Twin attacks and Downgrade attacks. The proposed solution has been evaluated for its performance, feasibility, and security through proof of concepts and compared with other literature results. The findings show that the proposed approach is effective in preventing different Wi-Fi attacks. Overall, this thesis can be valuable for researchers, network administrators, and other stakeholders interested in improving the security of Wi-Fi networks.

Contents

Dissertation Deposit Form	A
Abstract	C
Contents	D
List of Figures	J
List of Tables.....	L
Acknowledgment	M
Dedication	N
Publications	O
List of Abbreviations	P
Chapter 1: Introduction.....	2
1.1 Background and Motivation	2
1.2 Research Problem	2
1.3 Research Objective	3
1.4 Research Methodology	3
1.4.1 Research Design	3
1.4.2 Data Collection and Analysis	4
1.4.3 Limitations	5
1.5 Contribution to the Field.....	5
1.6 Thesis Outline	6
Chapter 2: Related Work.....	8
2.1 Introduction.....	8
2.2 WPA3 Weaknesses.....	8
2.3 CRYSTALS-Kyber	10

2.4	TOFU	12
2.5	Summary	16
Chapter 3: Wi-Fi Overview.....		18
3.1	Introduction.....	18
3.2	Wi-Fi Architecture	19
3.3	Wi-Fi Security Protocols	22
3.3.1	WEP	22
3.3.1.1	Authentication	22
3.3.1.2	Encryption	24
3.3.1.3	Weaknesses	25
3.3.2	WPA.....	27
3.3.2.1	Authentication	27
3.3.2.2	Encryption	28
3.3.2.3	Improvement Over WEP	29
3.3.2.4	Weaknesses	30
3.3.3	WPA2	30
3.3.3.1	Authentication and Key Exchange	30
3.3.3.1.1	Key Negotiation.....	31
3.3.3.1.2	Four-way Handshake (Key Exchange).....	32
3.3.3.1.3	Group Key Handshake (Two-way Handshake).....	34
3.3.3.2	Encryption	35
3.3.3.3	Weaknesses	37
3.3.3.3.1	Weak Password.....	37
3.3.3.3.2	Handshake Dictionary Attack.....	37

3.3.3.3.3	PMK-ID Dictionary Attack	38
3.3.3.3.4	De-authentication Attack	38
3.3.3.3.5	Evil Twin Attack	38
3.3.3.3.6	KRACK Vulnerability	39
3.3.4	WPA3	41
3.3.4.1	PMF	41
3.3.4.2	SAE Handshake	42
3.3.4.3	Transition Mode	46
3.3.4.4	Weaknesses	48
3.3.4.4.1	Downgrade Attacks	48
3.3.4.4.2	Side-channel Attacks	50
3.3.4.4.3	DoS Attacks	50
3.3.4.4.4	Lack of Post-quantum Protection	51
3.4	Summary	52
Chapter 4: Proposed Novel Solution to Solve WPA3 Weaknesses		55
4.1	Introduction	55
4.2	KAKE Handshake	55
4.2.1	Auth-commit Phase	56
4.2.2	Auth-confirm Phase	57
4.3	TOFU Approach	58
4.3.1	Public Key as a Network Name	60
4.3.2	Public Key and SSID as a Network Name	61
4.4	Summary	62

Chapter 5: Discussion and Feasibility Evaluation of the Proposed Solution	65
5.1 Introduction.....	65
5.2 Feasibility Evaluation	65
5.2.1 KAKE Handshake	66
5.2.1.1 What is CRYSTALS-Kyber?.....	66
5.2.1.2 Kyber Types.....	68
5.2.1.3 Kyber Algorithms	69
5.2.1.3.1 Kyber-PKE	69
5.2.1.3.2 Kyber-KEM.....	71
5.2.1.4 Parameters of Kyber	73
5.2.1.5 Kyber-KEM Key Exchange.....	74
5.2.1.6 Kyber Security Features	76
5.2.1.7 Kyber Limitation	77
5.2.1.8 Kyber vs. ECC.....	78
5.2.1.9 Kyber Performance.....	79
5.2.1.9.1 Memory Usage	80
5.2.1.9.2 CPU Usage	81
5.2.1.9.3 Network Traffic Statistics.....	87
5.2.1.10 Kyber in Wi-Fi WPA3	91
5.2.2 KAKE-SAE Hybrid Approach in WPA3.....	92
5.2.2.1 KAKE-SAE Hybrid Design	94
5.2.2.2 Advantages of the Hybrid Approach	95
5.2.2.3 Limitations of the Hybrid Approach	96

5.2.3	TOFU Approach	97
5.2.3.1	How TOFU Works?	97
5.2.3.2	Alternative Approaches	97
5.2.3.3	Why TOFU?	99
5.2.3.4	TOFU Approach Limitations.....	102
5.2.4	Integration of KAKE and TOFU in the Proposed Solution 103	
5.2.4.1	Benefits of Integration.....	104
5.2.4.2	Limitations of Integration.....	104
5.3	Evaluation Results and Findings	105
5.3.1	Technical Feasibility	105
5.3.1.1	Compatibility	105
5.3.1.2	Complexity	105
5.3.1.3	Security	106
5.3.1.4	Performance.....	106
5.3.2	Economic Feasibility	106
5.3.2.1	Cost-Effectiveness	106
5.3.2.2	Affordability	107
5.3.3	Operational Feasibility.....	107
5.3.3.1	Usability.....	107
5.3.3.2	Scalability	107
5.3.3.3	Ease of Implementation.....	107
Chapter 6: Conclusion, and Recommendation and Future Work		110
6.1	Conclusion	110

6.2	Recommendation and Future Work	111
	References.....	112

List of Figures

Figure 3.1: Wi-Fi network architecture	20
Figure 3.2: WEP's challenge-response handshake	23
Figure 3.3: WEP encryption and decryption	24
Figure 3.4: WPA TKIP encryption and decryption.....	29
Figure 3.5: Pairwise keys hierarchy.....	32
Figure 3.6: WPA2-PSK four-way handshake	34
Figure 3.7: WPA2-PSK encryption and decryption.....	36
Figure 3.8: AES-CTR mode.....	36
Figure 3.9: AES-CBC mode	37
Figure 3.10: KRACK against four-way handshake. Retrieved from (Vanhoeef & Piessens, Release the Kraken: new KRACKs in the 802.11 Standard, 2018).....	40
Figure 3.11: WPA3's SAE handshake. Retrieved from (Vanhoeef & Ronen, Dragonblood: A Security Analysis of WPA3's SAE Handshake, 2019)...43	
Figure 4.1: The proposed KAKE handshake.....	58
Figure 4.2: Network's SSID and hashed public key.	61
Figure 4.3: Network's password and public key prompt screen.....	61
Figure 4.4: High level overview of the proposed model	63
Figure 5.1: Kyber.KE key exchange. Retrieved from (Bos, et al., 2018)	75
Figure 5.2: Kyber.UAKE key exchange. Retrieved from (Bos, et al., 2018)	75
Figure 5.3: Kyber.AKE key exchange. Retrieved from (Bos, et al., 2018)	75
Figure 5.4: Memory usage comparison of various post-quantum KEMs. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).....	81

Figure 5.5: CPU usage comparison of various post-quantum KEMs in terms of Key Generation, Encryption, and Decryption. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).....	82
Figure 5.6: Comparison of various KEMs in terms of key sizes. Retrieved from (Tasopoulos, et al., 2022)	84
Figure 5.7: Comparison of various KEMs in terms of execution speed. Retrieved from (Tasopoulos, et al., 2022)	84
Figure 5.8: The computational cost of a protocol for user authentication. Retrieved from (Damir, Meskanen, Ramezani, & Niemi, 2022).....	86
Figure 5.9: Comparison of KEMs based on number of packets sent over the connection. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).....	88
Figure 5.10: Statistic for the packet size sent by each KEM. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).....	89
Figure 5.11: Connection duration in milliseconds for each KEM. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).....	90
Figure 5.12: High-level overview of KAKE-SAE hybrid approach	94
Figure 5.13: KAKE-SAE hybrid handshake	95

List of Tables

Table 2.1: Recent Research Findings on Wi-Fi Security Protocols and Cryptography	13
Table 3.1: Comparison between the four Wi-Fi security protocols.....	51
Table 5.1: Parameter Sets for Kyber KEM.	68
Table 5.2: Parameters values in Kyber	73
Table 5.3: Memory usage comparison of various post-quantum KEMs ..	80
Table 5.4: CPU usage comparison of various post-quantum KEMs in terms of Key Generation, Encryption, and Decryption.....	81
Table 5.5: Comparison of various KEMs in terms of security level, key sizes, and performance metrics	83
Table 5.6: The computational cost of a protocol for user authentication .	85
Table 5.7: Comparison of KEMs based on number of packets sent over the connection	87
Table 5.8: Statistic for the packet size sent by each KEM.	88
Table 5.9: Connection duration in milliseconds for each KEM.	89
Table 5.10: Comparison of trust establishment approaches	100

Acknowledgment

I would like to express my deepest gratitude to my supervisor, Dr. Alauddin Alomary, for his guidance, support, and encouragement throughout my thesis journey. His expertise, patience, and willingness to share his knowledge have been invaluable to me. I feel truly fortunate to have had his guidance, and I am grateful for his unwavering commitment to my academic success.

Dedication

To my parents for the inspiration, they have given me. To my wife for the support, she has provided to me...

Publications

This thesis is based on previously papers written in collaboration with my supervisor, Dr. Alauddin Alomary and Dr. Wael Elmedany:

JOURNAL PAPERS

1. (Not published) Alhamry, Mohamed, and Wael Elmedany. “Integrating Public Key Infrastructure with Internet of Things – Challenges and Solutions: A Review Paper.”

CONFERENCE PAPERS

1. (Published) Alhamry, Mohamed, and Alauddin Alomary. “Exploring Wi-Fi WPA2-PSK protocol weaknesses.” In 2022 International Conference on Data Analytics for Business and Industry (ICDABI), pp. 190-195. IEEE, 2022.
URL: <https://ieeexplore.ieee.org/abstract/document/10041465>
2. (Published) Alhamry, Mohamed, and Wael Elmedany. “Exploring Wi-Fi WPA2 KRACK Vulnerability: A Review Paper.” In 2022 International Conference on Data Analytics for Business and Industry (ICDABI), pp. 766-772. IEEE, 2022.
URL: <https://ieeexplore.ieee.org/abstract/document/10041548>

List of Abbreviations

AES	Advanced Encryption Standard
AI	Artificial Intelligence
AKA	Authentication and Key Agreement
AKE	Authenticated Key Exchange
AP	Access Point
CA	Certificate Authority
CBC–MAC	Cipher Block Chaining Message Authentication Code
CCA	Chosen Ciphertext Attack
CIRCL	Cloudflare Interoperable Reusable Cryptographic Library
CPA	Chosen Plaintext Attack
CRC-32	Cyclic Redundancy Check-32
CRYSTALS-Kyber	Cryptographic suite for algebraic lattices Kyber
CTR	Counter Mode
DoS	Denial-of-Service
EAP	Extensible Authentication Protocol
ECC	Elliptic Curve Cryptography
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECIES	Elliptic Curve Integrated Encryption Scheme
FFDHE	Finite Field Diffie-Hellman Ephemeral
FPGA	Field Programmable Gate Array
GTK	Group Temporal Key
HLS	High-Level Synthesis
HTTPS	Hypertext Transfer Protocol Secure
IC	Integrity Check
IEEE	Institute of Electrical and Electronics Engineers
IV	Initialization Vector

IoT	Internet of Things
KAKE	Kyber-based Authenticated Key Exchange
KCK	Key Confirmation Key
KDF	Key Derivation Function
KE	Key Exchange
KEK	Key Encryption Key
KEM	Key Encapsulation Mechanism
KEM	Key Encapsulation Mechanism
KRACK	Key Reinstallation AttaCK
LLC	Logical Link Control
LWE	Learning-With-Errors
MAC	Media Access Control
MFA	Multi-Factor Authentication
MIC	Message Integrity Code
MitM	Man-in-the-Middle
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
OpenPGP	Open Pretty Good Privacy
PAKE	Password-Authenticated Key Exchange
PE	Password Element
PKE	Public-Key Encryption
PKI	Public Key Infrastructure
PMF	Protected Management Frames
PMK	Pairwise Master Key
PMK-ID	Pairwise Master Key Identifier
PSK	Pre-Shared Key
PTK	Pairwise Transient Key

QSA	Quantum-Safe encryption Algorithm
RADIUS	Remote Authentication Dial-In User Service
RC4	Rivest Cipher 4
RSNE	Robust Security Network Element
SAE	Simultaneous Authentication of Equals
SSH	Secure Shell
SSID	Service Set Identifier
SUPI	Subscription Permanent Identifier
TK	Temporal Key
TKIP	Temporal Key Integrity Protocol
TOFU	Trust On First Use
WBAN	Wireless Body Area Network
WEP	Wired Equivalent Privacy
WLANs	Wireless Local Area Networks
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access II
WPA3	Wi-Fi Protected Access III

Chapter 1: Introduction

Chapter 1: Introduction

1.1 Background and Motivation

Wireless Local Area Networks (WLANs) have revolutionized the way we connect to the internet. WLANs, commonly known as Wi-Fi networks, have become an integral part of our daily lives, providing a means for billions of devices to connect to the internet. However, the security of Wi-Fi networks has always been a major concern due to the vulnerabilities in the security protocols. Attackers can exploit these vulnerabilities to gain access to sensitive information, such as passwords, credit card information, and personal data, compromising the privacy and security of users. Therefore, it is crucial to develop and implement effective security mechanisms to protect Wi-Fi networks from such attacks.

The current Wi-Fi security standard, Wi-Fi Protected Access III (WPA3), was introduced in 2018 to address the security weaknesses in its predecessor, Wi-Fi Protected Access II (WPA2). However, researchers have identified several vulnerabilities in the WPA3 protocol, including the Dragonblood vulnerabilities and Evil Twin attacks, which can compromise the security of the network and the user's data. Moreover, the utilization of public key cryptography in WPA3 is open to quantum attacks that might eventually become feasible in the years to come. Therefore, there is a pressing need to develop a solution that can address the issues in the WPA3 protocol and provide a more secure Wi-Fi network for users.

1.2 Research Problem

Wi-Fi networks have become ubiquitous in our daily lives, with billions of devices connected to them. The security of these networks is of utmost importance, yet the latest WPA3 protocol has been found to have several weaknesses that can be exploited by attackers. These include side-channel

attacks, Denial-of-Service (DoS) attacks, Downgrade attacks, Evil Twin attacks, and others. Moreover, the public key cryptography used in WPA3 is vulnerable to quantum attacks that may become feasible in the near future. Therefore, developing more robust and secure Wi-Fi security protocols is crucial to address these challenges and ensure the privacy and confidentiality of user data on Wi-Fi networks.

1.3 Research Objective

The objective of this research is to propose a novel solution that can improve the security of Wi-Fi networks by addressing the vulnerabilities in the WPA3 protocol. This solution should provide post-quantum protection against attacks and be easy to implement and use.

1.4 Research Methodology

1.4.1 Research Design

To achieve the objective of proposing a novel solution to enhance the security of Wi-Fi networks, we utilized a systematic approach to conduct our research. This involved identifying the weaknesses in the WPA3 protocol through an extensive literature review and designing a novel solution that addresses these vulnerabilities.

The methodology employed to develop the proposed solution involved several key steps. Firstly, a comprehensive literature review was conducted to identify the vulnerabilities present in the WPA3 protocol. This review served as a foundation for understanding the existing weaknesses and determining the areas that required improvement. Next, methods to address these vulnerabilities were identified. The selection process involved choosing the Kyber algorithm as a replacement for the public key algorithm currently used in WPA3, which has several weaknesses. Kyber was selected for its ability to address the vulnerabilities existing in WPA3 and

provide both classical and post-quantum computer security. Moreover, Kyber was chosen based on its excellent performance as one of the leading post-quantum cryptography algorithms. Additionally, the Trust On First Use (TOFU) method was chosen to establish trust between the client and the Access Point (AP). TOFU was preferred for its simplicity and the fact that it does not necessitate additional infrastructure, hardware, or software.

To assess the feasibility of implementing Kyber and TOFU in a Wi-Fi network, an in-depth literature review was conducted specifically focusing on their application in similar contexts. This evaluation helped determine the suitability of these approaches for enhancing WPA3 security.

Based on the findings and evaluations, a proposed solution was designed, consisting of two main components: the Kyber-based Authenticated Key Exchange (KAKE) handshake and the TOFU approach. The KAKE handshake was incorporated to address the vulnerabilities of WPA3, providing improved security and post-quantum protection. Meanwhile, the TOFU approach was utilized to establish trust between the client and AP, mitigating potential security risks.

Through this methodology, the proposed solution was developed, utilizing the KAKE handshake and TOFU approach to enhance the security of WPA3. This systematic approach ensured that the solution was based on a thorough understanding of the vulnerabilities, careful selection of appropriate methods, and an evaluation of their feasibility in the context of Wi-Fi networks.

1.4.2 Data Collection and Analysis

In order to evaluate the effectiveness of our proposed solution, we used a proof-of-concept approach and drew on existing literature results. We collected data on the performance, security, and usability of our proposed

solution by analyzing and comparing the results from the proof of concept with those of other resources.

1.4.3 Limitations

There were some limitations to our research, including the reliance on a proof-of-concept approach and the use of existing literature results to evaluate the proposed solution. As such, the results may not be directly applicable to real-world scenarios. However, we believe that our findings provide valuable insights into the effectiveness of our proposed solution and can guide future research in this area.

1.5 Contribution to the Field

This research makes a significant contribution to the field by proposing a novel solution that can improve the security of Wi-Fi networks by addressing the vulnerabilities in the WPA3 protocol. The proposed solution integrates two approaches, the KAKE handshake and the TOFU approach, to provide classical and post-quantum protection and additional security against Downgrade attacks and Evil Twin attacks.

The KAKE handshake is a secure method for two devices to authenticate each other and exchange a session key. It provides robust security against various types of classical attacks, including Man-in-the-Middle (MitM) attacks and replay attacks, and offers post-quantum protection against attacks from quantum computers. Moreover, KAKE is able to address vulnerabilities in both WPA2 and WPA3 protocols, making it a versatile solution for securing Wi-Fi networks. Another notable security feature of KAKE is that it provides forward secrecy, which ensures that even if an attacker gains access to the session key, network's password, or even network's private key, they cannot use it to decrypt past or future communications.

The TOFU approach addresses the vulnerabilities in the WPA3 protocol by providing additional security against Evil Twin attacks and Downgrade attacks. It uses a trust model in which a device trusts the authenticity of a peer based on its public key. Once a device has established trust with a peer, it will only connect to that peer with that specific public key in the future, providing additional protection against Downgrade attacks and Evil Twin attacks.

1.6 Thesis Outline

The thesis is structured as follows: Chapter 2 provides a review of related works on weaknesses of Wi-Fi security as well as Kyber and TOFU applications. Chapter 3 presents an overview of Wi-Fi technology, including the architecture, protocols, and security mechanisms. Chapter 4 presents our proposed solution, which utilizes the KAKE handshake and TOFU approach to address the known vulnerabilities in the Wi-Fi security protocols. In Chapter 5, we discuss the security, performance, and feasibility of our proposed solution using a proof-of-concept approach and results from other resources. Finally, Chapter 6 summarizes our findings, provides recommendations for future research, and concludes the thesis.

Chapter 2: Related Work

Chapter 2: Related Work

2.1 Introduction

In this chapter, we provide a comprehensive review of the related works that have informed our proposed solution to enhance the security of Wi-Fi networks. We discuss the weaknesses of the WPA3 protocol, the state-of-the-art in the field of CRYSTALS-Kyber application, particularly in wireless networks and limited constrained devices, and the feasibility of implementing TOFU approach in different applications including WPA3 networks.

2.2 WPA3 Weaknesses

The security of the WPA3 protocol has been subject to scrutiny since its introduction in 2018. Several vulnerabilities have been identified by researchers, which exploit weaknesses in the Dragonfly, or Simultaneous Authentication of Equals (SAE) handshake, authentication method and other components of the WPA3 protocol. In this section, we provide a comprehensive overview of the known weaknesses in the WPA3 protocol. Specifically, we discuss the findings of recent studies that highlight the vulnerabilities in the SAE authentication method, which is a key component of the WPA3-Personal security standard. These studies demonstrate authentication bypasses and side-channel attacks that can be directed against SAE's password encoding mechanism. Furthermore, we examine the susceptibility of the WPA3 protocol to rogue APs, such as Evil Twin attacks, and DoS attacks.

Mathy Vanhoef and Eyal Ronen (Vanhoef & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020) conduct an in-depth analysis of the security provided by Dragonfly (SAE handshake). Initially, they do implementation audits and demonstrate

timing leaks as well as authentication bypasses in WPA3 networks. After that, they investigate SAE's architecture and discuss DoS and downgrade threats. Their subsequent and most important outcomes are side-channel attacks directed against SAE's password encoding mechanism such as hash-to-curve method. They had the impression that the SAE implementation itself is responsible for this side channel leakage. In addition to this, they investigate how complicated it would be to brute-force the password using the information that was disclosed. Finally, they discuss vulnerabilities in countermeasures that are compatible with previous versions, specifically the WPA3 transition mode.

The (Kohlios & Hayajneh, 2018) demonstrate that the WPA3 protocol does not prevent an attacker from utilizing a rogue AP to trick a user into exposing the passphrase of a legitimate AP. Following a successful attack on a rogue AP, the intruder will have obtained the network password and will have the option to either legitimately connect to the network or launch an attack via an Evil Twin.

This paper (Chatzoglou, Georgios, & Constantinos, 2022) discussed the SAE password-authenticated key agreement method, which is used in the WPA3-Personal Wi-Fi security standard. While SAE is designed to be highly resistant to offline dictionary attacks, it remains vulnerable to a type of attack called "clogging" at the AP side. To address this issue, SAE includes an anti-clogging scheme. The paper describes a study that tested the resilience of SAE to DoS attacks, and found a significant number of novel DoS attacks affecting virtually any AP. The attacks ranged from temporary loss of connectivity to outright disconnection. The study highlights the need for improved implementation of the protocol.

This article (Sun, 2022) discusses SAE handshake used in Wi-Fi devices for peer-to-peer authentication. The article points out a vulnerability in the

randomness generation algorithm used in the protocol that can be exploited for impersonation attacks. The article recommends protocol amendments to improve security and encourages SAE implementations to be upgraded to protect against these attacks. SAE is considered a major authentication mechanism in Wi-Fi and is widely implemented in modern devices.

2.3 CRYSTALS-Kyber

Post-quantum cryptography has gained significant attention in recent years as a potential solution to the threat posed by quantum computers. Among the various post-quantum cryptography schemes, CRYSTALS-Kyber has emerged as a promising candidate due to its security and efficiency. The papers in this section explore the implementation and application of CRYSTALS-Kyber, ranging from optimization for mobile devices to hardware implementation on Field Programmable Gate Array (FPGA), and its use in protecting sensitive medical information and improving the security of authentication and key agreement protocols.

This paper (Zhao, Zhang, Huang, Liu, & Hancke, 2021) explores how to make Kyber, a secure Key Encapsulation Mechanism (KEM), work faster on mobile phones by optimizing it for high-performance processors from the ARM Cortex-A series. The authors used a special machine instruction called NEON to parallelize Kyber's core modules and developed optimized implementations based on NEON's characteristics. They also proposed a new strategy for Kyber implementation, which resulted in significant speed improvements compared to other implementations.

The (Da Costa, Camponogara, López, & Moisés, 2022) examines if a new post-quantum Public-Key Encryption system, CRYSTALS-Kyber, can be implemented in hardware-constrained Smart Meters. The authors tested the scheme on a System-on-a-Chip FPGA device and found that the proposed

hardware-software co-design implementation of the CRYSTALS-Kyber scheme reduces its execution time by around 70% compared to its software implementation, making it feasible for use in Smart Meters.

This paper (Xing & Shuguo, 2021) discusses the need for post-quantum cryptosystems to ensure information security in the future. It presents a hardware design of CRYSTALS-Kyber, a post-quantum key exchange mechanism protocol, implemented on a FPGA platform. The design achieved decent performance with limited hardware resources and is demonstrated in detail. The proposed design is more efficient than the corresponding High-Level Synthesis (HLS) based designs or hardware-software co-designs. The implementation computes key-generation, encapsulation and decapsulation phase and is adaptive Chosen Ciphertext Attack (CCA) secure.

The paper (LAGUEAGUE & CHEIKHAOUI, 2022) describes an authentication scheme for protecting patients' medical information in the Wireless Body Area Network (WBAN), which is a wireless sensor technology that constantly exchanges health information in real time. The scheme is based on post-quantum cryptography methods and is designed to be efficient and lightweight. The scheme uses an asymmetric encryption algorithm to encrypt or decrypt the secret data, and the key used for encryption is established using the Kyber with secure- Chosen Plaintext Attack (secure-CPA) level encryption scheme. The goal of the scheme is to provide a clearly defined defense line to protect patient information from unauthorized or malevolent opponents in the WBAN system.

This paper (Damir, Meskanen, Ramezani, & Niemi, 2022) addresses security and privacy vulnerabilities in the standardized 5G Authentication and Key Agreement protocol (5G AKA) by proposing a new protocol that is compatible with it. The new protocol utilizes KEMs for user identity

protection and perfect forward secrecy, making it adaptable to the quantum-safe setting. The post-quantum KEM CRYSTALS-Kyber and National Institute of Standards and Technology (NIST) post-quantum Round 4 candidate KEMs were used to analyze the performance of the proposed protocol. The results show that the protocol is feasible in practice and sometimes outperforms the public-key cryptography used in 5G AKA, i.e., Elliptic Curve Integrated Encryption Scheme (ECIES) which is based on Elliptic Curve Cryptography (ECC).

2.4 TOFU

The paper (Walfield & Koch, 2016) presents a design and implementation of a TOFU policy for Open Pretty Good Privacy (OpenPGP). This policy checks that the signer used the same key as in the past when verifying a signature. If not, it is a strong indicator of forgery or an active MitM attack. TOFU can proactively detect new attacks and reactively detect an attack if the singer gets a message through. The paper outlines how TOFU can be integrated into OpenPGP, addresses potential attacks against TOFU, and demonstrates the practicality of the approach. The paper concludes that TOFU's practical security is stronger than OpenPGP's current trust policy and better than the most popular alternative, an X.509-based Public Key Infrastructure (PKI).

The paper (Ginkel, Poll, & Wiggers, 2019) discusses the level of security provided by WPA3 and SAE and Opportunistic Wireless Encryption in Wi-Fi networks. The paper concludes that while these technologies significantly improve security, they still have vulnerabilities, especially against the Evil Twin attack. To counter this attack, several countermeasures exist, such as the use of PKI and TOFU. The paper suggests that the use of certificates is not a feasible solution for wireless networking, unlike web browsing. It also argues that the TOFU approach

is the best option regarding the four criteria they used: usability for the client, feasibility to implement the approach, overhead, and effectiveness.

The paper (Vanhoeef & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020) discusses mitigation techniques for downgrade attacks on WPA3-SAE, a security protocol for Wi-Fi networks. The authors suggest that a client should remember if a network supports WPA3-SAE and should never connect to the network using a weaker handshake once it has successfully connected using SAE. This is similar to the TOFU approach used by Secure Shell (SSH) and the Strict-Transport-Security header of Hypertext Transfer Protocol Secure (HTTPS).

Table 2.1 summarizes different researchers' work related to Wi-Fi security and Cryptography.

Table 2.1: Recent Research Findings on Wi-Fi Security Protocols and Cryptography

Topic	Reference	Key Findings
WPA3 Weaknesses	(Vanhoeef & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020)	Timing leaks and authentication bypasses, DoS and downgrade threats, side-channel attacks against SAE's password encoding mechanism.
	(Kohlilos & Hayajneh, 2018)	WPA3 protocol not preventing an attacker from using a rogue AP to trick a user into exposing the passphrase of a legitimate AP.

	(Chatzoglou, Georgios, & Constantinos, 2022)	SAE's vulnerability to DoS attacks and the need for improved implementation of the protocol.
	(Sun, 2022)	Vulnerability in the randomness generation algorithm used in SAE, which can be exploited for impersonation attacks.
CRYSTALS-Kyber Applications	(Zhao, Zhang, Huang, Liu, & Hancke, 2021)	Optimizing Kyber for high-performance processors from the ARM Cortex-A series using NEON to parallelize Kyber's core modules and developing a new strategy for Kyber implementation, resulting in significant speed improvements compared to other implementations.
	(Da Costa, Camponogara, López, & Moisés, 2022)	Feasibility of implementing the CRYSTALS-Kyber scheme in hardware-constrained Smart Meters by testing it on a System-on-a-Chip FPGA device, resulting in a 70% reduction in execution time compared to software implementation.
	(Xing & Shuguo, 2021)	A hardware design of CRYSTALS-Kyber implemented on a FPGA platform that is more efficient than the corresponding HLS based designs or hardware-software co-designs, is adaptive CCA secure, and computes key-generation, encapsulation, and decapsulation phases.
	(LAGUEAGUE &	An authentication scheme for protecting patients' medical

	CHEIKHAOUI, 2022)	information in the WBAN based on post-quantum cryptography methods using Kyber with secure-CPA level encryption scheme.
	(Damir, Meskanen, Ramezani, & Niemi, 2022)	A new protocol utilizing KEMs for user identity protection and perfect forward secrecy, adaptable to the quantum-safe setting, and using the post-quantum KEM CRYSTALS-Kyber and NIST post-quantum digital signature scheme.
TOFU Applications	(Walfield & Koch, 2016)	TOFU is a practical and effective security enhancement for OpenPGP, better than the current trust policy and X.509-based PKI
	(Ginkel, Poll, & Wiggers, 2019)	The TOFU approach is the best option for wireless Wi-Fi WPA3 network security based on usability, feasibility, overhead, and effectiveness
	(Vanhoeve & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020)	Suggests a client should remember if a network supports WPA3-SAE and should never connect to the network using a weaker handshake once it has successfully connected using SAE. Similar to the TOFU approach used by SSH and the Strict-Transport-Security header of HTTPS

2.5 Summary

This chapter provides a comprehensive review of related works that have informed the proposed solution to enhance the security of Wi-Fi networks. The "WPA3 Weaknesses" section describes the various types of attacks that can exploit the vulnerabilities in the WPA3 protocol, which is the latest security protocol for Wi-Fi networks. This section includes several research studies that demonstrate authentication bypasses, side-channel attacks, rogue APs, and DoS attacks on WPA3 networks.

The "CRYSTALS-Kyber" section highlights the state-of-the-art in the field of CRYSTALS-Kyber application, particularly in wireless networks and limited constrained devices. This section includes several research studies that explore the implementation and application of Kyber, ranging from optimization for mobile devices to hardware implementation on FPGA, and its use in protecting sensitive medical information and improving the security of authentication and key agreement protocols.

In the "TOFU" section, the feasibility of applying the TOFU approach in various applications is discussed. The section also includes some suggested solutions based on TOFU that aim to alleviate Evil Twin and Downgrade attacks on WPA3 networks.

This chapter lays the foundation for the proposed solution in Chapter 4 by providing an overview of the weaknesses in the WPA3 protocol and highlighting the latest developments in Kyber application and the feasibility of TOFU approach.

Chapter 3: Wi-Fi Overview

Chapter 3: Wi-Fi Overview

3.1 Introduction

Wireless networks have become an essential part of modern communication systems, enabling various applications such as home and business networks, mobile devices, and the Internet of Things (IoT). However, the widespread adoption of Wi-Fi technology has also brought to light the various security challenges and vulnerabilities that threaten the confidentiality, integrity, and availability of wireless communications. The increasing number of cyber-attacks targeting Wi-Fi networks highlights the need for more secure and robust security mechanisms to protect these networks.

To address these security concerns, several security protocols have been developed to secure Wi-Fi networks, such as Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and WPA2. WEP, the first Wi-Fi security protocol, was found to be easily exploitable due to its weak encryption and key management. As a result, WPA and WPA2 were introduced, providing stronger encryption and better key management. However, despite these improvements, both WPA and WPA2 have known vulnerabilities that can be exploited by attackers.

To further enhance Wi-Fi security, the Wi-Fi Alliance introduced the WPA3 protocol, which was designed to address the known vulnerabilities in WPA and WPA2. WPA3 introduces several new features, including stronger encryption, more robust key management, and enhanced protection against brute-force attacks. These new features make it more difficult for attackers to compromise Wi-Fi networks and reduce the risk of successful attacks. WPA3, while being a significant improvement over its predecessor, is not without its vulnerabilities. One such vulnerability is the Dragonblood

attacks, which exploit vulnerabilities in the Wi-Fi Alliance's Dragonfly key exchange protocol, SAE handshake, used in WPA3 to recover the password of the Wi-Fi network. Another vulnerability is the timing-based side-channel attack, which can be used to recover the Pairwise Master Key Identifier (PMK-ID) of a WPA3-protected Wi-Fi network. Additionally, WPA3 is vulnerable to downgrade attacks, similar to those that affect WPA2 (Vanhoeft & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020).

This chapter provides an overview of Wi-Fi technology, including the network topology and security protocols. It discusses the strengths and weaknesses of existing security protocols, including WEP, WPA, WPA2, and WPA3. Understanding the weaknesses of existing protocols is essential for developing more robust and secure security mechanisms for Wi-Fi networks.

3.2 Wi-Fi Architecture

The Wi-Fi architecture consists of various components that work together to provide wireless connectivity. These components include wireless APs, wireless clients, and the wireless medium itself (O'hara & Petrick, 2005).

Wireless APs are the central components of a Wi-Fi network. They are responsible for transmitting and receiving data packets between wireless clients and the wired network. An AP is typically connected to a wired network, such as an Ethernet network, and it broadcasts a wireless signal that can be used by wireless clients to connect to the network (Figure 3.1).

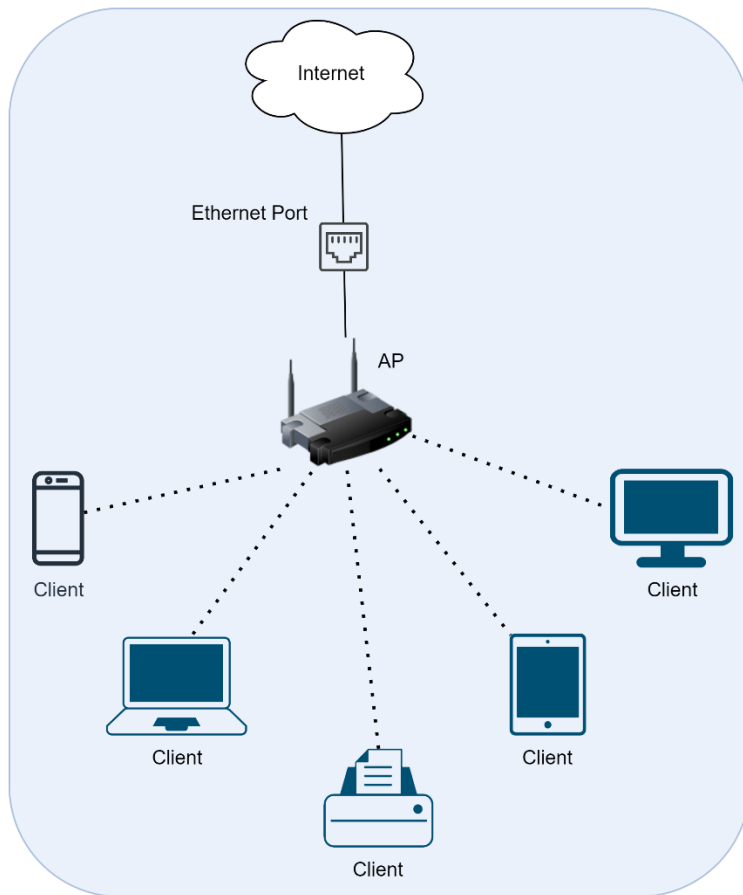


Figure 3.1: Wi-Fi network architecture

Wireless clients are devices that use the Wi-Fi medium to connect to a Wi-Fi network. These devices can be laptops, smartphones, tablets, or any other device that supports Wi-Fi connectivity. Wireless clients communicate with the AP using radio waves, which are transmitted over the air.

The wireless medium is the physical layer of the Wi-Fi architecture. It provides the means for wireless communication between the AP and wireless clients. The wireless medium is based on the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standard, which defines the physical and data link layers of the Wi-Fi architecture (Gast, 2005).

The IEEE 802.11 standard defines various physical layer specifications, including the frequency band, modulation scheme, and data rate. The

standard also defines various data link layer protocols, such as the Media Access Control (MAC) protocol and the Logical Link Control (LLC) protocol.

Wi-Fi networks can be classified into several topologies, depending on how the network is structured and how devices connect to each other. The most common Wi-Fi topologies are (Anastasi, Conti, & Gregori, 2004):

1. Infrastructure mode: In infrastructure mode, devices connect to a central AP that acts as a bridge between the wireless network and the wired network. The AP is responsible for managing the wireless network and providing connectivity to devices. Infrastructure mode is commonly used in home and business networks, as well as public hotspots.
2. Ad-hoc mode: In ad-hoc mode, devices connect directly to each other without the need for a central AP. This mode is often used for peer-to-peer communication, such as sharing files or playing multiplayer games.

Each topology has its own advantages and disadvantages, and the choice of topology depends on the specific use case and network requirements. In the rest of this thesis, we will focus on the infrastructure mode, which is the most common Wi-Fi topology. Nevertheless, the majority of the findings are applicable to ad-hoc networks as well.

Understanding the Wi-Fi architecture is essential for understanding the security mechanisms that are used to protect Wi-Fi networks. In the next section, we will discuss the various security protocols that have been developed to secure Wi-Fi networks.

3.3 Wi-Fi Security Protocols

The following is a historical sequence of changes that occurred during the development of the security protocols for secured communications on 802.11 wireless networks:

1. WEP
2. WPA
3. WPA2
4. WPA3

3.3.1 WEP

WEP was the first 802.11 security protocol introduced in 1997 for Wi-Fi networks. It was designed to provide the same level of reliable security to wireless networks as that of conventional wired networks. However, it was soon found to be vulnerable to attacks due to its weak encryption and key management.

3.3.1.1 Authentication

Open System (no authentication) and Shared Key authentication are the two types of authentications that WEP defines (Housley & Arbaugh, 2003). They are employed in the AP-client authentication process. Because the client must demonstrate that it is familiar with the shared WEP key, authentication was seen to be preferable to no authentication. Contrary to popular belief, turning on authentication actually lowers network security overall and makes it simpler to guess WEP key (Lashkari, Towhidi, & Hosseini, 2009).

In the case of the Shared Key authentication method, the WEP Pre-Shared Key (PSK) is the one that is utilized for authentication and association in a challenge-response handshake that consists of four steps (Al Naamany, Al Shidhani, & Bourdouce, 2006):

1. The client initiates communication with the AP by sending an authentication request.
2. In response, the AP sends a challenge in the form of clear text.
3. The client encrypts the challenge-text by using the PSK that has been specified, and then transmits it back as part of an association request.
4. The client's encrypted challenge will be decrypted by the AP. If the information provided here is a match for the challenge text, the AP will transmit a successful reply (Figure 3.2).

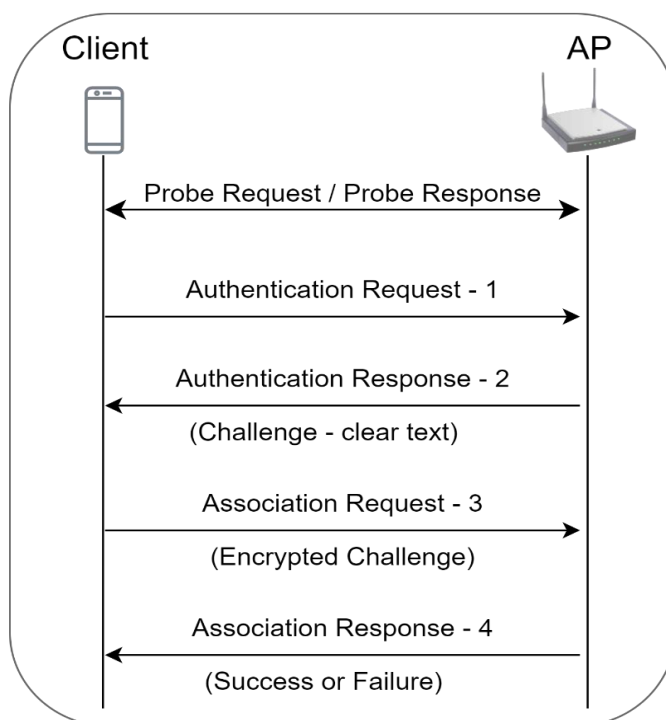


Figure 3.2: WEP's challenge-response handshake

If the association success, the WEP PSK is then put to use in order to encrypt the data frames through the use of the Rivest Cipher 4 (RC4) algorithm.

The weakness in WEP's authentication process is due to the shared key mechanism, which makes it vulnerable to dictionary attacks and brute-force attacks.

3.3.1.2 Encryption

WEP uses the RC4 technique for encryption. The size of the PSK can be either 40 bits or 104 bits. A 24-bit Initialization Vector (IV) is extended to the PSK using RC4 algorithm to produce a keystream with a size of 64 bits or 128 bits (depending on the size of the PSK used). The client chooses the IV, which should be updated so every packet is encrypted with a different keystream. Then, the resulting keystream is XORed with the plaintext to produce the ciphertext. Finally, the IV is sent in clear text over an open channel together with the ciphertext. For the purpose of maintaining data integrity, WEP employs the Cyclic Redundancy Check-32 (CRC-32) algorithm (Lashkari, Towhidi, & Hosseini, 2009). (See Figure 3.3 for WEP encryption and decryption).

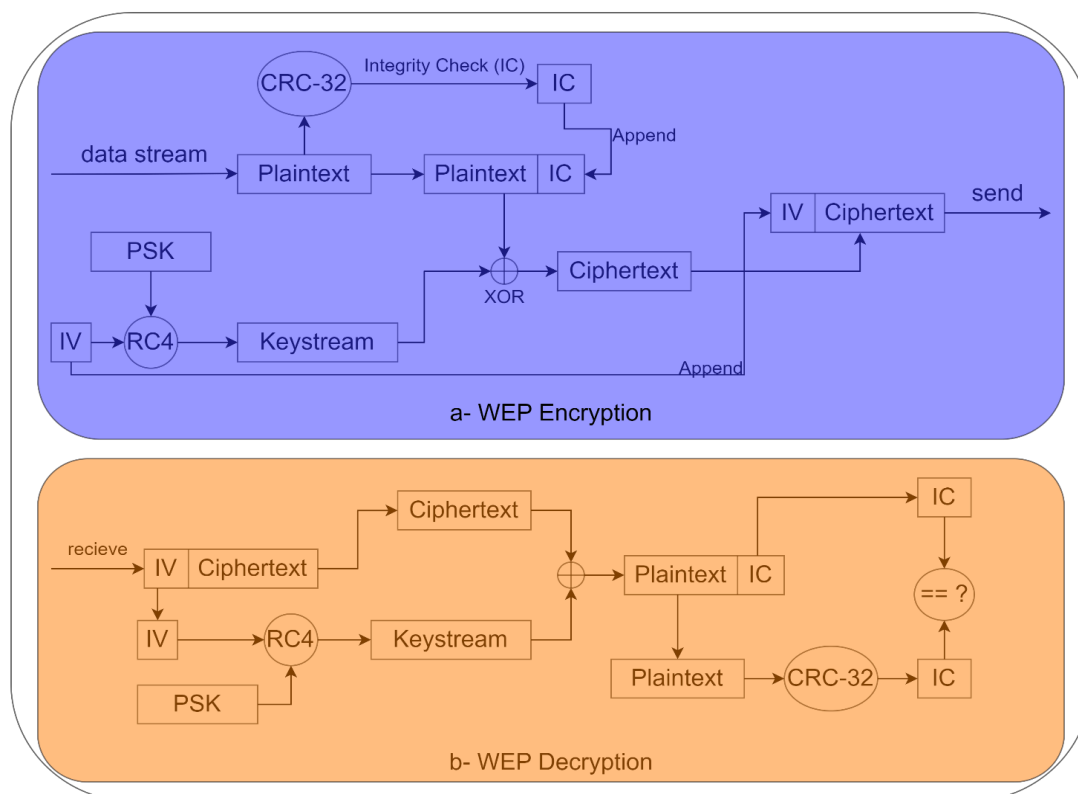


Figure 3.3: WEP encryption and decryption

3.3.1.3 Weaknesses

Obviously, the WEP does not provide the same degree of security as wired connections. WEP suffers from a wide variety of weaknesses, each of which significantly lowers the protocol's promised level of security (Ye & Yue, 2010).

- **WEP Authentication Weakness:** PSK authentication requires encrypting a challenge to prove WEP PSK knowledge. The issue is that an eavesdropping attacker can see both the challenge and the encrypted challenge. From those, the attacker can deduce the RC4 keystream used to encrypt the challenge and use it to encrypt future challenges. Moreover, a malicious client can perform a dictionary attack on the wireless network PSK using PSK authentication.
- **WEP Key Management Issue:** One of the disadvantages of the WEP standard is that key management is not defined in it. This is because keys without standardized key management have a tendency to last a long time and be of bad quality. When using PSK, any client that has access to the key material must be completely trusted to use it lawfully. This degree of confidence is unattainable. Anyone on the network can decode traffic sent for any other connected device if everyone on that network uses the same key.
- **WEP Encryption Weaknesses:** Because it was permissible for export under the majority of national encryption standards, the IEEE chose 40-bit encryption for WEP. Sadly, on present cheap PCs, keys for 40-bit RC4 encryption can be discovered via exhaustive search (brute force). A modern PC can quickly and efficiently scan through all possible combinations for the 40-bit WEP PSK secret key. Other than 40 bits, the 802.11 standard does not mention any alternative WEP key sizes. However, with strong compatibility, the majority of

providers have implemented an unofficial standard by merely increasing the key size to 104 bits. Because it sounds nicer than a 104-bit key, this is frequently referred to as a "128-bit" WEP key, but it is not an accurate comparison. The RC4 encryption key has a 24-bit IV in both scenarios (40-bits or 104-bits). 104-bit keys are undoubtedly more secure against brute-force cyberattacks than 40-bit keys. Moreover, with any WEP PSK size (40-bit or 104-bit), the 24-bit WEP IV allows for $2^{24} = 16,777,216$ possible RC4 keystreams. Keep in mind that the IV is transmitted unencrypted with each packet. After sending few packets, the chance of reusing IV is increased. If the RC4 keystream for a specific IV is determined, an attacker can either decrypt following packets that were encrypted using the same IV or fabricate packets. This implies that if you know what the keystream utilized to encrypt a packet, you can decode it without knowing the WEP PSK. Furthermore, the WEP Integrity Check (IC) Value uses CRC-32 for data integrity. Although it is a great checksum for finding errors, CRC-32 is a terrible option for cryptographic hash. Because the CRC-32 IC is a linear function of the message, an attacker can easily alter an encrypted message and fix the IC to make the message appear valid. Several straightforward attacks are possible when encrypted packets are modifiable. Because IV- and IC-based cyberattacks are not reliant on the size of the key, even extremely large keys all appear to be similar to the attacker. This is the most significant drawback associated with these types of attacks.

Key management issues, implementation flaws, and general encryption scheme vulnerabilities have plagued WEP. Although WEP may make things harder for attackers, it offers no actual protection from a motivated

attacker. As a result of these weaknesses, WEP has been deprecated and is no longer recommended for use in Wi-Fi networks.

3.3.2 WPA

The 802.11i (or WPA2) is the second generation of security protocols, and it was created in 2004 to enable more secure communication. Nevertheless, as a short-term solution, an intermediate solution called WPA was established by The IEEE 802.11 TGi in 2003 (Srinivasan & Michell, 2004).

3.3.2.1 Authentication

Users are required to enter a password during the authentication process while using WPA, which assists in ensuring that the Wi-Fi networks are secure. In addition to that, WPA is also compatible with authentication servers such as Remote Authentication Dial-In User Service (RADIUS) server (Khasawneh, Kajman, Alkhudaidy, & Althubyani, 2014).

The WPA protocol can be implemented using either the WPA-PSK mode or the WPA enterprise mode:

1. WPA-Enterprise: This mode was specifically built for networks, in enterprises, that are large in size. In this mode, an authentication server such as RADIUS server is utilized. When connecting to the network using a RADIUS server, users are required to make use of their own unique identities (a username and a password). In the event that a client is compromised, administrators have the ability to revoke access to that client independently of any other clients.
2. WPA-PSK: Individual users and small to medium-sized businesses face significant financial burdens and administrative challenges when trying to run a specialized authentication server. The WPA-PSK mode, which is also known as the WPA personal mode, offers

an authentication technique that is simple and more straightforward. In this configuration, an authentication server is not necessary at all. Users need to simply configure a PSK on each AP. If a client's PSK and the key specified on the WLAN's AP are identical, the client is allowed to access the WLAN. As the WPA's PSK is not utilized for the purpose of encryption, it does not provide any potential security problems in the same way that the WEP shared key authentication does.

In order to use WPA's PSK authentication, both the client and the AP need to be set with the same PSK. By the process of key negotiation, the client and the AP authenticate each other. When it comes time to negotiate a key, both the client and the AP will use their PSKs to decrypt the message that was sent by the other party. If the communications can be decrypted successfully, both the client and the AP have the identical PSK; if they use different PSKs, PSK authentication is unsuccessful. The process of key negotiation (including the four-way handshake for unicast communication and the two-way handshake for broadcast/multicast communication) will be discussed later under "WPA2" section.

3.3.2.2 Encryption

Under the context of this approach, the Temporal Key Integrity Protocol (TKIP) has been developed as a fix for WEP flaws. TKIP makes use of the RC4 algorithm as well, but with some significant additions and alterations. Throughout the course of the transmission, the keys will vary dynamically, and a considerably larger IV (48 bits) is utilized.

In order to play in a variety of keys throughout each session, a key mixing mechanism is employed. Moreover, a brand-new algorithm known as Michael is utilized as the Message Integrity Code (MIC) with a 64-bit key

in order to guarantee that the data will remain unchanged while it is being transmitted. In a nutshell, TKIP offers 128-bit per-packet key mixing, MIC, and a mechanism for re-keying (Khasawneh, Kajman, Alkhudaiby, & Althubyani, 2014). Figure 3.4 illustrates encryption and decryption in the WPA protocol.

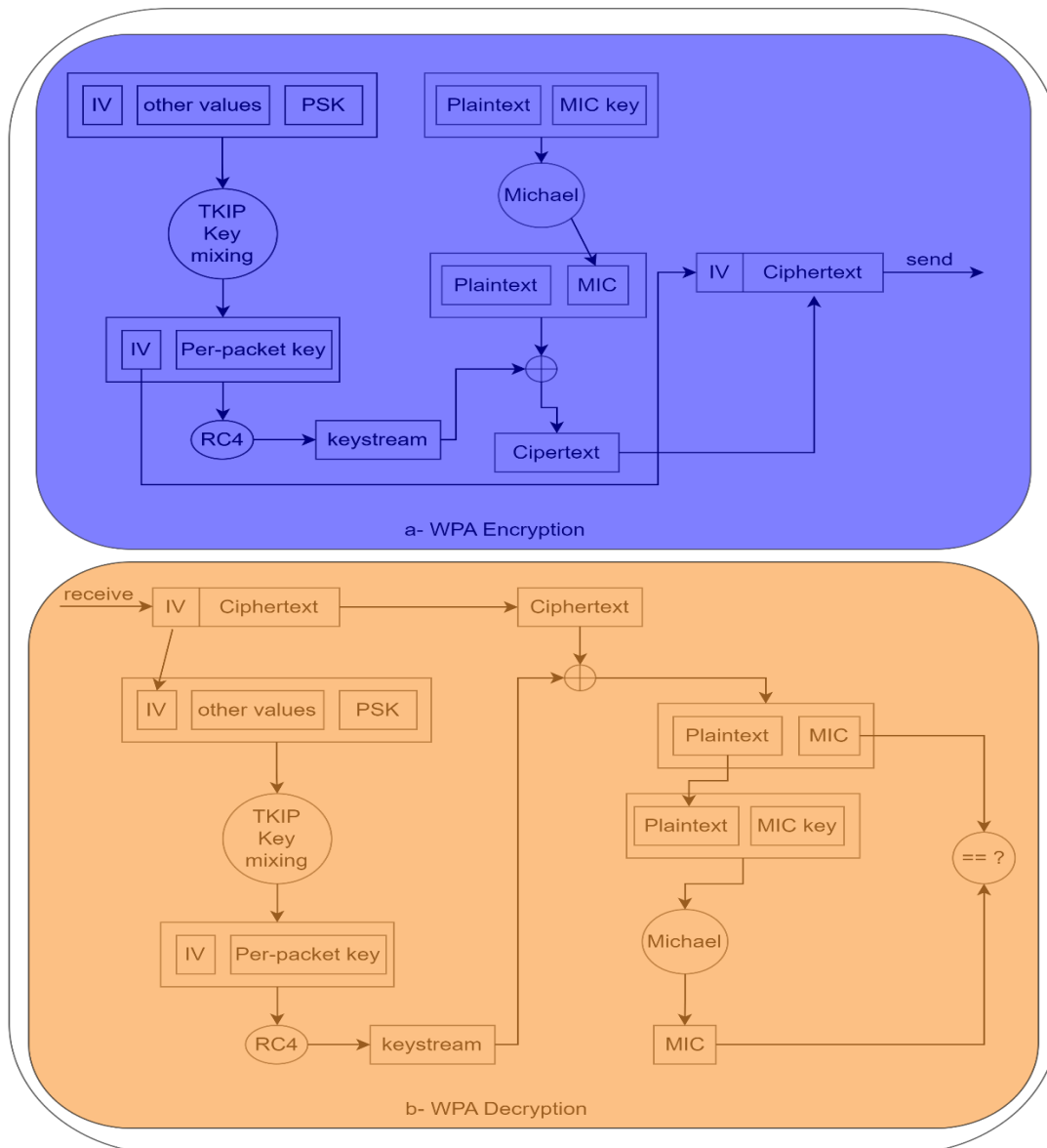


Figure 3.4: WPA TKIP encryption and decryption

3.3.2.3 Improvement Over WEP

- a. WEP used a short IV (24-bit) while WPA uses a larger IV (48-bit)

- b. The weak CRC-32 that was used in WEP data integrity has been replaced with a stronger algorithm known as Michael.
- c. WEP used PSK to encrypt the data while the algorithms TKIP and Michael in WPA derived a set of temporal keys from the PSK and other values.

3.3.2.4 Weaknesses

While having a secret key encryption that was more secure than that of WEP, WPA nevertheless featured a number of flaws that it had acquired from the earlier WEP standard and from the weak stream encryption algorithm RC4.

TKIP was a great move forward since it protected each data packet that was transferred from one device to another using a per-packet key system. However, the new TKIP WPA protocol recycled several features of the vulnerable WEP, hence, the same issues, that was found in the previous WEP, soon surfaced in the new WPA protocol.

When it was first deployed in 2003, the WPA was doomed to fail from the very early stages. As a result, the WPA was only intended to be a short-term solution until it was quickly replaced with the more robust WPA2 in 2004.

3.3.3 WPA2

The IEEE 802.11i standard, often known as WPA2, was ratified, in 2004, to enhance the security of wireless networks significantly better than the IEEE 802.11 standard WEP protocol, which had been found to have significant security flaws.

3.3.3.1 Authentication and Key Exchange

WPA2 protocol works in two modes, like its predecessor WPA, personal mode (WPA2-PSK) and enterprise mode (WPA2-Enterprise).

The passphrase for the WPA2-PSK, also known as the password, is pre-installed in both the AP and the wireless clients, and it must have a length of between 8 and 63 characters. The passphrase is kept secret and is used to generate all of the necessary keys that are required to secure the WPA2-PSK network.

3.3.3.1.1 Key Negotiation

WPA2-PSK defines two types of keys (Cremers, Kiesl, & Medinger, 2020):

1. Pairwise key: to protect unicast between the client and AP.
2. Group key: to protect broadcast or multicast between clients and AP.

In WPA2-PSK authentication process, the AP generates a Group Temporal Key (GTK), and the GTK is then sent to the clients that are connected to that AP. GTK is used to protect data confidentiality during broadcast or multicast communication between the AP and its clients. The same GTK is used by every client that connects to the same AP.

At both nodes, a 256-bit Pairwise Master Key (PMK) is obtained from the PSK, providing the client is informed of the AP's password. Then, in the four-way handshake, the PMK is used to derive 384-bit Pairwise Transient Key (PTK). The PTK, then, used to generate three keys (Sakib, Jaigirdar, Munim, & Akter, 2011):

1. 128-bit Key Confirmation Key (KCK): used to maintain data integrity during the four-way handshake.
2. 128-bit Key Encryption Key (KEK): used to protect data confidentiality during the four-way handshake.
3. 128-bit Temporal Key (TK): used to protect data confidentiality during client-AP communication following a successful four-way handshake.

Figure 3.5 shows the hierarchy of Pairwise keys in WPA2-PSK protocol.

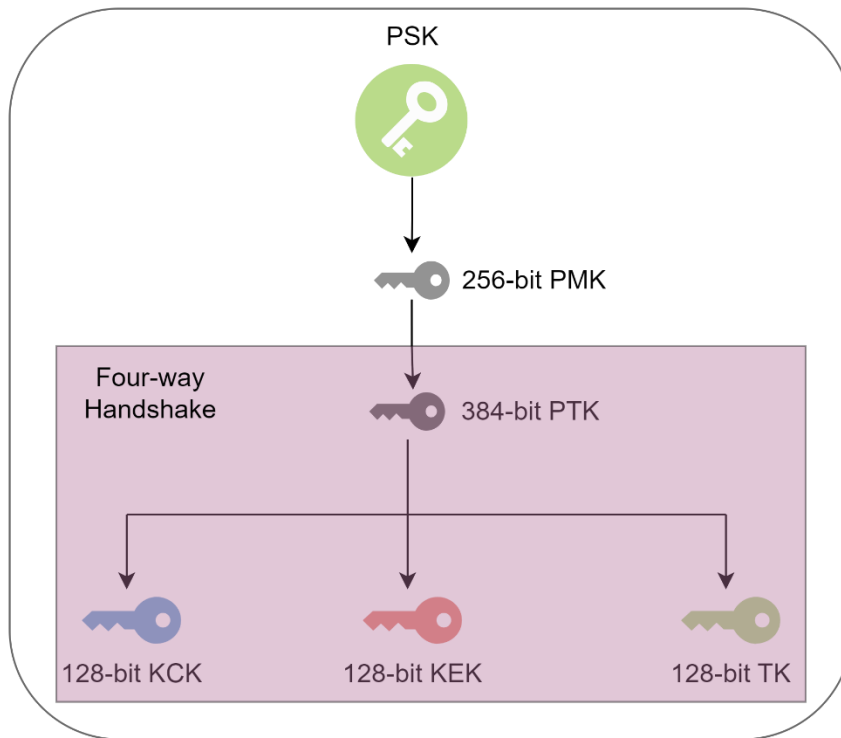


Figure 3.5: Pairwise keys hierarchy

3.3.3.1.2 Four-way Handshake (Key Exchange)

In order to fix flaws in the way that WEP handles security, the core 802.11 protocol was updated to include the 4-way handshake. Mechanism constructed so that an AP, also known as an authenticator, and a Wi-Fi client, also known as a supplicant, can simultaneously demonstrate to one another that they are knowledgeable with the PSK or PMK, without transmitting the PSK.

In the event that the handshake is successful, the primary objective of the four-way handshake is to construct the PTK, which is then put to use in the generation of the TK. TK is what secures the unicast communication between the client and the AP. In order to generate the PTK, both the client and the AP require a number of parameters:

PMK: is derived from the PSK in both, the client and the AP.

ANONCE: a random number generated by Authenticator and used only once.

SNONCE: a random number generated by the client/Supplicant and used only once.

MAC(A): Authenticator's mac address.

MAC(S): Supplicant's mac address.

$$PTK = F (PMK + ANONCE + SNONCE + Mac (A) + Mac (S))$$

Where F is a pseudorandom function.

All these parameters are available to both the client and the AP, with the exception of ANONCE and SNONCE. The client lacks the ANONCE, while the AP lacks the SNONCE.

The AP sends its ANONCE to the client as the first message of the four-way handshake. At this point, the client possesses all of the resources necessary to generate the PTK. The client then transmits its SNONCE to the AP, along with a MIC, as the second message. The MIC is primarily for the AP to detect that this message is indeed from this client and will be generated using KCK which is derived from the PTK. Now that the AP has received the second message, it possesses everything that is necessary to generate the PTK. The GTK, which has been encrypted with the KEK which is derived from PTK, is sent to the client by the AP as the third message. Finally, the client sends the fourth message to the AP, which indicates that the handshake was successful, and everything is installed and in working state (Singh, Moreira, Chothia, & Ryan, 2020). All the communications involved in the four-way handshake are encapsulated using the Extensible Authentication Protocol (EAP) over LAN (EAPoL

Protocol). Figure 3.6 depicts the four-way handshake process that occurs in the WPA2-PSK protocol.

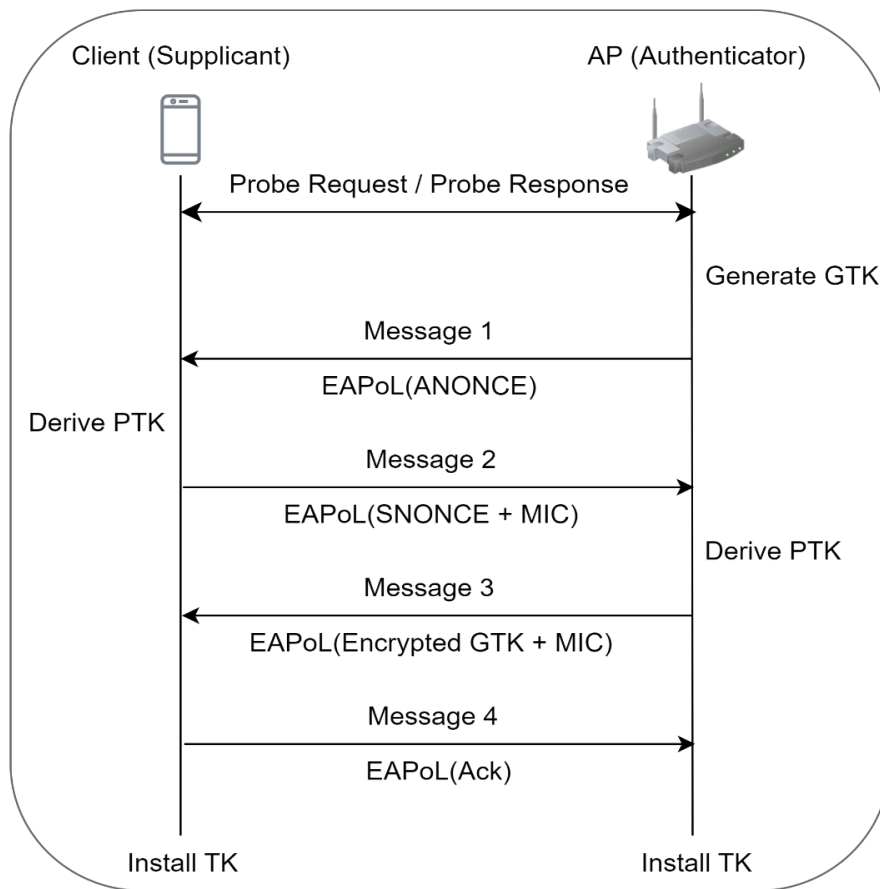


Figure 3.6: WPA2-PSK four-way handshake

Once a certain amount of time has passed, it is likely that the network's GTK will require a refresh. It is also needed to update the GTK when a device disconnects from the network and needs to connect again.

3.3.3.1.3 Group Key Handshake (Two-way Handshake)

WPA2 includes a group key handshake, which is a two-way handshake used to manage GTK updates. The group key handshake is very similar to the last two messages in the four-way handshake (message 3 and message 4). First, the AP will communicate the newly generated GTK to every client that is part of the network. The GTK is encrypted with the KEK, and a MIC, which is generated using KCK, is used to keep GTK from being

tampered with. The client will then respond to the AP with an acknowledgment that it has received the new GTK (Singh, Moreira, Chothia, & Ryan, 2020).

3.3.3.2 Encryption

In 2006, WPA2 became the official successor to WPA. With a new wave of security and encryption enhancements, WPA2 delivered the Advanced Encryption Standard (AES) to commercial Wi-Fi networks. This was the most notable of the WPA2 security and encryption advancements. AES is the security method for many online businesses since it is significantly more secure than RC4.

WPA2 included the Counter Mode (CTR) with Cipher Block Chaining Message Authentication Code (CBC–MAC) Protocol (CCMP) to replace the TKIP. CCMP is a far more secure encryption method. However, TKIP is still included in the WPA2 as a low-cost alternative and provides functionality for devices that only support WPA (Khasawneh, Kajman, Alkhudaidy, & Althubyani, 2014). The encryption and decryption process in the WPA2 protocol using the CCMP-AES standard is demonstrated in Figure 3.7.

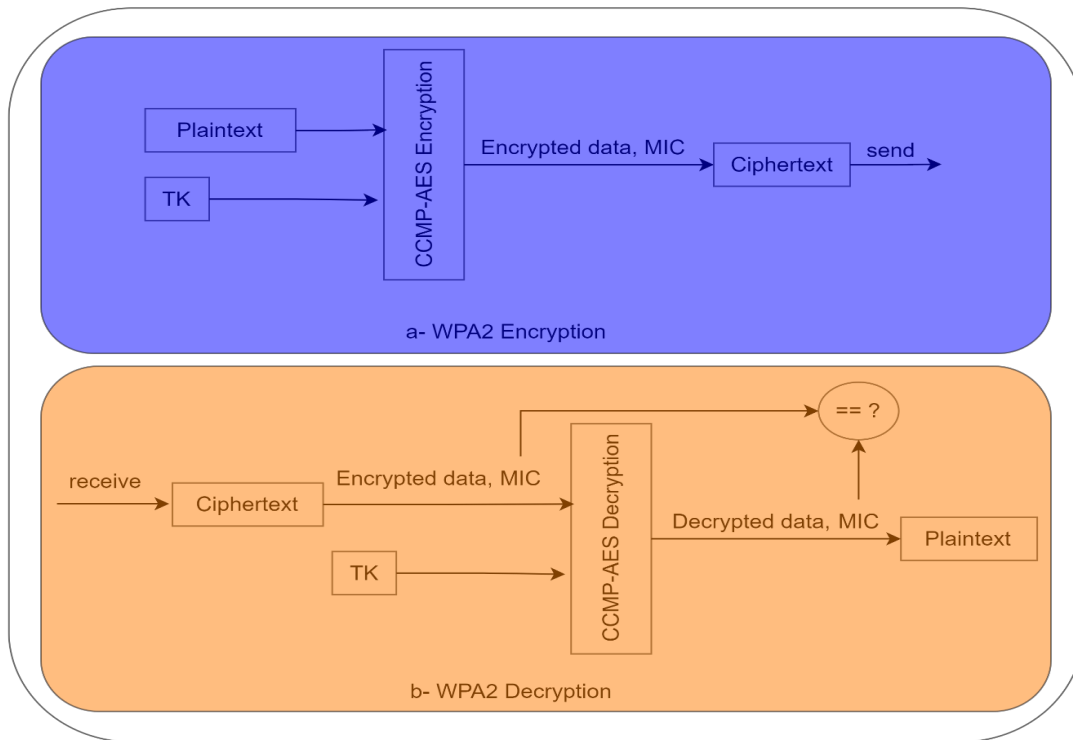


Figure 3.7: WPA2-PSK encryption and decryption

CCMP is based on AES-128 and works in CTR mode with CBC-MAC. Data integrity and authentication are handled by CBC-MAC while data confidentiality is handled by CTR mode. CTR mode is represented in Figure 3.8, while CBC mode is represented in Figure 3.9.

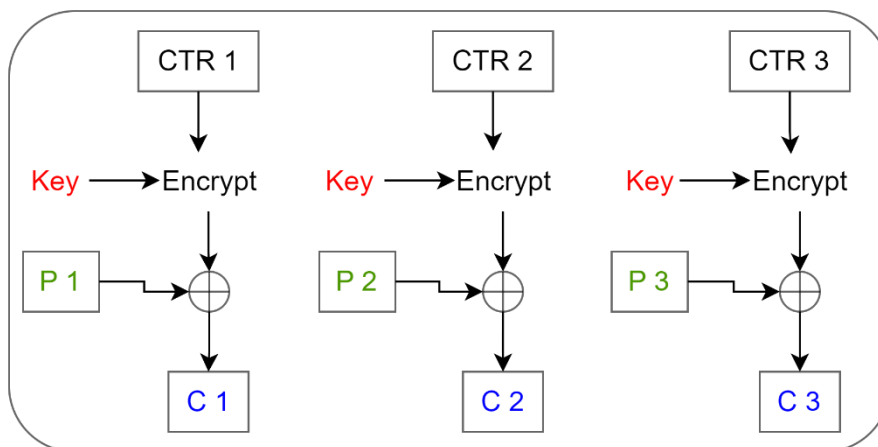


Figure 3.8: AES-CTR mode

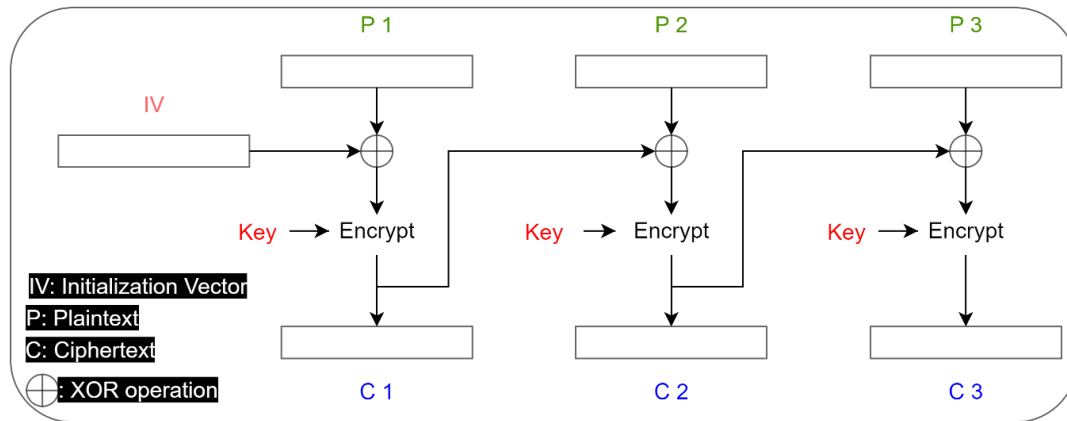


Figure 3.9: AES-CBC mode

3.3.3.3 Weaknesses

WPA2-PSK ensures security and privacy for home and small office networks by using pre-shared keys. Since it is simple to set up and needs only a single password to protect data, WPA2-PSK is also common in public places like airports, cafes, and universities. A malicious actor, however, can simply get access to the network and do harmful actions if WPA2-PSK is compromised (Kohlilos & Hayajneh, 2018).

3.3.3.3.1 Weak Password

The use of a single password for access to a network demands two things: a strong password and the faith that each individual trusted user would keep the password a secret on their respective devices. The rationale for this is that if the security of any one user is breached, then the security of all users is at risk.

3.3.3.3.2 Handshake Dictionary Attack

An attacker can obtain the necessary information from the WPA2-PSK four-way handshake by capturing it while a user is joining the network. This allows the adversary to obtain the information from the first two messages, which are transmitted in clear text. After gaining that piece of information, the attacker is in a position to carry out an offline brute force

attack using a dictionary in order to obtain the PMK. If successfully obtaining the PMK, such as in the case of a weak password, then the attacker will be able to decode all the device traffic with the AP.

3.3.3.3.3 PMK-ID Dictionary Attack

The AP transmits an EAPoL packet to the client during the WPA2 authentication phase, prior to the actual four-way handshake. The heading PMK-ID is contained in this EAPoL frame. The PMK, the client's MAC address, and the AP's MAC address are used to create the hash value, which is the PKM-ID. An attacker can use the PKM-ID to carry out dictionary-based brute-force attacks against the network password, by simply computing the hash using a selection password and comparing the result with the PKM-ID.

3.3.3.3.4 De-authentication Attack

Clients and APs communicate through session management frames. De-authentication frames, which are typically provided by the client and AP to signify the end of a session, are one sort of management frame. Frames are delivered encrypted without authentication under WPA2 session management. An attacker can stop communication between the AP and a client by impersonating the MAC addresses of both the client and the AP and then issuing fake De-authentication frames. De-authentication attack is a type of DoS attack.

3.3.3.3.5 Evil Twin Attack

This is a form of rogue AP attacks, the Evil Twin attack. In an attempt to get a client to connect to it, the attacker pretends to be a certain AP. Because there is no authentication, this attack is straightforward. The attacker will set up the rogue AP in this manner because all that is required for the client to be tricked is for the Service Set Identifier (SSID), MAC address, security

scheme, and password to match. The attacker will then need to send de-authentication frames to remove the client's authentication from the genuine AP. The client will start searching for a connection once it has been disconnected. The attacker becomes a MitM when the client connects to the malicious AP and will be able to intercept, monitor, and manipulate communications that the client is receiving and sending. In order to ensure that the client receives what they desire and is unaware of any intrusions, the attacker will forward Internet access to the client. Nevertheless, by acting as a gateway, the attacker sees all data before the client does.

3.3.3.3.6 KRACK Vulnerability

In 2016, a researcher named Vanhoef and his team at the University of Leuven demonstrated that there is a flaw in WPA2-PSK, and that the Key Reinstallation AttaCK (KRACK) attack has the potential to penetrate the entire protocol without even needing to know the password. In October of 2017, the KRACK vulnerability was widely publicized and discussed (Vanhoef & Piessens, Release the Kraken: new KRACKs in the 802.11 Standard, 2018).

A KRACK attack is a form of replay attack. Using the KRACK vulnerability, an attacker can decode the traffic on a network by repeatedly resetting the nonce in the WPA2-PSK protocol's 4-way handshake. The problem is inside the Wi-Fi standard itself, not within specific products or implementations that are compliant with the specification. As a consequence of this, even if WPA2-PSK network is appropriately configured and protected by a robust password, it is still susceptible to the KRACK vulnerability.

The following is an illustration of how a KRACK assault can be carried out: As long as the client is willing to accept message 3 retransmissions while it is in the condition of having the PTK installed, the attacker is

capable of forcing the PTK into being reinstalled. It is feasible to place a MitM between a client and an AP. This MitM position's aim is to stop message 4 from getting through to the AP, which will ultimately result in the retransmission of message 3. As a consequence of this, the client will be required to reinstall a PTK which has already been utilized. The nonce used in data secrecy protocols is recalculated as a direct consequence of this alteration, this grants an attacker the ability to replay, decrypt, and/or fabricate packets (Figure 3.10).

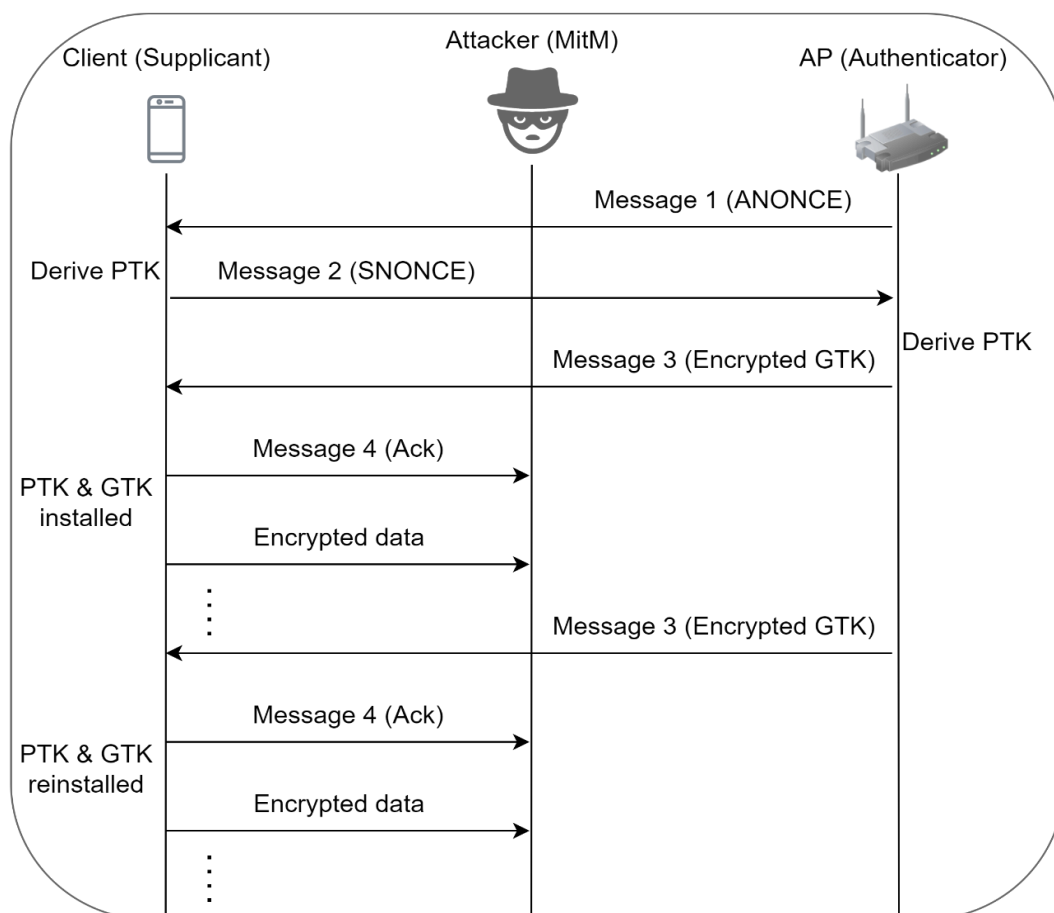


Figure 3.10: KRACK against four-way handshake. Retrieved from (Vanhoeef & Piessens, Release the Kraken: new KRACKs in the 802.11 Standard, 2018)

Because it must be carried out from inside the range of the Wi-Fi network by someone with a high level of technical understanding as well as specific hardware and software, the possibility of a KRACK attack is, fortunately, quite low. The Wi-Fi Alliance has not been made aware of any instances of

KRACK attack use up until this point. In addition, there is no known online tool that can be utilized to carry out an attack of this nature. On the other hand, there is still the possibility that some really crafty cybercriminals have developed their own KRACK attack tools (Alhamry & Elmedany, Exploring Wi-Fi WPA2 KRACK Vulnerability: A Review Paper, 2022).

3.3.4 WPA3

WPA3 is the latest wireless network security protocol that was introduced in 2018. It was designed to replace the previous Wi-Fi security standard, WPA2, which had been in use since 2004. WPA3 was developed to address the security vulnerabilities and weaknesses of its predecessor and to provide stronger protection for Wi-Fi networks. Since WPA3 is mostly derived on WPA2, there are only a few small modifications made to the encryption methods. The authentication protocols in WPA3 went through a significant revision for the new version (Appel & Guenther, 2020).

Two major modifications took place with the implementation of WPA3. Protected Management Frames (PMF) in addition to a more secure technique of authentication known as Dragonfly handshake, also known as Simultaneous Authentication of Equals (SAE) handshake. Moreover, in WPA3-SAE transition mode, a network can use the same password to enable WPA2 and WPA3 to support devices that don't support WPA3 (Bednarczyk & Piotrowski, 2019).

3.3.4.1 PMF

One of the key enhancements of WPA3 is the implementation of PMF, which is designed to protect against attacks targeting the management frames of Wi-Fi networks. Management frames are used by devices to exchange information about the network, such as probe requests, association requests, and disassociation notifications.

In previous Wi-Fi security protocols, management frames were sent in the clear, which made them vulnerable to interception and tampering by attackers. With PMF, management frames are encrypted and integrity-protected, which prevents attackers from tampering with or intercepting these frames. PMF is implemented using the 802.11w standard, which specifies the use of the Counter with CBC-MAC (CCMP) algorithm for encryption and integrity protection.

The use of PMF in WPA3 provides an additional layer of security for Wi-Fi networks, protecting against attacks such as de-authentication attacks and packet injection attacks that target management frames. However, in order to implement PMF, which is a mandatory feature in WPA3, support is required from both the client devices and the AP (Marais, Coetzee, & Blauw, 2020).

3.3.4.2 SAE Handshake

SAE is a Password-Authenticated Key Exchange (PAKE) protocol, and its security properties make it resistant to offline dictionary attacks, even when the passwords used are weak. This is due to the use of a Password Element (*PE*) derived from the password, along with other parameters such as *MAC* addresses and a counter, using the hash-to-element method. *PE* is used to derive a unique key for each session, making dictionary attacks impractical. Additionally, SAE provides protection for PMF by generating encryption keys that make it resistant to de-authentication attacks (Marais, Coetzee, & Blauw, 2020).

Before explaining the SAE in WPA3, keep in mind that SAE here is done by the ECC algorithm. an elliptic curve is the collection of points that can be represented by the equation:

$$y^2 = x^3 + ax + bx \pmod{p},$$

where a and b are values that define the curve,

p is a huge prime number that the curve is defined over,

The parameters a , b , and p are shared between the AP and the client before the WPA3's SAE takes place.

SAE (dragonfly handshake), in WPA3, consists of three phases: *PE* deriving phase, auth-commit phase, and auth-confirm phase (Figure 3.11).

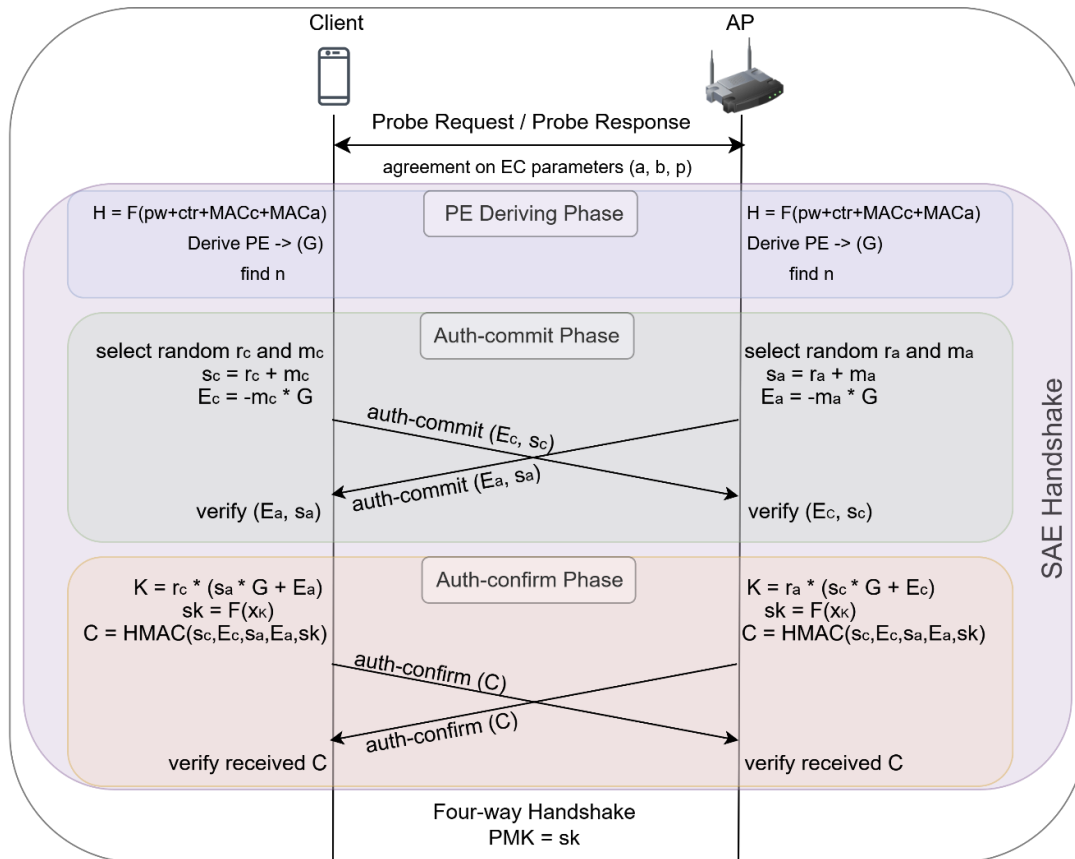


Figure 3.11: WPA3's SAE handshake. Retrieved from (Vanhoeef & Ronen, Dragonblood: A Security Analysis of WPA3's SAE Handshake, 2019)

PE Deriving Phase: *PE* is generated using the hash-to-element method.

A hash function (F) is used to compute the value (H) over:

The password (pw)

A counter (ctr)

The client *MAC* address (*MACc*)

The AP *MAC* address (*MACa*)

$$H = F(pw + ctr + MACc + MACa)$$

H is used as the x value to check if there is a solution for y in the equation $y^2 = x^3 + ax + b \pmod{p}$. If there is no solution to the equation, the ctr is incremented and another try is taken to identify a solution for y by making use of the new value of x . If a solution is found, then PE is the pair of (x, y) . PE is represented by the generator (the base point) (G). Then, a huge prime number (n), the order of G , can be defined:

The smallest prime number that holds $n * G = 0$

Auth-commit Phase:

At the client side:

Choose a random pair of numbers (private key), r_c and m_c , where:

$$1 \leq r_c \leq (n - 1),$$

$$1 \leq m_c \leq (n - 1),$$

$$\text{and } 1 \leq (r_c + m_c) \leq (n - 1)$$

Compute s_c and E_c (public key):

$$s_c = r_c + m_c$$

$$E_c = -m_c * G$$

share s_c and E_c with the AP.

At the AP side:

Do the same, as the client, to find r_a , m_a , s_a and E_a .

Share s_a and E_a with the client.

At each endpoint side verify that the received s and E are valid values:

$$1 \leq s \leq n,$$

E is a valid point in the curve

Auth-confirm Phase:

At the client side:

Calculate the secret shared point K :

$$K = r_c * (s_a * G + E_a)$$

This can be reduced further:

$$K = r_c * ((r_a + m_a) * G + (-m_a * G))$$

$$K = r_c * ((r_a * G) + (m_a * G) - (m_a * G))$$

$$K = r_c * r_a * G$$

Calculate the secret shared key sk :

$$sk = F(x_K),$$

where F is a hash function and x_K is the x value of point K

Calculate the hash value C :

$$C = HMAC(s_c, E_c, s_a, E_a, sk)$$

Send C to the AP.

At the AP side:

Calculate the secret shared point K :

$$K = r_a * (s_c * G + E_c)$$

This can be reduced further:

$$K = r_a * ((r_c + m_c) * G + (-m_c * G))$$

$$K = r_a * ((r_c * G) + (m_c * G) - (m_c * G))$$

$$K = r_a * r_c * G$$

Calculate sk and C (same as the client did).

Send C to the client.

At each endpoint, the received C is verified. If the verification is approved at both ends, then the dragonfly handshake succeeds and, next, the shared secret key (sk) is used as the PMK to derive the TK in the four-way handshake (Vanhoeef & Ronen, Dragonblood: A Security Analysis of WPA3's SAE Handshake, 2019).

Overall, the SAE handshake is a significant improvement over the PSK authentication method used in WPA2. It provides better protection against various types of attacks, including dictionary attacks and de-authentication attacks. Moreover, SAE handshake provides forward secrecy as a new unique session key is generated for each client for every new session. By using the ECC algorithm to derive cryptographic keys, SAE ensures that the keys are strong and resistant to attacks.

3.3.4.3 Transition Mode

In Transition Mode, the WPA3 network can simultaneously support both WPA2 and WPA3 security modes, allowing legacy devices to continue connecting to the network while new devices can use the stronger WPA3 security. The Transition Mode uses the four-way handshake of WPA2 along with the SAE handshake of WPA3.

The process starts with the client sending an association request to AP, which responds with an association response containing information about the supported security modes. Clients of a more recent version connect

using WPA3's SAE with PMF enabled, while clients of an earlier version connect using WPA2 without PMF. Even though it is offered as an optional feature in WPA2, WPA3 clients are mandated to make use of PMF.

In a Robust Security Network Element (RSNE), an AP will broadcast the cipher suites—also known as authentication and encryption algorithms—that it supports. The RSNE is broadcast, without being verified, as part of periodic beacon transmissions that are used to announce the availability of the network. Clients are required to specify the RSNE in association requests in order to provide the AP with information regarding the cipher suite that they want to utilize such as the 4-way handshake or the SAE handshake. Nevertheless, an attacker can fool the unauthenticated RSNE by fabricating their own beacons and using them. During the 4-way handshake, the RSNE of both the AP and the client is cryptographically authenticated so that this issue can be identified and resolved. The RSNE is always confirmed since the 4-way handshake is always done at a certain stage when a client or AP connects to a network for the first time. In the event that a mismatch is found, the handshake will be terminated. This stops a hacker from faking the RSNE and misleading the client into employing a less secure encryption algorithm as a result of their deception (Vanhoeef & Ronen, *Dragonblood: A Security Analysis of WPA3's SAE Handshake*, 2019).

It is important to note that in Transition Mode, the WPA2 handshake does not provide the same level of security as the SAE handshake, and therefore, there are some security considerations to keep in mind. For example, if an attacker can capture the WPA2 handshake of a legacy device, it can use it to perform an offline dictionary attack to recover the passphrase. To mitigate this risk, it is recommended to move away from Transition Mode as soon as possible by upgrading legacy devices to support WPA3.

Overall, Transition Mode provides a way to migrate to WPA3 while still supporting legacy devices. However, it is important to understand the security considerations and to plan for the eventual migration away from this mode.

3.3.4.4 Weaknesses

WPA3 has successfully thwarted the majority of the attacks that could have been carried out on WPA2. Dictionary attacks, de-authentication attacks, and the KRACK vulnerability can all be defended against with WPA3. With WPA3 networks, however, it is still feasible to carry out severe attacks such as rough AP attacks (Evil Twin attacks) (Kohlhos & Hayajneh, 2018). Worse yet, new vulnerabilities in the WPA3 implementation have been uncovered, which unfortunately re-enable dictionary attacks and other forms of DoS attacks to be carried out, again, on WPA3 networks. These are major security flaws that should not be ignored.

The Dragonblood attacks are named after a group of vulnerabilities that can be observed in the personal mode of the WPA3 security protocol. These were published in April 2019 by Mathy Vanhoef, the same researcher who discovered the KRACK vulnerability in WPA2, and Eyal Ronen (Vanhoef & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020). Dragonblood entails a wide variety of attacking techniques in their many forms such as downgrade attacks, side-channel attacks, and SAE handshake DoS attacks.

3.3.4.4.1 Downgrade Attacks

Downgrade attacks on WPA3 can be divided into two main types: transition mode downgrade attack and SAE's group downgrade attack.

1. Transition Mod Downgrade Attack: When WPA3 is in its transition mode, an AP will allow connections using either WPA3-SAE or

WPA2 with the same password. This ensures compatibility with legacy clients, while the 4-way handshake utilized by WPA2 identifies any attempts to lower the security of the network. WPA2's 4-way handshake can identify downgrade attempts, however at that moment an attacker has already collected enough information to launch a dictionary attack. This is because a dictionary attack can be performed with only one captured authenticated 4-way handshake communication. Although in theory only temporary, the transition mode downgrade attack is nonetheless somewhat troublesome. It is reasonable to assume that it will take at least a few years for WPA3 to gain sufficient popularity for it to completely replace WPA2. Until then, most WPA3 networks will most likely operate in transition mode. This means that until this hole is patched, all of these APs are open to brute-force and dictionary attacks.

2. SAE's Group Downgrade Attack: Several elliptic curves or multiplicative groups can be used to execute the SAE handshake, and the wireless system enables nodes to select groups in a user-defined order. This allows for more freedom but requires a secure way of group agreement. A major weakness of SAE is that the process it utilizes to bargain for the target group can be easily compromised. Using SAE, the client can include its preferred group in the commit frame, allowing for the group to be negotiated. In this case, the AP will respond with a commit frame whose status field reads "unsupported group" if it does not support the specified group. The client then sends another commit frame, this time utilizing the next preferred group. This procedure is repeated until the client chooses a curve that is supported by the AP. By faking a commit frame to falsely suggest that the AP does not support the chosen group, it is simple to trick the client into switching to a different

group. The group downgrade attack could be used in DoS attacks or to boost time-based side-channel attack.

3.3.4.4.2 Side-channel Attacks

Side-channel attacks on WPA3 can be divided into two main types: timing-based side-channel attack and cache-based side-channel attack. The hash-to-element algorithms are vulnerable to these side-channel attacks.

1. **Timing-based Side-channel Attack:** The time it takes for an AP to respond to a commit frame varies depending on the password used while deriving the PE from a set of security groups. An attacker can use the leaked information to launch a variant dictionary attack by comparing the actual response time of the AP to the expected response time of a password.
2. **Cache-based side-channel Attack:** In order to carry out this attack, an adversary needs to be in a position to see the pattern of memory access carried out by one of the participants in the SAE Handshake. An adversary is able to learn information about the password being used because of the memory access patterns that occur during the construction of the PE. This knowledge can be put to use in dictionary attacks, which are attacks that compare the patterns that have been observed with the patterns that are expected from a password that is being guessed.

3.3.4.4.3 DoS Attacks

Dragonfly Handshakes' commit phase has a high computational overhead, making it easy for an intruder to overwhelm an AP by delivering fake commit frames. This causes the AP's CPU to work overtime, which might slow down or halt normal operations of the AP. If and how countermeasures are put in place to prevent the side channel attacks mentioned above, the situation could become much more problematic.

3.3.4.4.4 Lack of Post-quantum Protection

The lack of post-quantum protection is another issue with WPA3. Post-quantum cryptography refers to cryptographic algorithms that are secure against attacks by quantum computers, which have the potential to break many of the cryptographic algorithms currently in use. As quantum computers become more powerful, they pose a serious threat to the security of traditional cryptographic algorithms, including those used in WPA3 (Kumar, Ottaviani, Gill, & Buyya, 2022). Therefore, it is essential to develop and implement post-quantum cryptography in Wi-Fi security protocols, including WPA3, to ensure the long-term security of wireless networks.

Table 3.1 provides a brief comparison between the four Wi-Fi security protocols WEP, WPA, WPA2, and WPA3.

Table 3.1: Comparison between the four Wi-Fi security protocols

Properties	WEP	WPA	WPA2	WPA3
description	Provides security similar to that in wired networks	IEEE 802.11i - temporal	IEEE 802.11i - completed	IEEE 802.11w
Release year	1999	2003	2004	2018
Encryption	RC4	TKIP-RC4	AES-CCMP	AES-CCMP
Session key size	40-bit/104-bit	128-bit	128-bit	128-bit

Cipher type	Stream	Stream	Block	Block
Key Management	Not available	Four-way handshake	Four-way handshake	SAE handshake
Integrity	CRC-32	MIC	CBC-MAC AES	SHA-2
PMF	Not available	Not available	Optional	Mandatory
Forward secrecy	Not supported	Not supported	Not supported	Supported
Weaknesses	RC4 vulnerabilities, CRC-32 vulnerabilities, dictionary attack, DoS attack, Evil Twin attack	RC4 vulnerabilities, Offline dictionary attack, de-authentication attack, KRACK, Evil Twin attack	Offline dictionary attack, de-authentication attack, KRACK, Evil Twin attack	Downgrade attacks, side-channel attacks, SAE handshake DoS attack, Evil Twin attack

3.4 Summary

Chapter 3 covers the most commonly used Wi-Fi security protocols, including WEP, WPA, WPA2, and WPA3. WEP was shown to be highly vulnerable to attacks and is no longer recommended for use. WPA and WPA2 are more secure and widely used today, but still have vulnerabilities such as weak passwords and the KRACK vulnerability. WPA3, the latest Wi-Fi security protocol, has improved the security of wireless networks by mitigating many of the vulnerabilities present in WPA2. However, new weaknesses have been discovered in the implementation of WPA3,

including rough AP attacks (Evil Twin attacks), downgrade attacks, side-channel attacks, and DoS (DoS) attacks. Additionally, the current security methods used to protect Wi-Fi networks do not provide post-quantum protection.

In the next chapter, we will discuss our proposed novel solution that could help to mitigate the weaknesses of the WPA3 protocol.

Chapter 4: Proposed Novel Solution to Solve WPA3 Weaknesses

Chapter 4: Proposed Novel Solution to Solve WPA3 Weaknesses

4.1 Introduction

In the previous chapter, we discussed the weaknesses and vulnerabilities present in the Wi-Fi security protocols. While WPA3 provides improved security compared to its predecessor, it still has its own set of issues, particularly in the identification and authentication process. In this chapter, we propose a novel solution to address the main issues identified in WPA3 such as side-channel attacks, SAE handshake DoS attacks, downgrade attacks, Evil Twin attacks, and lack of post-quantum protection.

Our proposed solution combines two approaches: the Kyber Authenticated Key Exchange (KAKE) handshake and the Trust On First Use (TOFU) approach. The technical details, the security features, and the performance of the proposed solution will be further discussed in chapter 5.

4.2 KAKE Handshake

The Cryptographic suite for algebraic lattices Kyber (CRYSTALS-Kyber) was chosen, in the proposed solution, to replace ECC, to be the mechanism for the secret shared key exchange. In this approach, The AP has a static pair of Kyber keys, consisting of a public key (Pu_a) and a private key (Pr_a). These keys will be generated during the initial AP setup process and can be updated at regular intervals or if other settings, such as the AP's name or password, are modified.

Before the KAKE handshake start, the AP will send its public key, Pu_c , to the client. The password is represented by the symbol pw . The proposed KAKE handshake consists of two phases: auth-commit phase and auth-confirm phase.

4.2.1 Auth-commit Phase

Client side:

Generate ephemeral public key Pu_c and private key Pr_c :

$$(Pu_c, Pr_c) \leftarrow Kyber.KeyGen()$$

Generate secret seed (s_c) with its ciphertext (c_c) using Kyber Encapsulation Function that take Pu_a as an input:

$$(c_c, s_c) \leftarrow Kyber.Encap(Pu_a)$$

Calculate the hash value (V_c) using the hash function H :

$$V_c \leftarrow H(pw + MAC_c + MAC_a + s_c)$$

Send (Pu_c, V_c, c_c) to the AP.

AP side:

Receive (Pu_c, V_c, c_c) .

Retrieve (s'_c) using Kyber Decapsulation Function that take Pr_a and c_c as inputs:

$$s'_c := Kyber.Decap(Pu_a, c_c)$$

Calculate the hash value (V'_c) using the hash function H :

$$V'_c \leftarrow H(pw + MAC_c + MAC_a + s'_c)$$

Verify V'_c :

if $V'_c \neq V_c$, Terminate the process.

Generate secret seed (s_a) with its ciphertext (c_a) using Kyber Encapsulation Function that take Pu_c as an input:

$$(c_a, s_a) \leftarrow Kyber.Encap(Pu_c)$$

Calculate the hash value (V_a) using the hash function H :

$$V_a \leftarrow H(pw + MAC_c + MAC_a + s_a)$$

Send (V_a, c_a) to the AP.

Client side:

Receive (V_a, c_a).

Retrieve (s'_a) using Kyber Decapsulation Function that take Pr_c and c_a as inputs:

$$s'_a := Kyber.Decap(Pu_c, c_a)$$

Calculate the hash value (V'_a) using the hash function H :

$$V'_a \leftarrow H(pw + MAC_c + MAC_a + s'_a)$$

Verify V'_a :

If $V'_a \neq V_a$, Terminate the process.

4.2.2 Auth-confirm Phase

Client side:

Derive the secret shared key (sk) using the hash function F :

$$sk \leftarrow F(s_a + s_c + pw)$$

Compute C :

$$C \leftarrow HMAC(V_a, c_a, V_c, c_c, sk)$$

Send C to the AP.

AP side:

Do the same, as the client, to derive sk and to compute C

Send C to the client.

Each party will verify the received C :

Terminate the process if the verification failed (Figure 4.1).

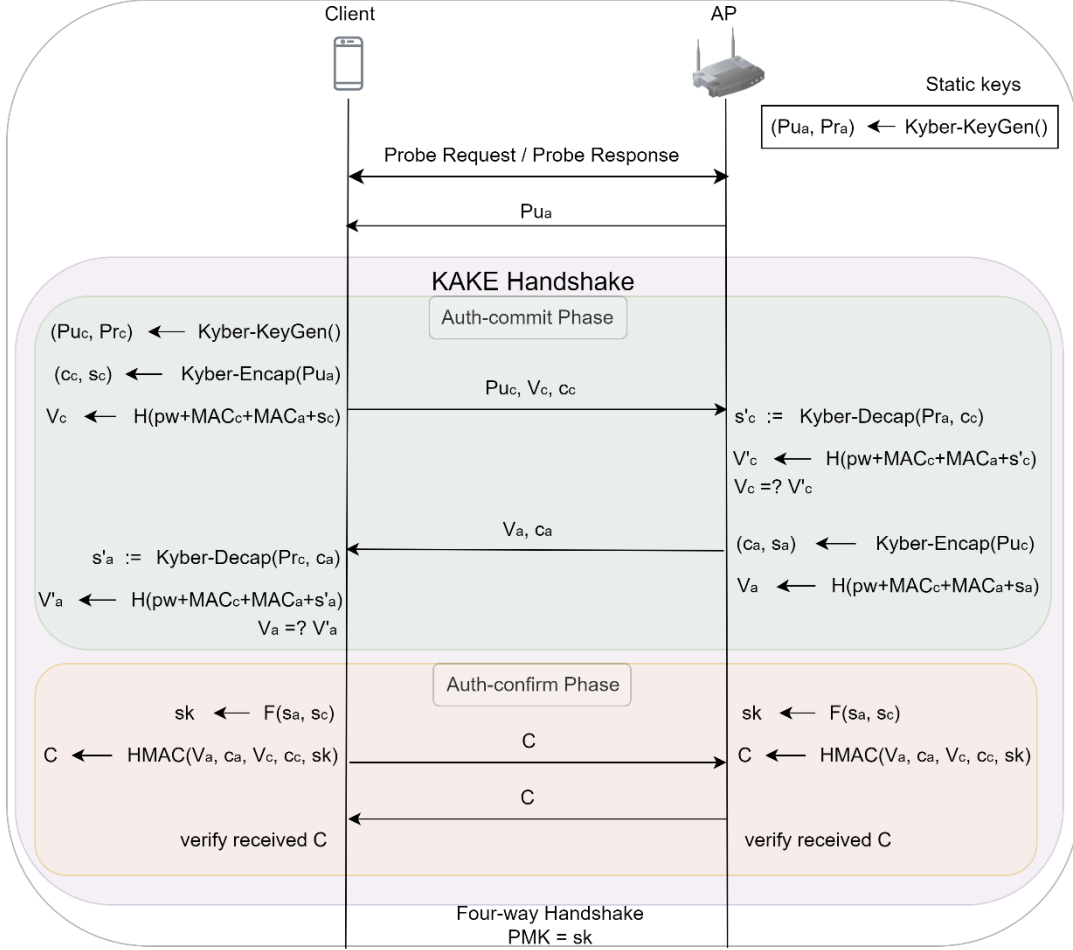


Figure 4.1: The proposed KAKE handshake.

4.3 TOFU Approach

TOFU is a security concept that is used to build trust between a client and a new machine (i.e., an AP) for which trust has not been built in the past. As soon as the connection is established, the client machine will search its internal trust database for the machine's identifier, which is typically some form of public key. In the event that it does not locate one, it will normally return control to the client user so that they can decide whether or not to trust the machine (i.e., AP). After the decision has been taken, the identifier

will be saved in the trust database of the client machine in order to facilitate subsequent connections (Wolf & Ortega-Arjona, 2022).

In the proposed solution, the following are the primary pieces of information that will be saved in the client's internal trust database during the very first connection with the AP:

1. AP's SSID (The network name)
2. AP's MAC address
3. AP's password
4. AP's public key
5. AP's cipher suites (i.e., SAE, KAKE, or four-way handshakes)

When a client attempts to connect to an AP, before the handshake process begins, the client will first search its own internal trust database in order to locate the SSID for the AP in concern. There are two different outcomes that could occur here:

1. The SSID can be located in the trust database that is stored on the client's local machine. Again, there are two different outcomes that could occur in this scenario:
 - a. The data that is associated with this SSID is identical to that which is present on the AP to which the client is attempting to connect. In this particular scenario, the client will go ahead and begin the handshake with the respective AP.
 - b. If there is any data associated with that SSID that is different than that which corresponds to that in the AP, the client machine will ask the user for the network's password and public key.
2. That SSID does not have a record to correspond to it in the client's internal trust database. In this scenario, the client machine will request the user for the network's password and public key; if the

password and public key are valid, the client machine will create a new entry, in its internal trust database, for that AP's SSID and store all of the relevant data associated with it.

First-time clients are the only ones at risk when relying on TOFU. Clients that have previously connected to the reliable AP will notice that the public key of the fraudulent AP does not match the record in the trust database, preventing the fraudulent AP from establishing a connection. However, there is a potential that new users will select the attacker's network rather than the original, reliable one, when they try to connect for the very first time.

4.3.1 Public Key as a Network Name

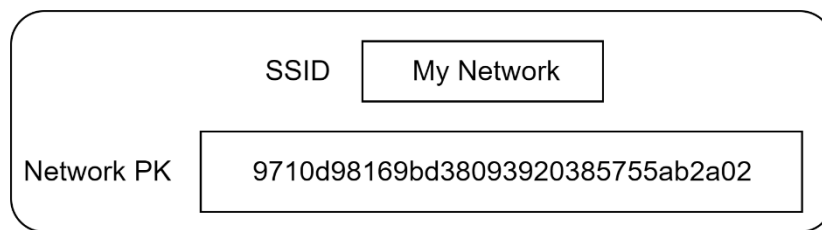
Using the public key of the network as the network name is yet another method of utilizing the network name to protect from attacks carried out by Evil Twins. The method rectifies the primary flaw in the TOFU approach, which is the reality that establishing a connection to a network for the very first time is not a risk-free endeavor. Now, when the network is being set up for the first time, it must produce a pair of a public key and a private key. After that, the public key turns into the network's name.

A client that is interested in connecting can now utilize the AP's public key to verify that it is really the original AP associated with that name. The fact that the private key is known only by the original AP renders it nearly impossible for an attacker to masquerade as the original AP.

Having to double-check for the right public key makes this method of utilizing the public key as the network name less user-friendly. Moreover, public keys are meaningless strings, often quite lengthy in size. It's possible that users won't be able to recognize the public keys appropriately.

4.3.2 Public Key and SSID as a Network Name

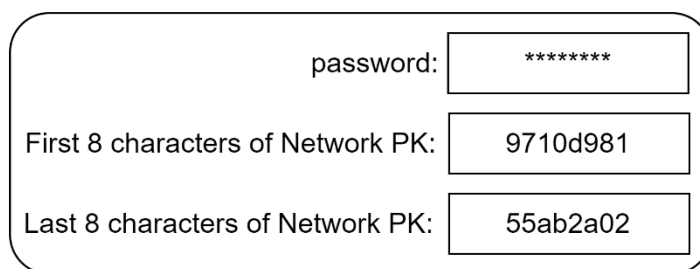
We propose that the network name should be composed of a meaningful string and a public key in order to address the previous problem. Using a hash function (such as MD5), the public key's size will be compressed to create a 128-bit string (i.e., 32 characters). So, when attempting a connection with an AP, the user will get a screen like the one in Figure 4.2.



SSID	My Network
Network PK	9710d98169bd38093920385755ab2a02

Figure 4.2: Network's SSID and hashed public key.

The public key is still less readable and less user-friendly even after being shrunk to 32 characters, though. To further streamline the procedure, we propose that the user should also be asked to enter the first 8 characters and the last 8 characters of the hashed public key when prompted to enter the network's password, as shown in Figure 4.3.



password:	*****
First 8 characters of Network PK:	9710d981
Last 8 characters of Network PK:	55ab2a02

Figure 4.3: Network's password and public key prompt screen.

The user will have to go through this in two cases:

1. When attempting to connect to that particular AP for the very first time.
2. When the AP changes any of its data that is saved in client's internal trust database such as AP's SSID, public key, MAC address, password, and/or cipher suites.

This is the responsibility of the user to ensure that the user picks the correct AP to establish a connection with.

4.4 Summary

The proposed model consists of two components:

1. KAKE handshake to protect against WPA2 and WPA3 vulnerabilities and to provide post-quantum protection.
2. TOFU technique to address downgrade and Evil Twin attacks on WPA3.

Figure 4.4 gives an overview of how the proposed model works to address weaknesses in WPA2 and WPA3 protocols. In the next chapter we will discuss and evaluate the proposed model.

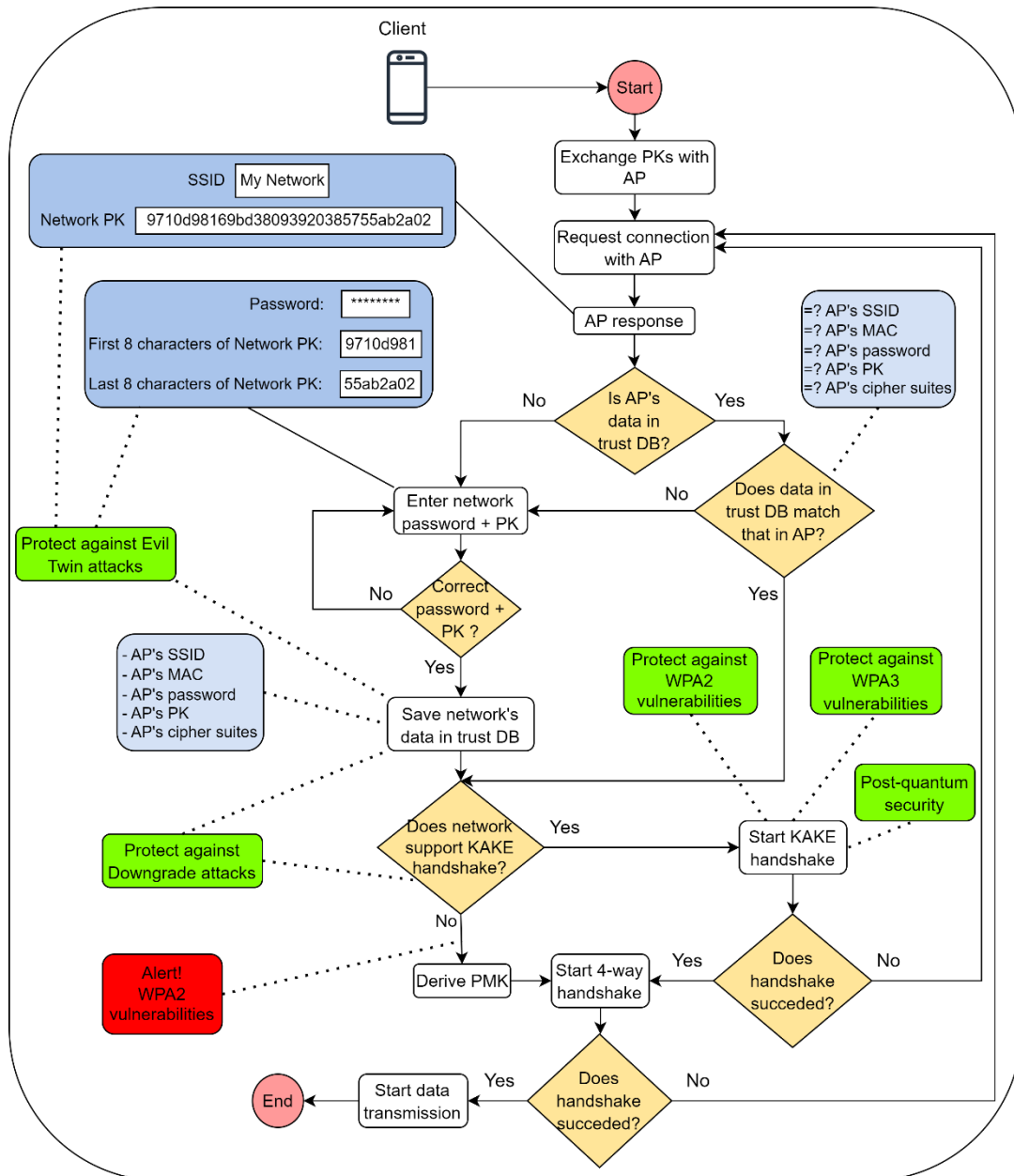


Figure 4.4: High level overview of the proposed model

Chapter 5: Discussion and Feasibility Evaluation of the Proposed Solution

Chapter 5: Discussion and Feasibility

Evaluation of the Proposed Solution

5.1 Introduction

Chapter four presented the proposed solution for securing Wi-Fi networks using a combination of KAKE handshake and TOFU approach. To evaluate the feasibility of this solution, we employ two methods: a proof-of-concept implementation and a review of recent research results.

The KAKE handshake is the basis for our proposed solution, which employs the CRYSTALS-Kyber post-quantum key exchange algorithm. We analyze the algorithm's features, limitations, and performance in terms of memory usage, CPU usage, and network traffic statistics. We also examine a proposed KAKE-SAE hybrid approach used in the WPA3 standard and the TOFU approach used in secure communication protocols. We propose an integration of KAKE and TOFU in our solution, which provides advantages and limitations that we also discuss. Finally, we summarize our findings and conclusions on the feasibility of the proposed solution.

5.2 Feasibility Evaluation

In order to assess the feasibility of our proposed solution for securing Wi-Fi networks, we will be using a combination of methods. The first method is a proof-of-concept approach, where we will discuss our solution and test its functionality and effectiveness. The second method involves looking at the results of recent research studies related to the performance and feasibility of the cryptographic algorithms and protocols we have chosen for our solution.

To conduct our proof-of-concept evaluation, we will be examining several key aspects of our proposed solution, including the KAKE handshake, a proposed KAKE-SAE hybrid approach, TOFU method, and the integration of the KAKE and TOFU approaches. Additionally, we will be evaluating the performance and security of the cryptographic algorithms we have selected, including the CRYSTALS-Kyber family of algorithms.

Through this evaluation, we hope to demonstrate the feasibility of our proposed solution and highlight any areas where further improvements or refinements may be necessary.

5.2.1 KAKE Handshake

The KAKE handshake is a key exchange protocol that provides both classic and post-quantum security. It is one of the two approaches that make up the proposed solution to address the weaknesses in the WPA3 protocol. In this section, we will discuss the KAKE handshake in detail, including its design, implementation, and security properties. We will also explain why we chose the CRYSTALS-Kyber algorithm and how it compares to the existing ECC used in WPA3.

5.2.1.1 What is CRYSTALS-Kyber?

CRYSTALS-Kyber is a type of Quantum-Safe encryption Algorithm (QSA) that is part of the Cryptographic suite for algebraic lattices suite of algorithms (Bos, et al., 2018). This algorithm is one of four encryption algorithms that have been selected by the NIST, in July 2022, for inclusion in its post-quantum cryptographic standard. The four selected encryption algorithms are expected to be included in NIST's post-quantum cryptographic standard, which is anticipated to be finalized within the next two years. Out of the four algorithms selected by NIST, CRYSTALS-Kyber was chosen for general encryption usage due to its advantages, including

relatively small encryption keys that two parties can exchange easily, as well as its fast operation speed. The other three algorithms were selected for digital signature purposes (Hanacek/NIST, 2022). Kyber is considered a "lattice-based" algorithm, which means that its security is based on a class of mathematical problems related to finding the shortest vectors between points in a high-dimensional lattice. Kyber's security is based on the difficulty of solving the Learning-With-Errors (LWE) problem over module lattices (Bos, et al., 2018). In terms of its key sizes, Kyber's secret keys range from 1.6 KB to 3.1 KB, while its public keys are around half that size.

Compared to other NIST algorithm candidates, Kyber is one of the most competitive proposals in terms of performance for key exchange (Alagic G. D., 2022). Its reasonable key sizes and speed operations (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022) make it an attractive option for use in the proposed solution, particularly since it offers post-quantum protection.

CRYSTALS-Kyber has already been adopted by several industry players and is being integrated into libraries and systems. For instance, Cloudflare has included Kyber as part of its Cloudflare Interoperable, Reusable Cryptographic Library (CIRCL), which also features other post-quantum algorithms. Similarly, Amazon has started to support hybrid modes that involve Kyber in its AWS Key Management Service. Moreover, in 2019, IBM showcased the "World's First Quantum Computing Safe Tape Drive" that leverages CRYSTALS-Kyber and CRYSTALS-*Dilithium* (Schwabe, 2020).

Kyber has three different parameter sets aiming at different security levels. Specifically, Kyber-512, Kyber-768, and Kyber-1024 (Schwabe, 2020). Table 5.1 summarizes the different Kyber parameter sets. Each parameter

set offers a different trade-off between security and efficiency, and the appropriate parameter set depends on the specific security requirements of the application.

Table 5.1: Parameter Sets for Kyber KEM.

Parameter Set	Security Level	Public Key Size (bytes)	Private Key Size (bytes)	Ciphertext Size (bytes)
Kyber-512	Livel-1 $\approx$ AES-128	800	1632	768
Kyber-768	Livel-3 $\approx$ AES-192	1184	2400	1088
Kyber-1024	Livel-5 $\approx$ AES-256	1568	3168	1568

While RSA keys are still smaller, Kyber key sizes are in the same magnitude. This is remarkable, as some post-quantum cryptography systems have keys that range in the hundreds of kilobytes or even in the megabyte range, as is the case for *Classic McEliece* (Alagic G. G.-S.-K., 2019).

Kyber-512 is a suitable choice for WPA3, as it offers an acceptable level of security comparable to AES-128, sufficient to resist classical and quantum computer attacks. For enterprise mode, Kyber-768 may be used as an even stronger option.

5.2.1.2 Kyber Types

CRYSTALS-Kyber has two types (Bos, et al., 2018):

1. Kyber-PKE: This type is used for Public-Key Encryption. It provides methods for generating key pairs, encrypting messages, and decrypting messages.
2. Kyber-KEM: This type is used for Key Encapsulation Mechanism. It provides methods for generating key pairs, encapsulating keys, and decapsulating keys.

Both types, Kyber-PKE and Kyber-KEM, are based on the Kyber key exchange algorithm and use the same three security levels (Kyber512, Kyber768, Kyber1024) with different key sizes and ciphertext sizes.

The purpose of each type is to provide a different security service: Kyber-PKE provides secure-CPA public-key encryption, while Kyber-KEM provides secure-CCA key encapsulation.

5.2.1.3 Kyber Algorithms

Notation: The polynomial ring $\frac{\mathbb{Z}[X]}{X^n + 1}$ is represented by R_q . Matrices consisting of polynomials are indicated by bold uppercase letters such as $A \in R_n \times k_q$. Vectors belonging to R_q are represented by bold lowercase letters, and polynomials in R_q are indicated by lowercase letters (Bos, et al., 2018).

5.2.1.3.1 Kyber-PKE

Key pair generation - *Kyber.PKE.Gen()*:

Outputs:

Public key pk ,

Private key sk

Generate random (ρ, σ) :

$$(\rho, \sigma) \leftarrow \{0,1\}^{256}$$

Sample Matrix A :

$$\mathbf{A} := \text{sam}(\rho)$$

Sample vector $\mathbf{s}$:

$$\mathbf{s} := \text{sam}(\sigma)$$

Sample vector $\mathbf{e}$:

$$\mathbf{e} := \text{sam}(\sigma)$$

Compute $\mathbf{t}$:

$$\mathbf{t} := \mathbf{A} * \mathbf{s} + \mathbf{e}$$

Set $sk = \mathbf{s}$.

Set $pk = (\rho, \mathbf{t})$.

Encryption - $\text{Kyber.PKE.Enc}(pk, m)$:

Inputs:

Public key $pk = (\rho, \mathbf{t})$,

plaintext message m , where $m \in \{0, 1\}^{256}$

Outputs:

Ciphertext c

Generate r :

$$r \leftarrow \{0, 1\}^{256}$$

Sample Matrix $\mathbf{A}$:

$$\mathbf{A} := \text{sam}(\rho)$$

Sample $\mathbf{r}$:

$$\mathbf{r} := \text{sam}(r)$$

Sample $\mathbf{e}_1$:

$$\mathbf{e}_1 := \text{sam}(b)$$

Sample e_2 :

$$e_2 := \text{sam}(b)$$

Compute $\mathbf{u}$:

$$\mathbf{u} := \mathbf{A}^T * \mathbf{r} + \mathbf{e}_1$$

Compute v :

$$v := (\mathbf{t}^T * \mathbf{r}) + e_2 + m$$

Set $c = (\mathbf{u}, v)$.

Decryption - $Kyber.PKE.Dec(sk, c)$:

Inputs:

Private key $sk = s$,

Ciphertext $c = (\mathbf{u}, v)$

Outputs:

Plaintext message m

Compute m' :

$$m' := v - (\mathbf{s}^T * \mathbf{u})$$

Set $m = m'$.

5.2.1.3.2 Kyber-KEM

Key pair generation - $Kyber.Gen()$:

Outputs:

Public key pk ,

Private key sk

Generate z :

$$z \leftarrow \{0,1\}^{256}$$

Generate $(\rho, \mathbf{t}, \mathbf{s})$:

$$(\rho, \mathbf{t}, \mathbf{s}) \leftarrow Kyber.PKE.Gen()$$

Set $sk = (\mathbf{s}, z, \rho, \mathbf{t})$.

Set $pk = (\rho, \mathbf{t})$.

Encapsulation - $Kyber.Encap(pk)$:

Inputs:

Public key $pk = (\rho, \mathbf{t})$,

Outputs:

Secret Key K

Ciphertext c

Generate random m :

$$m \leftarrow \{0,1\}^{256}$$

Compute (K', r, d) :

$$(K', r, d) := G(pk, m)$$

where G is a hash function $\{0,1\}^* \rightarrow \{0,1\}^{3 \times 256}$

Compute $(\mathbf{u}, v)$:

$$(\mathbf{u}, v) := \text{Kyber.PKE.Enc}((p, t), m; r)$$

Compute N :

$$N := H(K', c)$$

where H is a hash function $\{0,1\}^* \rightarrow \{0,1\}^{256}$

Set $c = (\mathbf{u}, v, d)$

Set $K = N$

Decapsulation - $\text{Kyber.Decap}(sk, c)$:

Inputs:

Private key $sk = (s, z, \rho, t)$,

Ciphertext $c = (\mathbf{u}, v, d)$

Outputs:

Secret Key K

Compute m' :

$$m' := \text{Kyber.PKE.Dec}(s, (\mathbf{u}, v))$$

Compute (K'', r', d') :

$$(K'', r', d') := G(pk, m')$$

Compute $(\mathbf{u}', v')$:

$$(\mathbf{u}', v') := \text{Kyber.PKE.Enc}((p, t), m'; r')$$

If $(\mathbf{u}', v', d') = (\mathbf{u}, v, d)$:

Set $K = H(K'', c)$

Else:

Set $K = H(z, c)$

Note: The exact implementation of the algorithms may vary depending on the specific implementation of Kyber-PKE or Kyber-KEM.

5.2.1.4 Parameters of Kyber

Table 5.2 summarizes the different parameters values in Kyber algorithm (from NIST post-quantum cryptography standardization, round three).

Table 5.2: Parameters values in Kyber

Parameter	Value
n	256
q	3329
k	2 (<i>Kyber512</i>), 3 (<i>Kyber768</i>), or 4 (<i>Kyber 1024</i>)
η	3
du	10 (<i>Kyber512</i>), 10 (<i>Kyber768</i>), or 11 (<i>Kyber 1024</i>)
dv	4 (<i>Kyber512</i>), 4 (<i>Kyber768</i>), or 5 (<i>Kyber 1024</i>)
δ	2^{-139} (<i>Kyber512</i>), 2^{-164} (<i>Kyber768</i>), 2^{-174} (<i>Kyber1024</i>)

The value of n is set to 256, which indicate a uniformly random 256-bit string used to pseudo-randomly generate a public matrix of polynomials $\mathbf{A}$. q is the smallest prime satisfying $q \equiv 1 \bmod 2n$, and is set to 3329. k is set to 2 for Kyber512, 3 for Kyber768, or 4 for Kyber1024. k is the rank of polynomials in the public key and the rank of polynomials used in the error correction step. It controls the dimensions of the lattice and thereby largely the security. η is set to 3. η is the standard deviation of the error distribution. It is used to balance security and failure probability δ , as well as to determine the size of the error vectors used in the key generation and encapsulation steps. The values of du represents the digits for encryption of u , and is set to 10 (for Kyber512 and kyber768) and 11 (for Kyber1024). dv represents the digits for encryption of v , and is set to 4 (for Kyber512 and kyber768) and 5 (for Kyber1024). The failure probability δ is set to 2^{-139} for Kyber512, 2^{-164} for Kyber768, or 2^{-174} for Kyber1024. These parameters are chosen to balance security, failure probability, public-key size, and ciphertext size (Alagic G. D., 2022) & (Coron, Gérard, Montoya, & Zeitoun, 2021).

5.2.1.5 Kyber-KEM Key Exchange

Kyber offers three types of key exchange: Kyber.KE, Kyber.UAKE, and Kyber.AKE (Bos, et al., 2018).

1. Kyber.KE: This protocol is obtained by directly applying the Kyber-KEM, where the public key is hashed into a pre-key, and the ciphertext is hashed into the final key (Figures 5.1).
2. Kyber.UAKE: This protocol is a one-sided (unilateral) Authenticated Key Exchange (AKE) protocol. In this protocol, one party (P1) knows the static (long-term) key of the other party (P2) (Figures 5.2).

3. Kyber.AKE: This protocol is an authenticated key exchange, where each party knows the static (long-term) key of the other party (Figures 5.3).

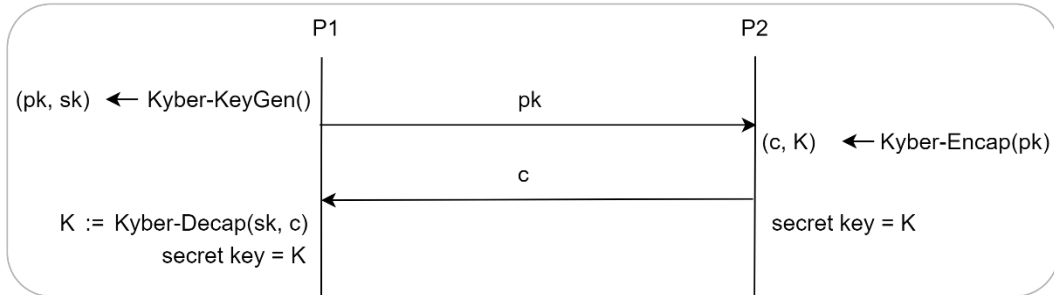


Figure 5.1: Kyber.KE key exchange. Retrieved from (Bos, et al., 2018)

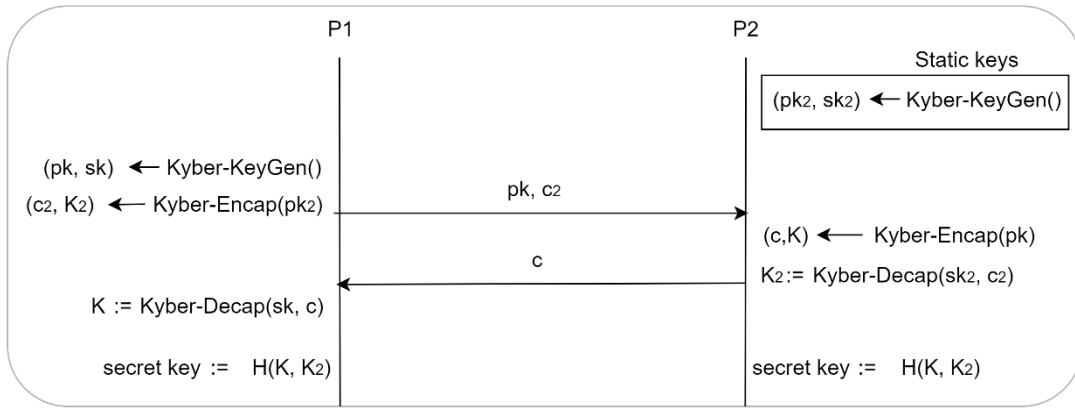


Figure 5.2: Kyber.UAKE key exchange. Retrieved from (Bos, et al., 2018)

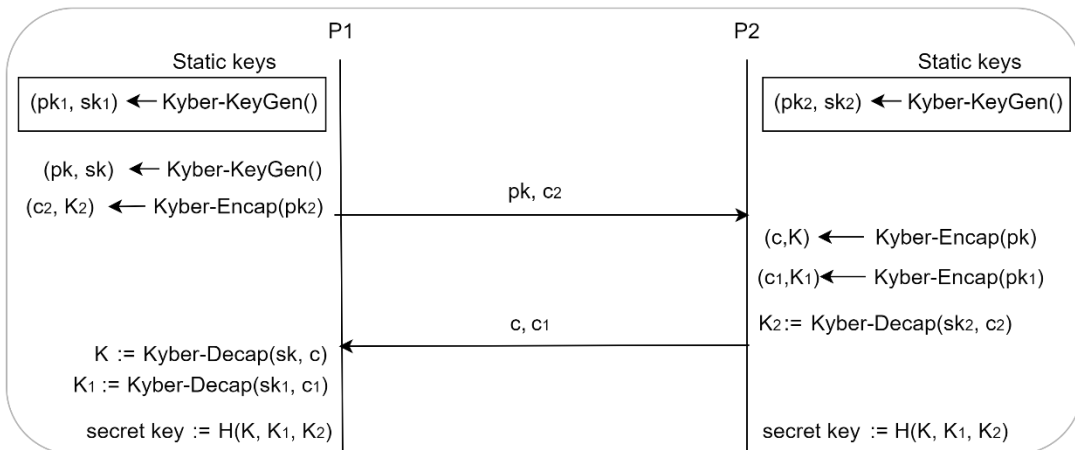


Figure 5.3: Kyber.AKE key exchange. Retrieved from (Bos, et al., 2018)

The shared key obtained in Kyber.KE protocol includes the complete "view" of each participant. However, this protocol only provides security

against passive adversaries and fails to protect against MitM attacks. On the other hand, the shared secret key, derived from Kyber.UAKE and Kyber.AKE protocols, depends not only on the ephemeral key and ciphertext but also on the static (long-term) keys and associated ephemeral ciphertexts. The authenticated key-exchange protocols follow a generic construction from any CCA-secure encryption scheme. In summary, Kyber.KE is the simplest and most straightforward option, Kyber.UAKE is suitable for situations such as when a pre-shared secret, or a password is available, and Kyber.AKE is the most secure option where both parties have long-term public keys.

Kyber.UAKE can be a possible implementation for WPA3, as it provides authentication and key exchange between two parties with pre-shared secret and only one party, the AP, having the long-term key.

5.2.1.6 Kyber Security Features

Here are some key security features of Kyber (Bos, et al., 2018):

- Secure-CCA security: Kyber is a secure-CCA KEM based on the hardness of the LWE problem.
- Authenticated key exchange: In addition to the KE-KEM, Kyber also provides options for authenticated key exchange, including UAKE and AKE. Both options provide secure key exchange and protect against active attacks such as MitM attacks.
- Hashing for security: Kyber uses hashing to enhance security.
- Forward secrecy: Kyber provides forward secrecy, which means that even if an attacker compromises the long-term secret key used to encrypt messages or the secret shared (password), they will not be able to decrypt previously encrypted messages. This is because each session is encrypted with a unique, one-time session key that is

derived from the long-term secret keys and from the ephemeral secret keys that is changed with each session.

- Resistance to quantum attacks: Kyber is resistant to attacks by quantum computers due to its use of LWE-based cryptography. This makes it a potentially useful option for post-quantum security.
- Resistance to side-channel attacks: Some implementation of Kyber is designed to be resistant to side-channel attacks, which are attacks that exploit weaknesses in the physical implementation of a cryptographic system, such as power consumption or electromagnetic radiation. This means that it is supposed to be secure even if an attacker has physical access to the device running the encryption algorithm (Jati, Gupta, Chattopadhyay, & Sanadhya, 2021) & (Hamoudi, et al., 2021).
- Security analysis: Kyber has undergone extensive security analysis, including formal security proofs and testing against a variety of attacks. This ensures that it provides a high level of security against both classical and quantum attacks (Hanacek/NIST, 2022).
- Low computational and memory overhead: Kyber has a low computational and memory overhead, which means that it can be used in resource-constrained environments such as embedded systems or mobile devices (Da Costa, Camponogara, López, & Moisés, 2022) & (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).

5.2.1.7 Kyber Limitation

Some limitations or disadvantages of Kyber are:

- Limited adoption: Kyber is a relatively new algorithm and has not been widely adopted yet. This might be a concern for applications

that require well-established and widely tested cryptographic algorithms.

- **Limited documentation:** Compared to other cryptographic algorithms, the documentation for Kyber is relatively limited. This might make it challenging for developers to understand and implement the algorithm correctly.
- **Key sizes:** Kyber uses relatively large key sizes, which can be a disadvantage in situations where memory or bandwidth are limited.

5.2.1.8 Kyber vs. ECC

Kyber and ECC are two different cryptographic schemes with their own strengths and weaknesses. Here are some comparisons:

- **Key size:** Kyber uses larger keys than ECC, but still within acceptable limits for modern cryptography (Alagic G. D., 2022).
- **Security:** While both Kyber and ECC offer comparable levels of security, Kyber's security is based on a different mathematical problem that is believed to be more resistant to attacks from both classical and quantum computers.
- **Speed:** Kyber is generally faster than ECC in both key generation and encryption/decryption operations, although performance can depend on the specific implementation and hardware used (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022).
- **Implementation:** Both Kyber and ECC can be implemented in hardware and/or software (Xing & Shuguo, 2021) & (Maruyama, Nukada, & Matsuoka, 2009). Kyber is generally considered to be less implementation complicated than ECC, as it has a more straightforward key exchange process. Additionally, Kyber's parameters are easier to choose, which can simplify implementation

(Vanhoeft & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020) & (Bos, et al., 2018).

- Flexibility: Kyber is considered to be more flexible in terms of key sizes, offering three parameter sets with varying levels of security and key sizes (Bos, et al., 2018), while ECC depends on the choice of elliptic curve parameters, and it's important to choose appropriate parameters that are both secure and efficient. This can be a limitation to the flexibility of ECC, as the curve's security and efficiency are dependent on specific parameters (Vanhoeft & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020). However, ECC has been widely implemented and standardized, making it more widely compatible across different systems and devices.
- Standardization: ECC is more widely used and standardized than Kyber, although Kyber is gaining momentum and is currently being considered for standardization in various industry and standards bodies (Hanacek/NIST, 2022).

It's worth noting that the relative strengths and weaknesses of Kyber and ECC can vary depending on the specific use case and deployment environment.

5.2.1.9 Kyber Performance

While CRYSTALS-Kyber is known for its efficiency and security, it is important to evaluate its performance in comparison to other post-quantum cryptographic algorithms, and to ECC, to determine its suitability for use in WPA3.

In this section, we will compare the performance of CRYSTALS-Kyber with other post-quantum cryptographic algorithms and ECC, using recent research studies as our data sources. We will consider factors such as key

generation and encryption/decryption speed, key size, memory usage, and network traffic statistics, and draw conclusions about the relative performance of these cryptographic algorithms.

As part of the NIST standardization process, in its third phase, for post-quantum cryptographic algorithms, researchers (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022) have conducted performance measurements on several potential candidates for securing wireless networks. In their study, they evaluated the suitability of these algorithms for use in resource constrained IoT devices, providing guidelines for selecting the best algorithm for different applications. The results of the study indicate that several of the NIST-selected algorithms offer a good balance between efficiency and security, making them a promising option for IoT devices.

5.2.1.9.1 Memory Usage

The memory usage of five KEMs was measured by (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022), including Kyber512. The RAM usage was measured for a single run of the program, which included the key exchange process, key generation, encapsulation, and decapsulation. The results show that Kyber512 requires more memory compared to LightSaber but has similar memory usage to NTRUhps2048509 and NTRULPr653 (see Table 5.3, and Figure 5.4).

Table 5.3: Memory usage comparison of various post-quantum KEMs

KEM	Memory usage (in bytes)
Kyber512	18528
NTRUhps2048509	18080
LightSaber	994
NTRULPr653	14064

FrodoKEM-640	921360
--------------	--------

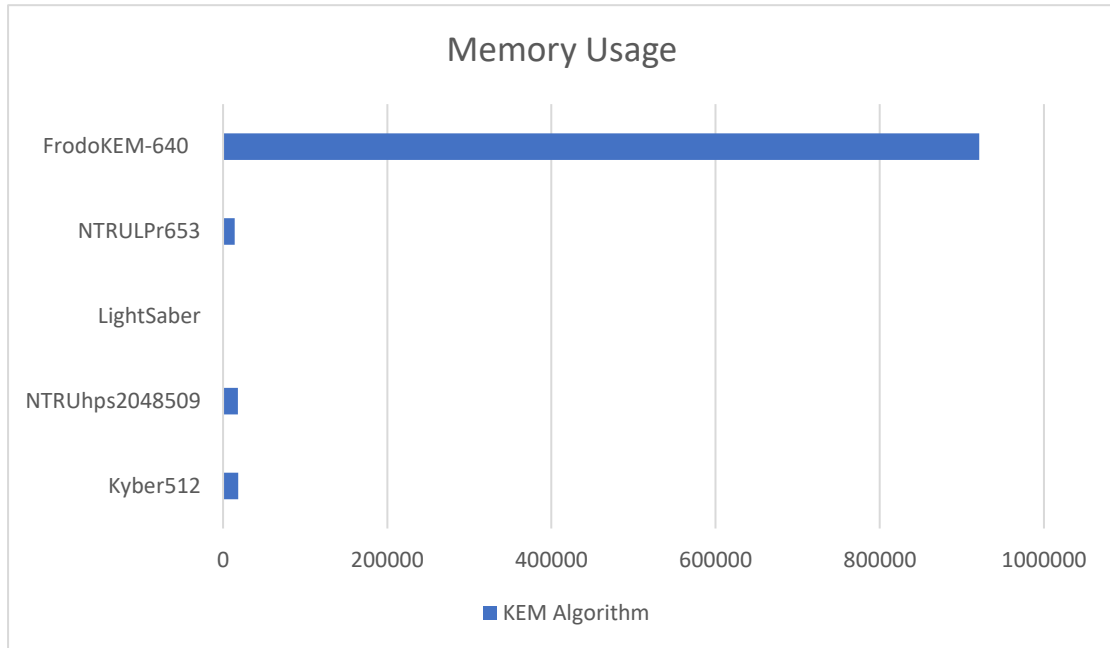


Figure 5.4: Memory usage comparison of various post-quantum KEMs. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022)

5.2.1.9.2 CPU Usage

To evaluate the performance of each KEM in terms of CPU usage, the study (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022) measured the time taken for each operation: key generation, encryption, and decryption. The researchers ran each operation 1000 times and recorded the total time taken for each KEM. The results of this measurement are summarized in Table 5.4 and Figure 5.5 below:

Table 5.4: CPU usage comparison of various post-quantum KEMs in terms of Key Generation, Encryption, and Decryption

KEM	Key Gen (ms)	Enc (ms)	Dec (ms)	Total (ms)
Kyber512	51.3695	69.3704	83.3001	204.0401
LightSaber	63.3441	87.2609	104.7546	255.3597

NTRUhps2048509	7626.0645	288.0449	684.8025	8595.9121
NTRULPr653	10251.6578	19191.646	28705.77	58149.07
FrodoKEM640	911.0205	3468.2189	3444.5259	7823.7653

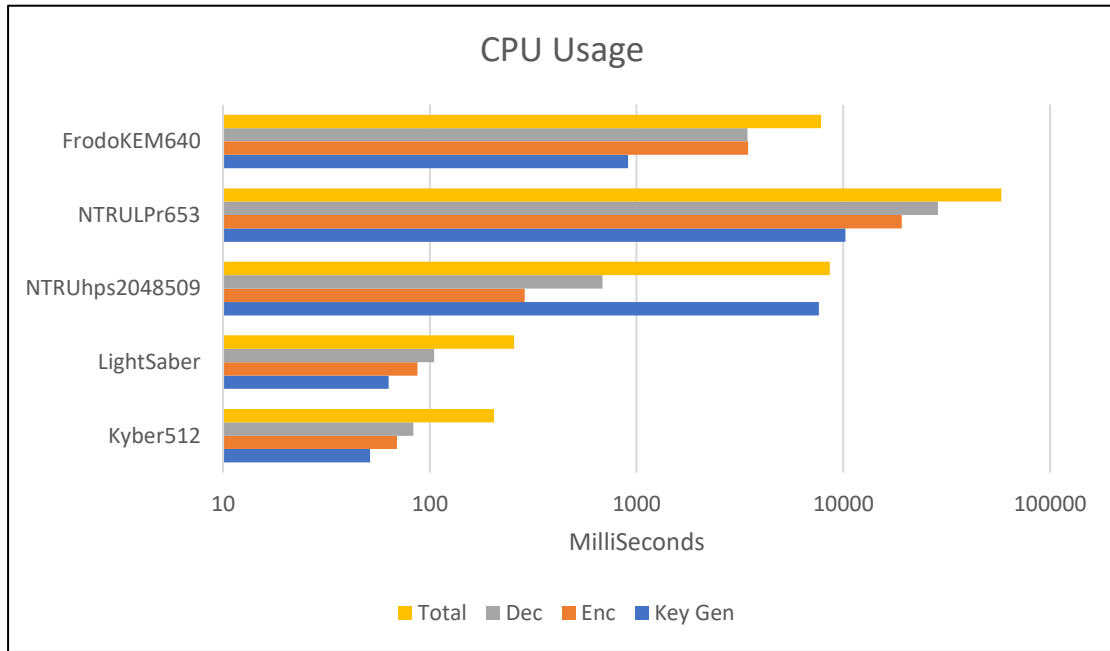


Figure 5.5: CPU usage comparison of various post-quantum KEMs in terms of Key Generation, Encryption, and Decryption. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022)

We can see that Kyber512 is the most efficient KEM in terms of CPU usage, with the lowest total time taken for all operations.

Another study (Tasopoulos, et al., 2022) describes how Transport Layer Security (TLS) 1.3 can be modified to accommodate post-quantum cryptography algorithms from the NIST post-quantum cryptography process, including Kyber with all levels, resulting in a proposed variant called post-Quantum TLS 1.3. The study evaluates the proposed solution on the ARM Cortex-M4 embedded platform NUCLEOF439ZI, which is suitable for resource-constrained devices.

Kyber provides the most superior overall performance when compared to all other methods. Kyber has moderate-sized keys and resulting ciphertexts and also displays exceptional execution speed. In fact, it outperforms the classical Finite Field Diffie-Hellman Ephemeral (FFDHE) by an order of magnitude and even surpasses Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) by multiple milliseconds (Table 5.5, Figure 5.6, and Figure 5.7).

Table 5.5: Comparison of various KEMs in terms of security level, key sizes, and performance metrics

KEM	Security Level	Public Key (bytes)	Ciphertext (bytes)	Key Gen (ms)	Encap (ms)	Decap (ms)
FFDHE	-	256	256	203.92	204.08	-
ECDHE	-	32	32	8.428	17.687	-
Kyber512	1	800	768	8.133	6.239	3.419
BIKE-L1	1	1541	1573	200.62	25.969	411.308
SIKEp434	1	330	346	309.235	498.23	530.2
Kyber768	3	1184	1088	12.224	11.412	7.924
SIKEp610	3	462	486	732.786	1341.1	1346.5
Kyber1024	5	1568	1568	12.918	11.623	8.239
SIKEp751	5	564	596	1262.75	2036.0	2183.0

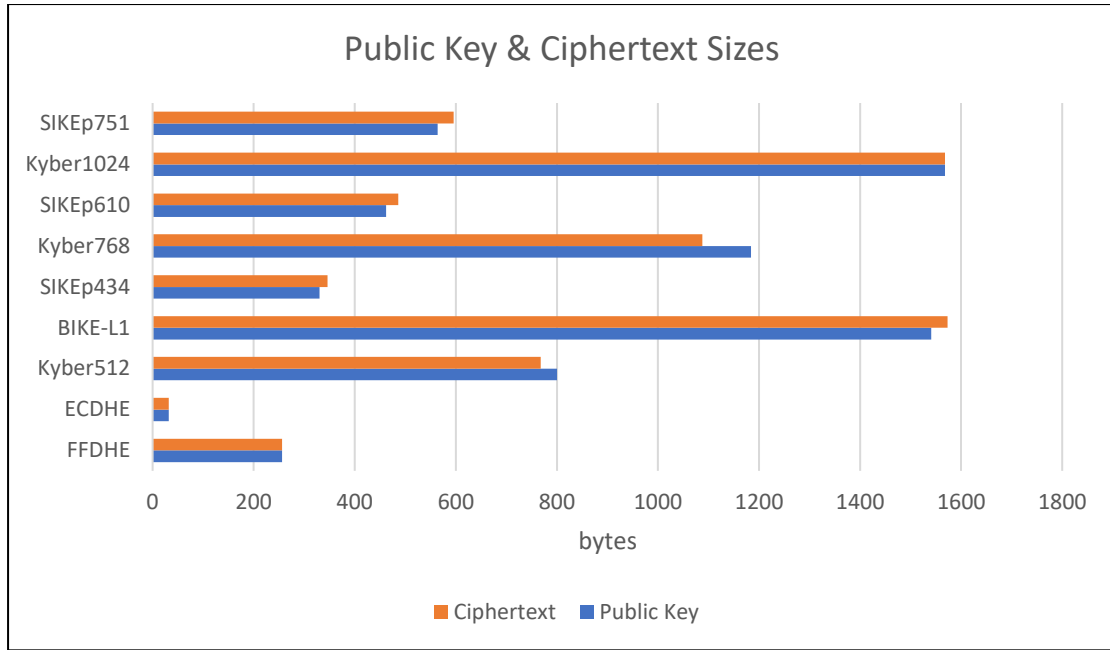


Figure 5.6: Comparison of various KEMs in terms of key sizes. Retrieved from (Tasopoulos, et al., 2022)

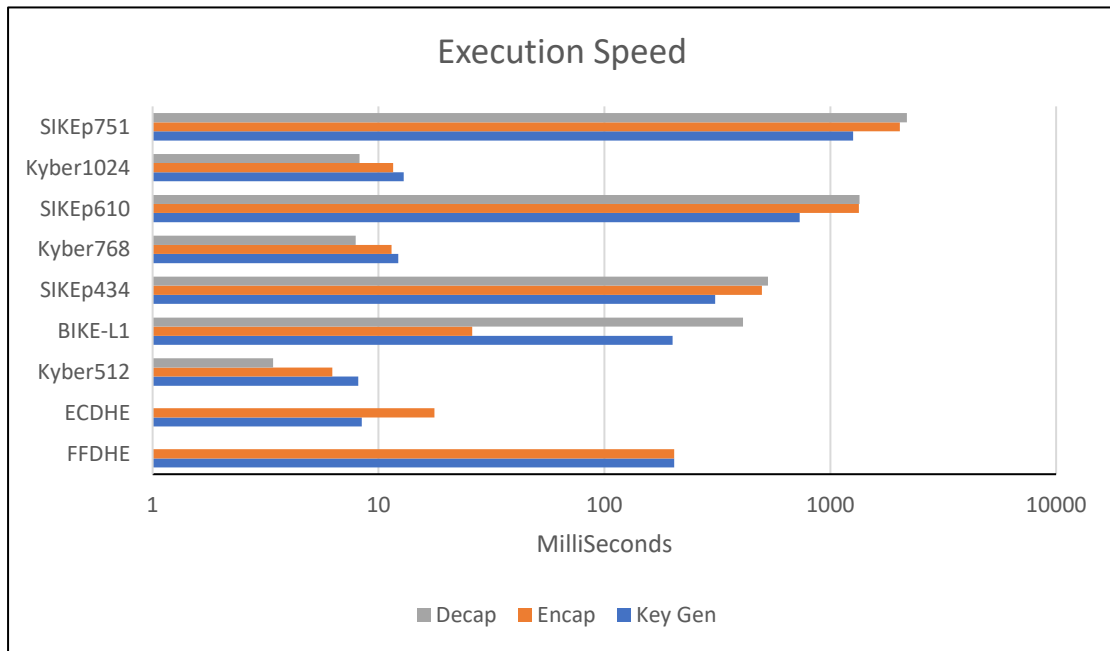


Figure 5.7: Comparison of various KEMs in terms of execution speed. Retrieved from (Tasopoulos, et al., 2022)

The work (Damir, Meskanen, Ramezani, & Niemi, 2022) discusses the computational cost of a protocol for user authentication in 5G networks. The protocol uses asymmetric primitives such as ECIES and CRYSTALS-Kyber for key encapsulation and encryption. The authors compare the running time of the protocol with 5G AKA, which uses ECIES for

Subscription Permanent Identifier (SUPI) encryption. The cost Figures of the asymmetric primitives are presented in Table 5.6 and Figure 5.8, which is generated using open-source cryptographic libraries. The computations are performed on a 3.5 GHz Core processor. The authors use KEM implementations with parameters offering the security level required by 3GPP, equivalent to 128-bit AES (level 1).

Table 5.6: The computational cost of a protocol for user authentication

KEM	At User Equipment (UE)	At Home Network (HN)
ECIES Curve25519	0.04	0.04
ECIES Secp256r1	0.18	0.18
Kyber512	0.026	0.019
BIKE	0.882	0.672
HQC	0.421	0.331

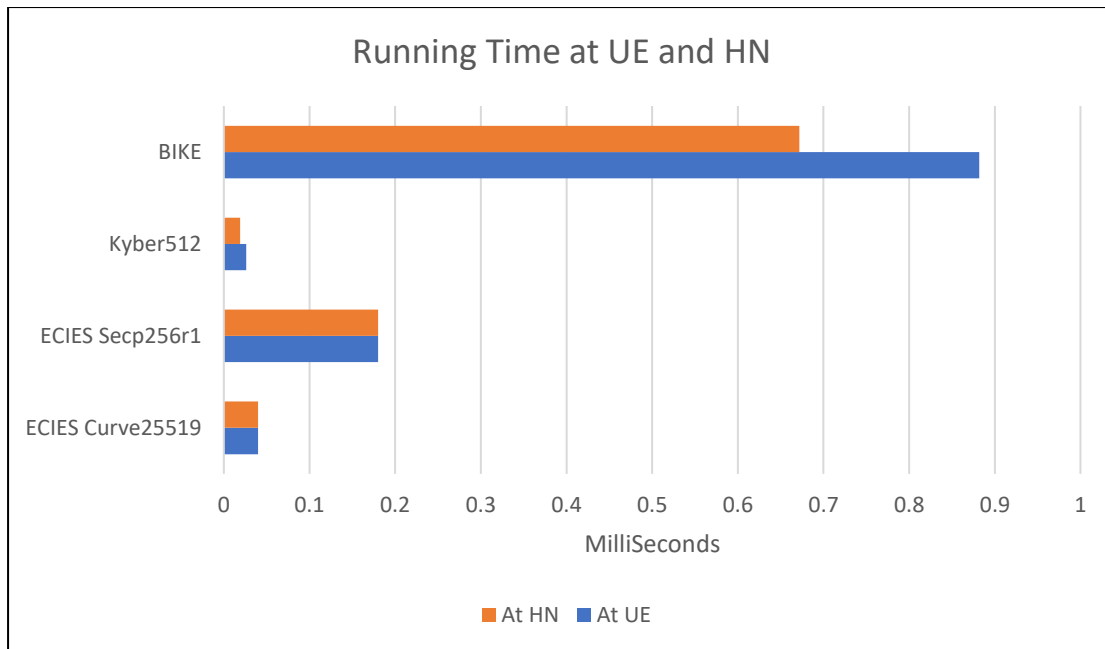


Figure 5.8: The computational cost of a protocol for user authentication. Retrieved from (Damir, Meskanen, Ramezani, & Niemi, 2022)

The study found that Kyber outperformed both ECIES profiles in terms of computational cost. Overall, the study compared the performance of different algorithms in different scenarios and found Kyber to be the most efficient.

The status report on the third round of the NIST post-quantum cryptography standardization process (Alagic G. D., 2022) describes Kyber KEM, and compares its public key and ciphertext sizes with other KEMs, finding them to be acceptable for most applications. The paper also reports that Kyber has fast key generation, encapsulation, and decapsulation in software and has been optimized in various hardware and software settings. In terms of high-speed FPGA implementations, Kyber is a leading performer among the finalist lattice KEMs. Overall, the study suggests that Kyber has sufficient performance in many different environments.

5.2.1.9.3 Network Traffic Statistics

In addition to the performance measurements of Kyber512 and other post-quantum algorithms, (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022) also collected data related to the connection, including the number of packets, packet size, and duration of the connection. The authors compared these metrics between Kyber512 and two widely used elliptic curves, P-256 and X25519.

Number of Packets: Table 5.7 and Figure 5.9 shows the statistics for the number of packets sent over the connection, all computed from a benchmark of 1000 runs.

Table 5.7: Comparison of KEMs based on number of packets sent over the connection

KEM	Number of Packets
Kyber512	24.086
LightSaber	30.086
NTRUhps2048509	28.051
P-256	27.938
X25519	27.926

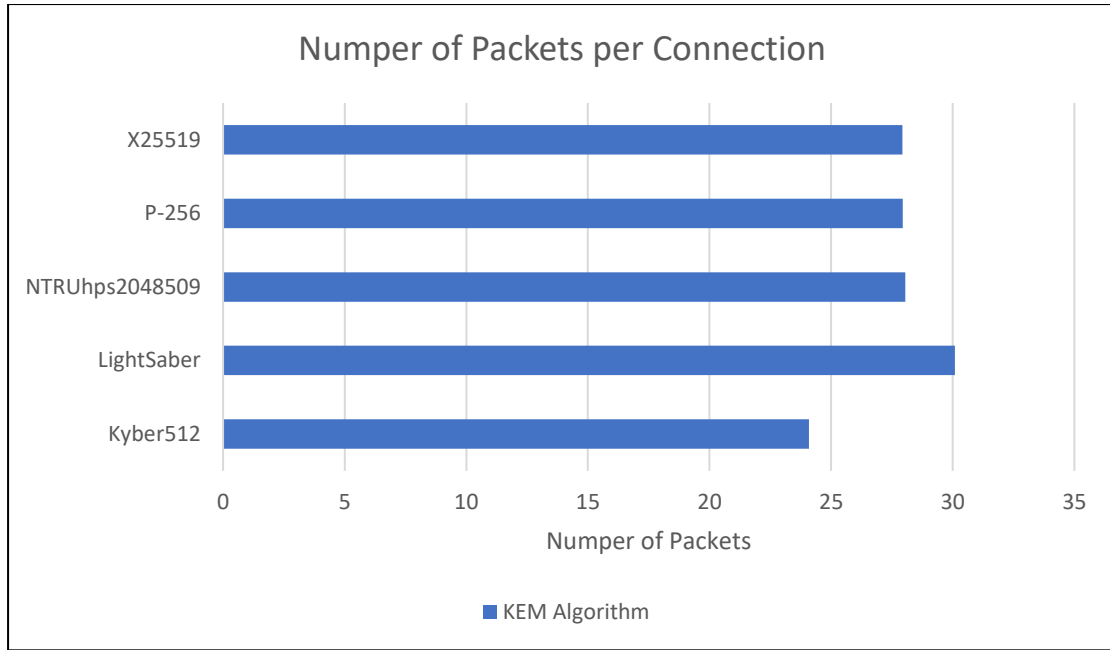


Figure 5.9: Comparison of KEMs based on number of packets sent over the connection. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022)

Kyber512 shows the best performance in terms of the number of packets sent over the connection, followed by the elliptic curves.

Packet Size: Table 5.8 presents the statistics for the packet size on the connection for 1000 benchmark runs. The mean number of bytes for all the connections is shown in Figure 5.10.

Table 5.8: Statistic for the packet size sent by each KEM.

KEM	Mean Packet Size (bytes)
Kyber512	8236.481
LightSaber	9168.695
NTRUhps2048509	9016.969
P-256	7748.311
X25519	7681.331

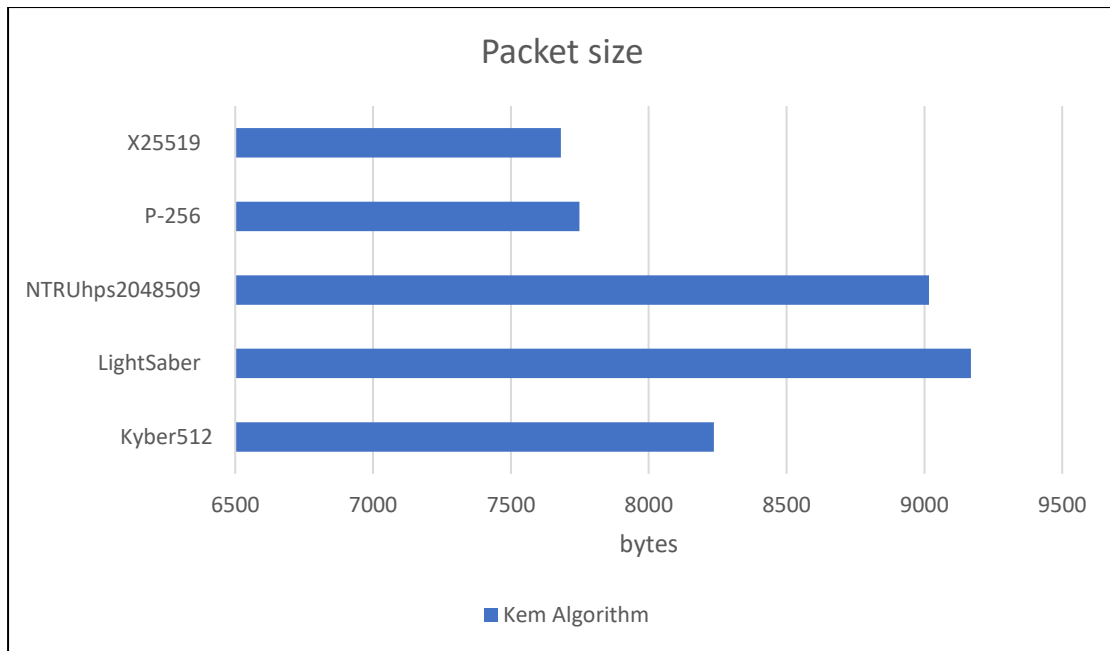


Figure 5.10: Statistic for the packet size sent by each KEM. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022)

Although the elliptic curves outperformed the post-quantum mechanisms in terms of packet size, Kyber512 was still relatively close to them and outperformed both NTRUhps2048509 and LightSaber with a smaller mean packet size of 8263 bytes.

Connection Duration: Table 5.9 presents the connection duration statistics based on 1000 benchmark runs, while Figure 5.11 depicts the mean values for each KEM used, measured in milliseconds.

Table 5.9: Connection duration in milliseconds for each KEM.

KEM	Duration (ms)
Kyber15	0.2133
LightSaber	14.9727
NTRUhps2048509	15.1022
P-256	14.9884
X25512	14.9901

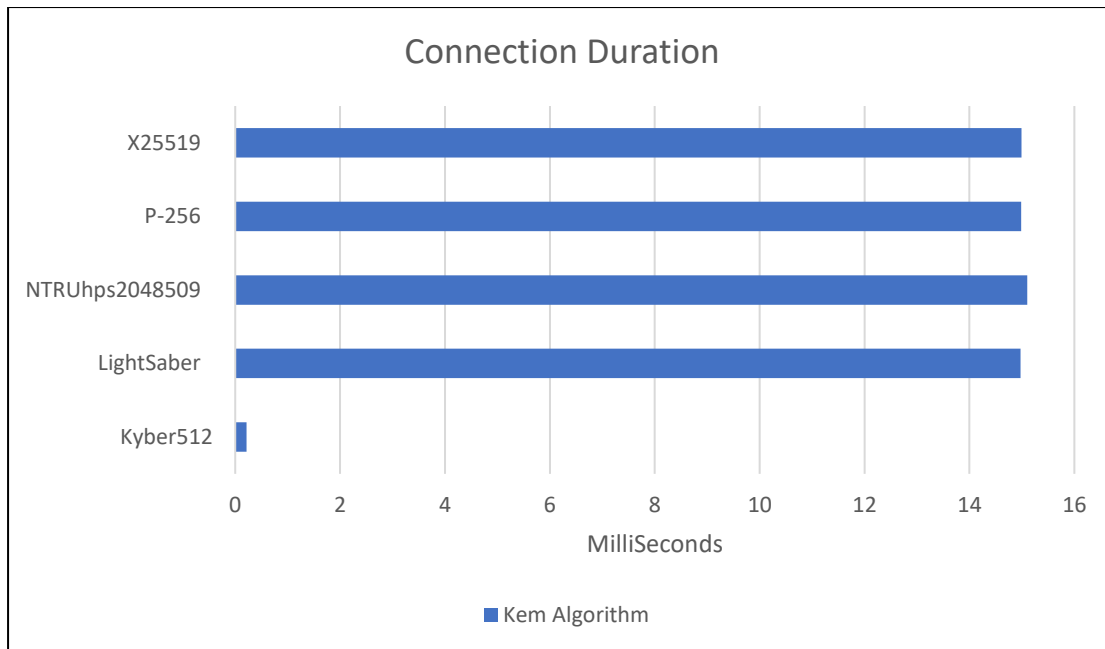


Figure 5.11: Connection duration in milliseconds for each KEM. Retrieved from (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022)

We can see that, except for Kyber512, all KEMs have a duration of about 15ms on this hardware. The best performing is Kyber512, with a mean of 0.21ms, considerably outperforming the other KEMs by a significant factor.

Kyber512 demonstrated superior performance compared to NTRUhps2048509, LightSaber, and even to the elliptic curves.

In terms of the connection data, the post-quantum KEMs follow a consistent order across all cases, with Kyber512 outperforming both NTRUhps2048509 and LightSaber. On the other hand, while elliptic curves outperform post-quantum KEMs in terms of packet numbers, they are outperformed by Kyber512 in the other two categories.

Based on the study of KEMs' performance for resource-constrained devices, it focused on lattice-based mechanisms with the lowest security level considered safe for everyday use. Specifically, it found that Kyber512 and LightSaber were the best performing KEMs, although there is a trade-off between energy consumption and memory storage.

To evaluate the network usage performance, the study compared the KEMs against elliptic curves implemented in the OpenSSL cryptography library. It found that, in general, the elliptic curves performed similarly to the post-quantum KEMs, with Kyber512 sometimes outperforming them.

5.2.1.10 Kyber in Wi-Fi WPA3

WPA3 vulnerabilities, such as DoS and side channel attacks, are mainly related to the hash-to-group and hash-to-curve algorithms used in SAE's password encoding method. These algorithms are used to encode passwords and other data into points on elliptic curves, which are used in the authentication process. However, implementing these algorithms without side-channel leaks is very difficult. Even after patches were released, novel side-channel attacks were still able to exploit vulnerabilities in the implementation. Resource-constrained devices may not fully implement all side-channel defenses due to the high overhead costs. Additionally, correctly implementing all backwards-compatible side-channel countermeasures is non-trivial (Vanhoeft & Ronen, Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd, 2020).

Therefore, solving WPA3 vulnerabilities using ECC, alone, is challenging. Instead, Kyber, a post-quantum cryptographic algorithm, presents a promising solution to tackle these issues since it can be constructed to withstand side-channel attacks (Hamoudi, et al., 2021). Kyber is also efficient and has been optimized for use in resource-constrained devices (Da Costa, Camponogara, López, & Moisés, 2022) & (Septien-Hernandez, Arellano-Vazquez, Contreras-Cruz, & Ramirez-Paredes, 2022). As a post-quantum cryptographic algorithm, Kyber is also resilient against attacks by quantum computers, which are expected to become more powerful in the future. Kyber is a promising candidate for implementation in WPA3, as it is a post-quantum secure key exchange algorithm that has been extensively

researched and peer reviewed (Alagic G. D., 2022). Its security features, such as its resistance against attacks from classical and quantum computers, make it an attractive option for long-term security.

However, while Kyber offers strong security features for post-quantum key exchange, its implementation in WPA3 may require significant changes, which could potentially introduce compatibility issues with existing WPA3 protocol. Additionally, as a relatively new algorithm, vendors may be hesitant to adopt Kyber due to concerns about its maturity and the need for additional testing and verification.

Therefore, careful consideration and evaluation of its advantages, limitations, and implementations are necessary before adopting Kyber as a standard for wireless security. One potential solution is to integrate Kyber with ECC for key exchange in WPA3, creating a hybrid approach that combines the strengths of both algorithms. This would allow for a smoother transition to post-quantum security while still leveraging the established benefits of ECC.

5.2.2 KAKE-SAE Hybrid Approach in WPA3

While KAKE handshake offers strong post-quantum security features and is efficient, vendors may be hesitant to adopt it due to concerns about its maturity and compatibility with existing network infrastructure. In this section, we propose a potential solution to these challenges: a hybrid approach that combines the strengths of KAKE and SAE for key exchange in WPA3.

The implementation of a KAKE-SAE hybrid approach in WPA3 requires careful consideration of several factors:

- **Compatibility:** One important consideration is the compatibility of the KAKE-SAE hybrid approach with existing network

infrastructure. Any changes made to WPA3 to incorporate the hybrid approach would need to be backward-compatible with existing devices to avoid disrupting the network.

- **Performance:** Another consideration is the performance impact of the hybrid approach. While both Kyber and ECC have good performance, combining them may affect the overall performance of the system. Therefore, testing and evaluation of the hybrid approach's performance should be conducted to ensure that it meets the required standards.
- **Security:** Since the hybrid approach combines two different algorithms, it's important to evaluate the overall security of the system.
- **Resource requirements:** Implementing the hybrid approach may require additional computational resources, memory, and power consumption. Therefore, device manufacturers and system designers need to consider these requirements before adopting the approach.
- **Migration:** Any new security standard requires a transition period during which old and new systems need to coexist. Therefore, migration strategies need to be carefully planned and implemented to avoid disruption of the network.
- **Patent and licensing:** Another consideration is the patent and licensing issues associated with Kyber and ECC. Device manufacturers and system designers need to be aware of any patent and licensing requirements before implementing the hybrid approach.

5.2.2.1 KAKE-SAE Hybrid Design

In our proposed hybrid approach, there are three phases in the key exchange: KAKE-auth-commit phase, SAE-auth-commit phase, and Auth-confirm phase.

Figure 5.12 provides a high-level overview of the KAKE-SAE hybrid approach.

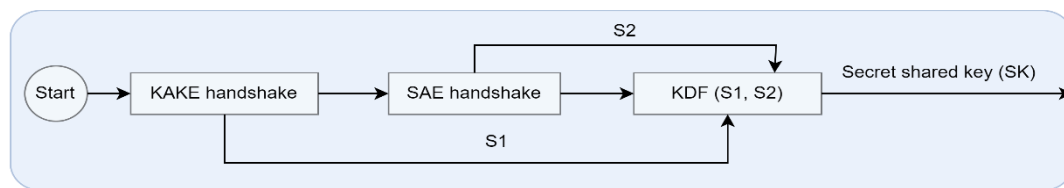


Figure 5.12: High-level overview of KAKE-SAE hybrid approach

During the KAKE-auth-commit phase, the first phase of our proposed KAKE handshake is used to generate the first shared secret (s1). Then, In the SAE-auth-commit phase, the first and second phases of WPA3 SAE handshake are used to generate the second shared secret (s2). Finally, in the Auth-confirm phase, a Key Derivation Function (KDF) is used to derive the final shared key (K) from s1 and s2 (Figure 5.13).

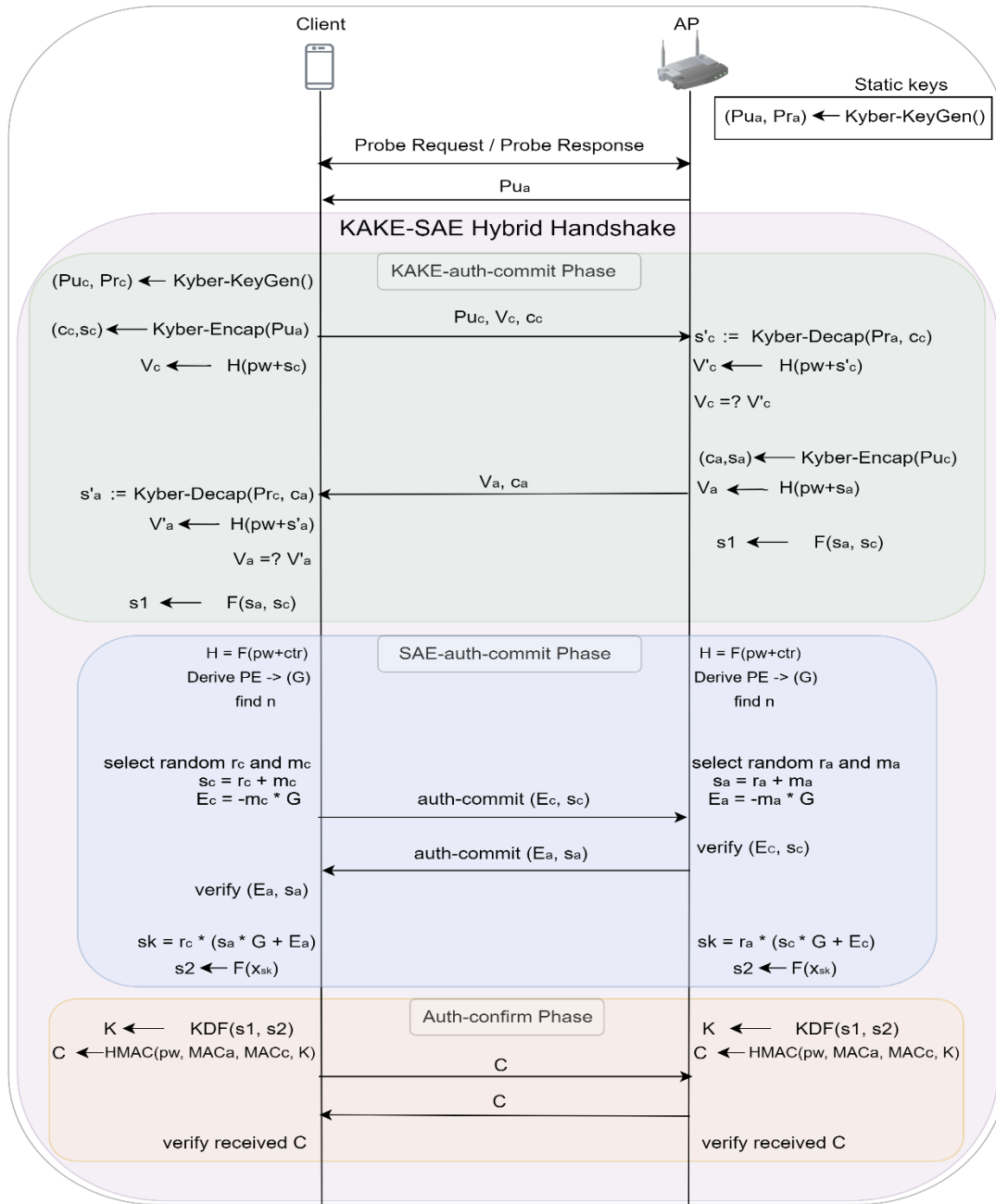


Figure 5.13: KAKE-SAE hybrid handshake

5.2.2.2 Advantages of the Hybrid Approach

The hybrid approach combining KAKE and SAE handshakes in WPA3 key exchange provides several advantages:

- Stronger security: The use of KAKE handshake provides a higher level of security against potential attacks by quantum computers. Combining it with SAE handshake ensures that the overall security

of the system is maintained while transitioning to post-quantum security.

- **Smooth transition:** The hybrid approach allows for a smoother transition to post-quantum security while still leveraging the established benefits of ECC. This reduces the need for an immediate overhaul of existing network infrastructure and eliminates the risk of a complete switch to a new algorithm that may have compatibility issues with current hardware.
- **Flexibility:** The use of a KDF allows for flexibility in choosing the desired key size and security level. This means that the hybrid approach can be customized to suit the specific security requirements of different applications and environments.
- **Compatibility:** The hybrid approach is compatible with existing WPA3 networks and can be easily integrated into new network setups, providing a more robust and secure wireless security solution.

5.2.2.3 Limitations of the Hybrid Approach

Here are some potential limitations of the KAKE-SAE hybrid approach in WPA3:

- **Complexity:** The implementation of the hybrid approach is more complex than just using a single algorithm for key exchange, which can increase the risk of implementation errors and security vulnerabilities.
- **Compatibility:** As a new and relatively untested algorithm, there may be compatibility issues with existing network infrastructure and devices, which could limit its widespread adoption.
- **Increased processing overhead:** The use of two algorithms can increase the processing overhead and impact network performance,

especially on low-powered devices with limited computational resources.

- **Key management:** Managing and storing multiple keys can be more complex and challenging than managing a single key, which can increase the risk of security breaches and key compromise.
- **Lack of standardization:** As the Kyber algorithm is not yet standardized by NIST, the long-term viability and compatibility of the hybrid approach may be uncertain.

5.2.3 TOFU Approach

In addition to the KAKE handshake, the proposed solution also incorporates the TOFU approach for additional security. TOFU is a method of establishing trust between two parties during their first interaction, by using their identities, such as public keys, without the need for a pre-existing trust relationship.

5.2.3.1 How TOFU Works?

In general, when two devices connect for the first time, they exchange cryptographic keys and store them locally. For subsequent connections, the devices use the previously exchanged keys to authenticate each other. As long as the keys have not been compromised, the devices can trust each other and proceed with the connection.

5.2.3.2 Alternative Approaches

some other methods for establishing trust:

- a. **PKI:** This method involves the use of a trusted third party, called a Certificate Authority (CA), to issue digital certificates that authenticate the identities of users and devices. The strengths of PKI are that it is widely used and can provide a high level of security. The weaknesses are that it can be complex to manage and may not

be suitable for resource-constrained devices (Prodanović, Vulić, & Tot, 2019). Additionally, it may not be feasible to deploy a centralized CA for wireless networks like WPA3.

- b. PSK: This method involves pre-sharing a secret key between the client and the AP. The strengths of PSK are that it is simple to implement and can provide a level of security if the key is sufficiently long and complex. The weaknesses are that it is susceptible to dictionary attacks and requires manual key management, which can be difficult to scale in large networks (Alhamry & Alomary, Exploring Wi-Fi WPA2-PSK protocol weaknesses, 2022). Additionally, once a key is compromised, it must be replaced on all devices, which can be time-consuming and disruptive.
- c. Multi-Factor Authentication (MFA): This approach requires users to provide two or more pieces of evidence to prove their identity, such as a password and a fingerprint scan. MFA provides an additional layer of security beyond a single factor (such as a password) and can prevent many common attacks, such as brute-force attacks. However, it can also be more cumbersome for users and may require additional hardware or software (Ometov, et al., 2018).
- d. Zero Trust: This method involves assuming that all devices and users are untrusted and requires authentication and authorization for every access attempt. The strengths of Zero Trust are that it can provide a high level of security and can be used in any environment. The weaknesses are that it can be complex to manage and may require additional infrastructure to support continuous authentication and authorization. Additionally, it may not be suitable for resource-constrained devices (Teerakanok, Uehara, & Inomata, 2021).

In the context of WPA3, PKI may not be suitable due to the complex infrastructure requirements and potential difficulty in managing large-scale deployments. PSK was used in previous Wi-Fi protocols, but it has weaknesses as previously discussed. MFA can provide an extra layer of security beyond a single password for authentication; however, it may require additional hardware or software and can be inconvenient for users. Zero Trust is an emerging approach that could be used in WPA3, but it may require additional infrastructure and may not be suitable for resource-constrained devices. Overall, the choice of trust establishment method for WPA3 will depend on a range of factors, including security requirements, cost, and ease of management.

5.2.3.3 Why TOFU?

TOFU is considered a better choice for WPA3 because it is simple and easy to use. It does not require any prior communication or exchange of certificates, making it a lighter and convenient approach. TOFU also provides a certain level of security against MitM attacks, as it allows users to compare the public key, they receive with the one they expect. This provides a certain level of assurance that the communication is with the intended party, and not with an attacker impersonating them. Additionally, the use of TOFU does not require any centralized authority or certificate management system, which can be a significant advantage in certain contexts where such systems are not feasible or practical. Overall, TOFU is a simple, lightweight, and effective approach for establishing trust in WPA3, making it a suitable choice for wireless security.

For a comparison of the advantages and disadvantages of each approach, please refer to Table 5.10.

Table 5.10: Comparison of trust establishment approaches

Trust Establishment Approach	Description	Strengths	Weaknesses
PKI	This method involves the use of a trusted third party, called a Certificate Authority (CA), to issue digital certificates that authenticate the identities of users and devices.	Widely used, can provide a high level of security.	Complex to manage, may not be suitable for resource-constrained devices. Not feasible to deploy a centralized CA for large-scale wireless networks like WPA3.
PSK	This method involves pre-sharing a secret key between the client and the AP.	Simple to implement, can provide a level of security if the key is sufficiently long and complex.	Susceptible to dictionary attacks, requires manual key management which can be difficult to scale in large networks. Once a key is compromised, it

			must be replaced on all devices, which can be time-consuming and disruptive.
MFA	This approach requires users to provide two or more pieces of evidence to prove their identity, such as a password and a fingerprint scan.	Provides an additional layer of security beyond a single factor (such as a password), can prevent many common attacks, such as brute-force attacks.	Can be more cumbersome for users, may require additional hardware or software.
Zero Trust	This method involves assuming that all devices and users are untrusted and requires authentication and authorization	Can provide a high level of security, can be used in any environment.	Can be complex to manage, may require additional infrastructure to support continuous authentication and authorization. May not be

	for every access attempt.		suitable for resource-constrained devices.
TOFU	This method involves trusting a public key the first time it is encountered and storing it for future use.	Simple and easy to use, no prior communication or exchange of certificates required, lightweight and convenient approach.	Provides a certain level of security against MitM attacks, but not as robust as other methods. Does not provide continuous authentication and authorization.

5.2.3.4 TOFU Approach Limitations

a list of some limitations of TOFU approach in the context of WPA3 implementation:

- Trusting a compromised device during the first connection: The TOFU approach assumes that the device used during the first connection is not compromised. A possible solution to this limitation is to perform additional security checks on the device before establishing the connection. This can include verifying the device's public key, such as in our approach, and running a security scan.
- False positives: TOFU can sometimes result in false positives where a legitimate device is incorrectly flagged as unauthorized. This can

lead to DoS attacks and can impact the user experience. A possible solution is to implement a fallback mechanism to allow users to manually verify the device's identity in case of false positives. In our case, a screen is shown to prompt the user to enter the network's password and public key.

- Limited protection against MitM attacks: While TOFU provides protection against passive attacks, it offers limited protection against active MitM attacks. One possible solution to this is to use a combination of TOFU and other authentication, such as KAKE handshake in our approach, to provide stronger protection against these attacks.
- No revocation mechanism: The TOFU approach does not provide a mechanism to revoke keys or identities of compromised devices. A possible solution to this is to implement a key revocation mechanism that allows for keys to be revoked in case of a compromise.

Overall, while TOFU has its limitations, it remains a useful approach for establishing trust in wireless networks, particularly in the context of resource-constrained devices like those used on the IoT (IoT). By combining TOFU with other authentication methods, such as KAKE handshake, and implementing additional security measures, it is possible to mitigate most of the limitations and provide stronger security for WPA3 implementations.

5.2.4 Integration of KAKE and TOFU in the Proposed

Solution

As we discussed before, the proposed solution integrates the KAKE handshake with the TOFU approach to provide an additional layer of security. During the first connection between two devices, and before they perform the KAKE handshake, the AP sends its public key to the client,

which stores it locally using the TOFU approach. For subsequent connections, the client uses the previously exchanged keys to authenticate the AP and proceed with the KAKE handshake.

5.2.4.1 Benefits of Integration

The integration of KAKE and TOFU in the proposed solution offers several benefits, such as:

- Improved security: The combination of KAKE and TOFU provides a strong authentication mechanism that ensures the security and privacy of wireless communications.
- Reduced vulnerability to attacks: The use of KAKE and TOFU can make it more challenging for attackers to impersonate legitimate devices or intercept wireless communications, thereby mitigating Dragonblood and Evil Twin attacks.
- Better usability: The combination of KAKE and TOFU provides a seamless and straightforward authentication mechanism that does not require the user to remember or enter any passwords after the initial connection.
- Additionally, the solution does not require any additional infrastructure, making it a cost-effective and efficient solution for securing wireless networks.

5.2.4.2 Limitations of Integration

While the integration of KAKE and TOFU in the proposed solution offers several benefits, there are also some limitations to consider, such as:

- Increased complexity: The integration of KAKE and TOFU may increase the complexity of the WPA3 implementation, which could potentially affect performance and efficiency.
- Lack of backward compatibility: The proposed solution is not backward compatible with previous versions of WPA, which means

that devices that do not support WPA3 will not be able to connect to the network.

- Relatively not user-friendly: The process of entering the password and parts of the hashed public key in the first connection with the AP can be challenging for some users.

5.3 Evaluation Results and Findings

In this section, we evaluate the technical, economic, and operational aspects of the proposed solution based on the results obtained from other recent research and the proof-of-concept approach conducted in this chapter.

5.3.1 Technical Feasibility

The proposed solution's technical feasibility was evaluated based on its compatibility, complexity, security, and performance aspects.

5.3.1.1 Compatibility

The proposed solution is not compatible with previous versions of WPA protocols, but a hybrid version of the solution is compatible with WPA3 protocol. This is achieved through the proposed version of the KAKE- SAE hybrid approach. However, it should be noted that this hybrid approach may still be vulnerable to security issues present in WPA3. To mitigate these vulnerabilities, one possible solution is to use separate passwords for each mode, although this could potentially increase network management complexity.

5.3.1.2 Complexity

The integration of KAKE and TOFU may increase the complexity of the WPA3 implementation, which could relatively affect performance and efficiency. However, the proposed solution does not require any additional

infrastructure, making it a cost-effective and efficient solution for securing wireless networks.

5.3.1.3 Security

The proposed solution provides a strong authentication mechanism that ensures the security and privacy of wireless communications. The use of KAKE and TOFU can make it more challenging for attackers to impersonate legitimate devices or intercept wireless communications, thereby mitigating Dragonblood and Evil Twin attacks. Furthermore, utilizing CRYSTALS-Kyber, a promising post-quantum cryptography candidate, in the KAKE handshake provides additional protection against future attacks from quantum computers.

5.3.1.4 Performance

Compared to other post-quantum key encapsulation mechanisms (KEMs) and even the Elliptic Curve Cryptography (ECC) algorithm, the Kyber key generation, encapsulation, and decapsulation processes are exceptionally fast. Moreover, the TOFU approach is very light and straightforward, which does not impact the performance of the proposed solution. However, the relatively large keys of Kyber may be challenging for some devices with very small memory.

5.3.2 Economic Feasibility

The economic feasibility of the proposed solution was evaluated based on its cost-effectiveness and affordability.

5.3.2.1 Cost-Effectiveness

The proposed solution does not require any additional infrastructure, making it a cost-effective solution for securing wireless networks.

5.3.2.2 Affordability

The proposed solution is affordable since it does not require any additional infrastructure, hardware, or software components. However, it may require additional training for network administrators and end-users to properly configure and use the new authentication mechanism.

5.3.3 Operational Feasibility

The operational feasibility of the proposed solution was evaluated based on its usability, scalability, and ease of implementation.

5.3.3.1 Usability

The combination of KAKE and TOFU provides a seamless and straightforward authentication mechanism that does not require the user to remember or enter any passwords after the initial connection. However, the process of entering the password and parts of the hashed public key in the first connection with the AP can be challenging for some users.

5.3.3.2 Scalability

The proposed solution can be easily scaled to support large-scale wireless networks since it does not require any additional hardware or software components.

5.3.3.3 Ease of Implementation

The proposed solution can be easily implemented since it does not require any additional infrastructure or hardware. However, it may require additional training for network administrators and end-users to properly configure and use the new authentication mechanism.

Overall, the results showed that the proposed solution is feasible and does not significantly impact network performance or efficiency, although

further evaluation and testing are necessary to determine its practicality and effectiveness in real-world deployments before implementing the solution.

Chapter 6: Conclusion, and Recommendation and Future Work

Chapter 6: Conclusion, and Recommendation and Future Work

6.1 Conclusion

Wireless networks have become an integral part of our daily lives, and as such, they are increasingly targeted by cybercriminals. The WPA3 protocol was designed to address some of the vulnerabilities found in previous versions of the WPA protocol and provide better security and privacy for wireless communications. However, as we have discussed in this thesis, there are still some weaknesses that could be exploited by attackers.

In this thesis, we proposed an integration of the KAKE handshake and the TOFU approach in the WPA3 implementation to improve the security and privacy of wireless communications. The proposed novel solution offers several benefits, including improved security, reduced vulnerability to attacks, post-quantum protection, better usability, and ease of implementation without the need for additional infrastructure. The proof-of-concept implementation showed that the proposed solution is feasible and does not significantly impact network performance or efficiency.

However, the proposed solution is not without limitations, such as lack of backward compatibility, and the relatively not user-friendly process of entering the password and part of the hashed public key during the first connection. Therefore, before adopting the proposed solution in real-world deployments, further evaluation and testing are necessary to determine its practicality and effectiveness.

In conclusion, the WPA3 protocol is a significant improvement over previous versions of the WPA protocol in terms of security and privacy for wireless communications. However, as wireless networks continue to evolve, so do the methods used by attackers to exploit vulnerabilities.

Therefore, it is essential to continue researching and developing new techniques and approaches to secure wireless networks effectively. The proposed integration of KAKE and TOFU in the WPA3 implementation is one such approach that shows promise, and it could be further enhanced and refined to provide even better security and privacy for wireless communications in the future.

6.2 Recommendation and Future Work

Based on the analysis and evaluation of the current state of wireless security and the proposed solutions in this research, several recommendations can be made to improve the security and privacy of wireless networks. It is recommended to adopt the latest security protocols like WPA3, use strong passwords and passphrases, regularly update firmware and software, educate users on security best practices, and develop security standards and best practices.

Future work could involve integrating other security mechanisms with the proposed solution, developing migration strategies for existing networks, integrating with enterprise-level security solutions, and conducting regular security audits. Additionally, future research could explore machine learning and AI techniques, blockchain technology, securing IoT devices, and exploring user-friendly authentication methods to enhance wireless network security. By addressing these research challenges, more robust and secure wireless networks can be developed to meet the increasing demand for reliable and secure wireless communication.

References

- Al Naamany, A. M., Al Shidhani, A., & Bourdouce, H. (2006). IEEE 802.11 wireless LAN security overview. *Ijcsns*, 6(5B), 138-156.
- Alagic, G. D. (2022). Status report on the third round of the NIST post-quantum cryptography standardization process. *US Department of Commerce, NIST*.
- Alagic, G. G.-S.-K. (2019). Status report on the first round of the NIST post-quantum cryptography standardization process. *NIST*.
- Alhamry, M., & Alomary, A. (2022). Exploring Wi-Fi WPA2-PSK protocol weaknesses. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 190-195). IEEE.
- Alhamry, M., & Elmedany, W. (2022). Exploring Wi-Fi WPA2 KRACK Vulnerability: A Review Paper. *2022 International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 766-772). IEEE.
- Anastasi, G., Conti, M., & Gregori, E. (2004). Ieee 802.11 ad hoc networks: Protocols, performance, and open issues. *Mobile Ad hoc networking*, 69-116.
- Appel, M., & Guenther, I. S. (2020). WPA 3-Improvements over WPA 2 or broken again? *Network* 7, 7-10.
- Bednarczyk, M., & Piotrowski, Z. (2019). Will WPA3 really provide Wi-Fi security at a higher level? *XII Conference on Reconnaissance and Electronic Warfare Systems. 11055*, pp. 369-376. SPIE.
- Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., . . . Stehlé, D. (2018). CRYSTALS-Kyber: a CCA-secure module-

- lattice-based KEM. *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 353-367). IEEE.
- Chatzoglou, E., Georgios, K., & Constantinos, K. (2022). How is your Wi-Fi connection today? DoS attacks on WPA3-SAE. *Journal of Information Security and Applications*.
- Coron, J.-S., Gérard, F., Montoya, S., & Zeitoun, R. (2021). High-order polynomial comparison and masking lattice-based encryption. *Cryptology ePrint Archive*.
- Cremers, C., Kiesel, B., & Medinger, N. (2020). A formal analysis of IEEE 802.11's WPA2: Countering the cracks caused by cracking the counters. *Proceedings of the 29th USENIX Security Symposium*. USENIX.
- Da Costa, V. L., Camponogara, Â., López, J., & Moisés, V. R. (2022). The Feasibility of the CRYSTALS-Kyber Scheme for Smart Metering Systems. *IEEE Access*, 131303-131317.
- Damir, M. T., Meskanen, T., Ramezani, S., & Niemi, V. (2022). A Beyond-5G Authentication and Key Agreement Protocol. *16th International Conference, NSS 2022* (pp. 249-264). Cham: Springer Nature Switzerland: Network and System Security (NSS).
- Gast, M. (2005). *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc.
- Ginkel, R. v., Poll, E., & Wiggers, T. (2019). Security in public Wi-Fi networks. Radboud University.
- Hamoudi, M., Korchi, A. B., Guilley, S., Takarabt, S., Karray, K., & Souissi, Y. (2021). Side-Channel Analysis of CRYSTALS-Kyber and A Novel Low-Cost Countermeasure. *Security and Privacy*:

- Second International Conference, ICSP 2021* (pp. 30-46). Jamshedpur, India: Springer International Publishing.
- Hanacek/NIST, N. (2022, July 5). *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms*. Retrieved from NIST: <https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>
- Housley, R., & Arbaugh, W. (2003). Security problems in 802.11-based networks. *Communications of the ACM*, 46(5), 31-34.
- Jati, A., Gupta, N., Chattopadhyay, A., & Sanadhya, S. K. (2021). A configurable Crystals-Kyber hardware implementation with side-channel protection. *Cryptology ePrint Archive*.
- Khasawneh, M., Kajman, I., Alkhudaiby, R., & Althubyani, A. (2014). A survey on Wi-Fi protocols: WPA and WPA2. *Recent Trends in Computer Networks and Distributed Systems Security: Second International Conference, SNDS 2014* (pp. 496-511). Trivandrum, India: Springer Berlin Heidelberg.
- Kohlilos, C. P., & Hayajneh, T. (2018). A comprehensive attack flow model and security analysis for Wi-Fi and WPA3. *Electronics*, 7(11), 284-312.
- Kumar, A., Ottaviani, C., Gill, S. S., & Buyya, R. (2022). Securing the future internet of things with post-quantum cryptography. *Security and Privacy*, 5(2), e200.
- LAGUEAGUE, S. E., & CHEIKHAOU, N. (2022). LIGHTWEGHT AUTHENTICATION SCHEME FOR PATIENT MONITORING BASED ON POST-QUANTUM CRYPTOGRAPHY. *Doctoral dissertation*. UNIVERSITY of M'SILA.

- Lashkari, A. H., Towhidi, F., & Hosseini, R. S. (2009). Wired equivalent privacy (WEP). *2009 International Conference on Future Computer and Communication* (pp. 492-495). IEEE.
- Marais, S., Coetzee, M., & Blauw, F. (2020). Simultaneous deauthentication of equals attack. *Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2020 International Workshops* (pp. 545-556). Nanjing, China: Springer International Publishing.
- Maruyama, N., Nukada, A., & Matsuoka, S. (2009). Software-based ECC for GPUs. *2009 Symposium on Application Accelerators in High Performance Computing (SAAHPC'09)*, 107.
- O'hara, B., & Petrick, A. (2005). *IEEE 802.11 handbook: a designer's companion*. IEEE Standards Association.
- Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y. (2018). Multi-factor authentication: A survey. *Cryptography*, 2(1), 1-31.
- Prodanović, R., Vulić, I., & Tot, I. (2019). A survey of PKI architecture. *KNOWLEDGE BASED SUSTAINABLE DEVELOPMENT*, 169.
- Sakib, A. N., Jaigirdar, F. T., Munim, M., & Akter, A. (2011). Security improvement of WPA 2 (Wi-Fi protected access 2). *IJEST*, 3(1), 723-729.
- Schwabe, P. (2020, December 23). *CRYSTALS Cryptographic Suite for Algebraic Lattices*. Retrieved from pq-crystals: <https://pq-crystals.org/kyber/>
- Septien-Hernandez, J.-A., Arellano-Vazquez, M., Contreras-Cruz, M. A., & Ramirez-Paredes, J.-P. (2022). A Comparative study of post-

- quantum cryptosystems for Internet-of-Things applications. *Sensors*, 22(2), 489.
- Singh, R. R., Moreira, J., Chothia, T., & Ryan, M. D. (2020). Modelling of 802.11 4-way handshake attacks and analysis of security properties. *Security and Trust Management: 16th International Workshop, STM 2020* (pp. 3-21). Guildford, UK: Springer International Publishing.
- Srinivasan, K., & Michell, S. (2004). Performance of state based key hop (SBKH) protocol for security on wireless networks. *IEEE 60th Vehicular Technology Conference, 2004 (VTC2004-Fall)*. 7, pp. 5214-5218. IEEE. Retrieved from http://www.wi-fi.org/files/wp_3_Securing%20Wi-Fi%20In%20The%20Enterprise_2-6-03.pdf
- Sun, S. (2022). A Chosen Random Value Attack on WPA3 SAE authentication protocol. *Digital Threats: Research and Practice (DTRAP)*, 1-8.
- Tasopoulos, G., Li, J., Fournaris, A. P., Zhao, R. K., Sakzad, A., & Steinfeld, R. (2022). Performance Evaluation of Post-Quantum TLS 1.3 on Resource-Constrained Embedded Systems. *Information Security Practice and Experience: 17th International Conference, ISPEC 2022* (pp. 432-451). Taipei, Taiwan: Springer International Publishing.
- Teerakanok, S., Uehara, T., & Inomata, A. (2021). Migrating to zero trust architecture: Reviews and challenges. *Security and Communication Networks*, 1-10.
- Vanhoef, M., & Piessens, F. (2018). Release the Kraken: new KRACKs in the 802.11 Standard. *Proceedings of the 2018 ACM SIGSAC*

- Conference on Computer and Communications Security* (pp. 299-314). Toronto, ON, Canada: ACM.
- Vanhoef, M., & Ronen, E. (2019). Dragonblood: A Security Analysis of WPA3's SAE Handshake. *IACR Cryptol. ePrint Arch*, 383.
- Vanhoef, M., & Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 517-533). IEEE.
- Walfield, N. H., & Koch, W. (2016). TOFU for OpenPGP. *Proceedings of the 9th European Workshop on System Security*, (pp. 1-6).
- Wolf, G., & Ortega-Arjona, J. L. (2022). A proposal for the survival of the OpenPGP decentralized trust network. *Proceedings of the 2022 ACM Conference on Information Technology for Social Good* (pp. 418-423). Limassol, Cyprus: Association for Computing Machinery (ACM).
- Xing, Y., & Shuguo, L. (2021). A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 328-356.
- Ye, P., & Yue, G. (2010). Security Research on WEP of WLAN. *Proceedings of the Second International Symposium on Networking and Network Security (ISNNS'10)* (pp. 2-4). PR China: Jinggangshan.
- Zhao, L., Zhang, J., Huang, J., Liu, Z., & Hancke, G. (2021). Efficient Implementation of Kyber on Mobile Devices. *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 506-513). IEEE.

المخلص

أصبحت شبكات Wi-Fi جزءًا لا يتجزأ من حياتنا اليومية، حيث يتم ربط مليارات الأجهزة بها. ومع ذلك، كان أمن شبكات Wi-Fi دائمًا مصدر قلق كبير بسبب الثغرات في بروتوكولات الأمان. بروتوكول الأمان الحالي WPA3 عرضة لأنواع مختلفة من الهجمات، بما في ذلك ثغرات "Dragonblood" والهجمات "Evil Twin" التي يمكن أن تعرض بيانات المستخدم الحساسة للخطر. وعلاوة على ذلك، فإن التشفير العام الذي يستخدم في WPA3 عرضة لهجمات الحواسيب الكمية، التي يمكن أن تصبح ذات جدوى في السنوات القادمة. لحل هذه المشكلات الأمنية، يجب أن يتم تطوير طريقة فعالة لتعزيز أمن شبكات Wi-Fi. تقترح هذه الرسالة الحل الجديد لعناوين بروتوكول Wi-Fi الأمنية من خلال دمج نهجين: "Kyber-based Authenticated Key Exchange (KAKE) handshake" والنهج "Trust On First Use (TOFU)". يوفر "KAKE handshake" الأمان ضد الثغرات في بروتوكولات WPA2 و WPA3 ويوفر حماية ما بعد الحواسيب الكمية. في الوقت نفسه، يتعامل نهج "TOFU" مع ثغرات في WPA3 عن طريق توفير أمان إضافي ضد هجمات "Evil Twin" والهجمات "Downgrade". تم تقييم الحل المقترح لأدائه وجدواه وأمانه من خلال الأدلة المفصلة ومقارنته بنتائج أدبيات أخرى. تظهر النتائج أن النهج المقترح فعال في منع مختلف هجمات Wi-Fi. بشكل عام، يمكن أن تكون هذه الرسالة قيمة للباحثين ومسؤولي الشبكات وأصحاب المصلحة الآخرين المهتمين.

جامعة البحرين
كلية تقنية المعلومات



تخفيف تهديدات أمن شبكات الواي فاي باستخدام
تطبيقات التشفير بالمفتاح العام

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير في الأمن الإلكتروني

إعداد

محمد علي محمد الحمري

٢٠٠٣٣٢١٨

إشراف

اد. علاء الدين العمري

(أستاذ مشارك)

جامعة البحرين

مملكة البحرين

يونيو ٢٠٢٣