KIPS Trans. Comp. and Comm. Sys.
Vol.12, No.6 pp.197~208
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2023.12.6.197

N-WPA2: Practical WPA2 Without Key Exchange of 4-way Handshake Using NFT Authentication

Tae-Young Eun† · Alshihri Saad†† · Soo-Yong Park†††

ABSTRACT

In the coming future, anyone using the Internet will have more than one NFT. Unlike FT, NFT can specify the owner, and tracking management is easier than FT. Even in the 2022 survey, WPA2 is the most widely used wireless protocol worldwide to date. As it is a protocol that came out in 2006, it is a protocol with various vulnerabilities at this time. In order to use WPA2-EAP or WPA3 (2018), which were released to compensate for the vulnerabilities of WPA2, additional equipment upgrades are required for STA (station) and AP (access point, router), which are connected devices. The use of expensive router equipment solves the security part, but it is economically inefficient to be introduced in Small Office Home Office (SOHO). This paper uses NFT as a means of authentication and uses the existing WPA2 as it is without equipment upgrade, defend crack tools of WPA2 that have been widely used so far and compared to the existing WPA2, it was shown that it was not difficult to actually use them in SOHO.

Keywords : :Blockchain, NFT, Wireless Lan, WPA2, Wi-Fi, Hacking, Security

NFT를 이용한 4-방향 핸드셰이크의 키 교환이 없는 실용적인 WPA2

은 태 영† · Alshihri Saad†† · 박 수 용†††

요 약

다가오는 미래에는 인터넷을 사용하는 사람이라면 누구나 NFT를 1개 이상 가지게 될 것이다. NFT는 FT와는 다르게 소유자를 명시할 수 있고, FT에 비해 추적관리도 용이하다. 2022년의 조사에서도 현재까지 전 세계적으로 가장 많이 사용되고 있는 무선 프로토콜은 WPA2이다. 2006년에 나온 프로토콜인 만큼 현시점에서는 다양한 취약점이 존재하는 프로토콜이다. 취약점을 보완하기 위해 2018년에 새로 나온 WPA3나 기존의 WPA2를 강화한 WPA2-EAP를 사용하기 위해선 접속하는 기기인 STA(스테이션)과 AP(액세스포인트, 공유기)에 추가적인 장비 업그레이드가 필요하다. 고가의 라우터 장비를 사용하면 보안적인 부분은 해결되지만 SOHO(Small Office Home Office)에서 도입하기엔 경제적인 비효율성이 있다. 본 논문에서는 NFT를 인증 수단으로 사용하여 기존의 WPA2를 그대로 사용하고 장비적인 업그레이드를 하지 않으면서 현재까지 널리 사용되고 있는 크랙 툴들을 방어하며 기존 WPA2와 비교해서도 실제로 SOHO에서 사용하는데 무리가 없음을 보였다.

키워드 : 블록체인, NFT, 무선 네트워크, WPA2, Wi-Fi, 해킹, 보안

1. 서 론

무선 LAN(Local Area Network, 근거리 통신망) 기술이란 일정 범위 내의 사용자로 하여금 어느 곳에서든 물리적인 연결 없이 네트워크에 접속이 가능하도록 하는 기술이다. 흔히 Wireless Lan이라고 하며 유선 LAN과 연결되어 있는 서버와 접속하기 위해 무선 AP(Access Point, 공유기)라는 장치를 사용한다. 가장 초기에 채택된 라우터 암호화 방식은 WEP이다. 1999년 Wi-Fi 보안 표준으로 채택되었고 이후 미국이 암호화 비트수 제한조치를 풀어서 192비트 WEP까지 등장하였다. 그러나 키 용량을 늘렸음에도 보안상 허점이 다수 존재했으며 2005년 미 연방 수사국(FBI)가 무료 소프트웨어를 통해 WEP 암호를 몇 분 안에 해독하는 과정을 시연하였다. 오늘날 WEP는 비영리 Wi-Fi 기술 인증기관인 Wi-Fi 연합(Wi-Fi Alliance)에서 2004년 공식적으로 WEP를 퇴출시켰다. Wi-Fi 연합은 WEP를 대체하기 위해 무선데이터 보호(Wi-Fi Protected Access, WPA)방식을 출시했으며 WPA는 2003년 공식 채택되었다. WPA는 256비트로써 기존 WEP의 64비트 및 128비트 보다 더욱 강력해졌으나 WPA의 핵심 구성요소인 TKIP이 WEP의 방식을 그대로 재활용함으로써 이는 결국 취약점으로 이어질 수밖에 없었다. 2006년 WPA는 무선데이터보호 II (Wi-Fi Protected Access II, WPA2) 방식으로 대체되었다. WPA2는 WPA와 달리 AES 알고리즘이 기본 적용되어 있으며CCMP(Counter Cipher

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터 지원사업의 연구결과로 수행되었음(IITP-2022-2017-0-01628*).
† 비 회 원 : 서강대학교 컴퓨터공학과 석사과정
†† 비 회 원 : 서강대학교 컴퓨터공학과 박사과정
††† 정 회 원 : 서강대학교 컴퓨터공학부 교수 및 지능형 블록체인센터장
Manuscript Received : October 31, 2022
Accepted : December 22, 2022
* Corresponding Author : Soo-Yong Park(sypark@sogang.ac.kr)

Mode with Block Chaining Message Authentication Code Protocol) 방식이 TKIP을 대체한다. WPA2의 문제점과 취약점은 제2.2.4 장에서 다룬다. WPA2의 문제점에 대응하기 위해 Wi-Fi 연합은 2018년 WPA3 보안 프로토콜을 발표했다. WPA2의 근본적인 단점은 불완전한 4-way handshake이며, PSK(PreSharedKey, 사전 공유키) 사용 시 Wi-Fi 연결을 위험에 노출시킨다. WPA3는 암호 키 추측을 통해 연결 과정에 침입하는 것을 어렵도록 추가적인 보안을 구현하였다. 대표적으로 WPA2의 PSK를 장치 간 동시 인증(SAE, Simultaneous Authentication of Equals)로 대체하여 WPA2의 가장 취약한 크랙 툴인 KRACK(Key Reinstallation Attack)으로부터 공격을 보호한다. 하지만 접속하는 기기인 STA(스테이션)과 AP(액세스포인트, 공유기)가 SAE를 지원해야한다. 최신의 스마트폰들은 WPA3를 지원하지만 아직까지 사용하고 있는 대다수의 공유기에서 WPA3를 지원하지 않는다. 이는 소프트웨어적으로 가능한 것이 아닌 하드웨어적으로 업그레이드 되어야 한다. 라우터 회사들은 AP의 SAE 기능을 2019년부터 출시되는 모델에 순차적으로 적용하여 출시하고 있다.

Wigle.net 조사에 의하면 2022년에도 전 세계에서 가장 많이 사용하는 프로토콜은 WPA2이다(Fig. 1).

2006년에 당시에는 엄연히 좋은 성능의 프로토콜이었으나 현 시점에 와서는 3만원도 안 되는 모니터링 무선 장비를 구입해서 누구나 빠른 시간 내에 크랙 시도가 가능하다[1]. 공격 방법으론 WPS를 이용한 Pixie Dust 크랙, KRACK, 사전 암호 예상 파일을 통한 키 유도 공격, WPA2 Handshakes 패킷 Capturing 분석 등이 널리 쓰이고 있다. WPA2의 사용이 위험한 예시는 공공장소에서 나타나는데, 해커들이 카페나 공항 같은 공공장소 근처에 차를 타고 돌아다니면서 원거리에서 무선 AP를 해킹하여 개인정보를 수집해가는 Rogue War Driving이 활개치고 있다[2]. 실제로 이러한 툴을 가지고 해킹이 얼마나 쉽게 이루어지는지 학술대회에 발표한 저자의 논문이 있다[3].

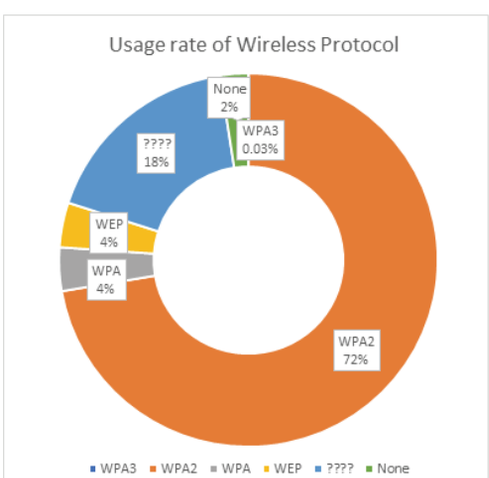

Fig. 1. Usage rate of Wireless Protocol in 2022

그러므로 아직까지는 많이 사용하고 있지만 현재 WPA2의 이러한 보안 실태를 고려했을 때, 저자는 WPA2는 그대로 쓰고 장비적인 업그레이드는 하지 않으면서 기존 해킹 툴로부터 안전하며 기존 WPA2와 비교해보아도 일반적인 SOHO Network(Small Office Home Office, 이하 SOHO) 환경에서도 실질적으로 사용하기에 무리가 없는 실용성을 강조한 N-WPA2를 제안하고자 한다.

2. 배경 지식과 WPA2 해킹 및 보안 이론

2.1 블록체인과 NFT(Non-Fungible Token)

블록체인은 P2P네트워크를 통해서 관리되는 분산 데이터베이스의 한 형태로 거래 정보를 담은 장부를 중앙 서버 한곳에 저장하는 것이 아닌 블록체인에 연결된 모든 컴퓨터에 저장 보관하는 기술이다. 2008년에 S.Nakamoto의 논문[4] "Bitcoin: A Peer-to-Peer Electronic Cash System"으로 블록체인의 기술이 세상에 많이 알려지게 되었고 그는 비트코인을 개발하면서 발생하는 문제점들을 블록체인을 개발 및 적용함으로써 해결했다. 블록에는 이전 사용자들이 거래했던 트랜잭션이 모두 기록되어 있고, 이것은 P2P 방식(Fig. 2)으로 모든 사용자에게 똑같이 전파되어 거래 내역을 한 특정 개인이 임의로 수정하거나 삭제할 수 없다.

각 블록은 발견된 날짜와 이전 블록의 해시값에 대해 연결고리를 가지고 있으므로 이러한 블록의 집합을 블록체인이라 한다. 쉽게 표현하면 거래장부를 블록, 그것들을 연결한 것을 체인, 합쳐서 블록체인이라고 보면 된다. 비트코인과 같은 퍼블릭 블록체인에서 블록의 트랜잭션이 위조되지 않았다는 것을 검증하기 위해 Proof-of-Work을 도입했다. PoW(지분증명)란, 다음 블록에 올 해시값을 계산하기위해 비트코인 네트워크 시스템이 정한 난이도 이하의 값이 될 때까지 블록의 헤더 Nonce를 바꿔서 맞춰가는 과정이다. 10분에 한 번씩 블록이 생성되며 이렇게 6번을 맞춘 것을 비트코인에서는 6 Confirm Finalizing이라고 한다. 이것이 이루어지면 이 논문을 쓰는 현시점의 컴퓨팅 파워로는 이전 블록들의 값을 변조시킬 수 없다.

이더리움[5]은 2009년의 비트코인 논문에 영감을 얻어 2015년 비탈릭 부테린이 창안한 또 다른 퍼블릭 블록체인 플랫폼이자 플랫폼 자체 통화 이름이다. 비트코인과의 가장 큰 차이점은 Smart Contract의 도입으로 인한 다양한 응용성

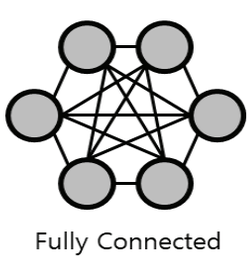

Fig. 2. Network Structure of Bitcoin

Table 1. ERC-721 Standard

Function Name	Attributes
balanceOf	Returns the number of NFT Owner's owned
ownerOf	Returns the owner address of an NFT with a specific toeknID
approve	Allow specific accounts to use one NFT onwer's own
getApproved	Returns whether certain NFTs have been authorized for use by other accounts
setApprovalForAll	Allow specific accounts to use all NFT Owner's own
isApprovedForAll	Returns whether the owner has allowed a particular account to use it for all of NFTs
transferFrom	NFT Ownership Transfer
safeTransferFrom	Send NFT ownership after confirming that the receiving address can receive NFT

이다. 즉, 비트코인이 결제나 거래 관련 시스템, 화폐로써의 기능에만 집중하는 반면, 이더리움은 Smart Contract를 통해 거래나 결제뿐 아니라 계약서, 전자투표, DAO 등 다양한 탈중앙화 분산 애플리케이션(DApp)을 누구나 만들고 사용할 수 있게끔 하는 플랫폼이다. 이러한 점 때문에 비트코인부터 이더리움 이전까지는 1세대 블록체인이라 불리고 이더리움부터는 2세대라고 불린다. 이더리움의 Smart Contract를 활용한 본 논문의 주제인 NFT(ERC-721)는 다음 표와 같은 표준을 가지고 있다(Table 1).

2020년부터 2022년까지 COVID-19로 인한 NFT의 폭발적인 붐, 스캠, 해킹 등의 여러 문제들이 많이 발생하였지만 블록체인과 NFT의 자체의 문제는 없었고 대부분 유저의 개인정보 유출이나 NFTM(NFT-Market)의 문제였다. 저자는 NFT의 그 고유한 원본성, 소유권 및 신분증으로써의 역할을 인증 수단으로 이용하려고 한다.

2.2 ERC-721과 디지털 서명

1) ERC-721

Ethereum Request for Comment(ERC)란 Internet Engineering Task Force(IETF)에서 제정하는 인터넷 표준 절차인 RFC의 형식을 따온 이더리움 플랫폼만의 절차(프로토콜)이다. 이더리움에서 거래 토큰으로 사용되는 ERC-20은 Smart Contract로 구현할 수 있는 API용 토큰 표준으로 Fabian Vogelsteller에 의해 제안되었으며 토큰으로써의 역할과 기능을 정의한다. ERC-20은 대체 가능이라는 의미로 사용되는 Fungible Token인데 '대체 가능'의 의미는 화폐를 떠올리면 된다. 저자가 가진 100원과 A라는 사람이 가진 100원은 둘 다 동일한 가치를 지닌다. 이것을 누군가에게 주어도 마찬가지이다. ERC-721은 이더리움 플랫폼에서 대체할 수 없거나 고유한 토큰(Non-Fungible Token)을 작성하는 방법을 설명하는 표준 규약(프로토콜)이다. ERC-721 토큰은 ERC-20 토큰과는 반대로 '대체 불가능'한 특징을 지닌다. 다시 말해 ERC-721로 발행되는 토큰은 모두 각각의 고유한 가치(Value)를 갖고 있다.

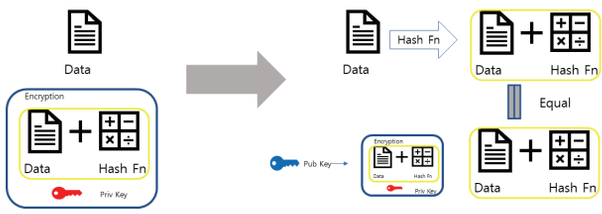

Fig. 3. Signing and Verification

2) 디지털 서명

현재 대부분의 구글이나 네이버 같은 잘 알려진 웹사이트에 접속할 때, 클라이언트의 브라우저와 서버 측에서는 암호화 통신을 위해 SSL인증을 사용하는데 이때 사용하는 것이 전자서명이다. 하지만, 블록체인과 같은 Web3 환경에서는 디지털 서명과 코드 서명을 사용한다. 코드 서명부터 언급하면 응용프로그램에 전자 서명을 해서 제대로 된 인증절차를 거쳐 배포할 수 있게 하는 기술이다. 디지털 서명은 서명과 검증으로 표현할 수 있다. 블록체인에서 지갑을 이용한 서명을 할 때(데이터를 보낼 때) 데이터와 데이터를 해시함수 돌린 것에 지갑의 개인키로 암호화 한 것 2개를 검증자에게 같이 보낸다(Fig. 3).

데이터를 받은 검증자는 받은 데이터를 해시함수를 돌리고 암호화된 채로 받은 데이터를 알려진 공개키로 복호화 해서 그 두개가 일치하는지 검증한다. 전자를 서명이라 하고 후자를 검증이라고 한다. 이러한 서명과 검증을 통해 Smart Contract에서 자격이 있는 자가 값(변수)을 호출할 수도 있고 변경(Update)할 수도 있다. 제2.2.4 장에서 설명하겠지만 결론적으로 WPA2는 통신에 필요한 많은 Key와 그 정보들이 공기 중에 Signal로써 전파되기 때문에 문제가 발생한다. 디지털 서명에서는 데이터를 암호화하고 복호화하는 개인키(Private Key)가 전송되지 않는 점이 WPA2 통신과의 차이점이고 핵심이다.

2.3 OpenWRT on SBC(Single Board Computer)

1) 고가의 라우터 장비

제2.2.4 장에 나올 대다수의 보안 취약점들은 보안 회사의 고가의 라우터 장비를 사용하면 많은 부분을 해결할 수 있다. 대표적으로 보안 회사인 C사의 라우터 장비가 유명한데, 일반적인 허브나 공유기 말고 보안 기능이 들어간 라우터의 경우 적게는 한화 50만원부터 고급 제품의 경우는 300만원 이상하는 제품이 많다. 2022년 Cyber Defense Magazine의 조사에 의하면 전 세계 Cyber Attack 중 전체의 43%가 SOHO 환경을 대상으로 하고 있다고 발표했다. SOHO를 쉽게 말하면 가정, 중소기업으로 표현할 수 있다. 가정이나 중소기업에서는 사실상 고가의 라우터 장비가 필요하지도, 구

입할 여력도 되지 않는다. 그렇다고 해서 WPA2를 그대로 사용하면 제2.2.4 장에 언급하는 많은 취약점이 노출된다. 한마디로 보안회사의 고가의 라우터 장비는 SOHO Network에서 실용적, 합리적이지 않다.

2) SBC(Single Board Computer)

SBC는 컴퓨터 기능에 필수적인 마이크로프로세서, 메모리, 입출력 등의 기능이 있는 단일 회로 기판으로 구성된 완전한 컴퓨터이며 초소형 크기와 저전력이라는 특징을 가진다. 2022년에 SBC 회사는 중국 회사까지 합치면 정말 무수히 많아졌고, 국내에도 H라는 회사에서 국산 SBC를 제작 및 판매하고 있다. SBC 마다 성능, 가격이 다양하지만 저자가 논문에서 사용할 것은 가장 범용적이고 교육용으로도 많이 사용되는 라즈베리파이(Raspberry Pi) 4B 모델이다. 컴퓨터 보급이 부족한 제3 국가에서 교육용으로 많이 쓰이던 라즈베리파이가 4B모델에 와서는 그 성능이 arm으로 포팅된 윈도우11까지 구동할 수 있다. 영국의 라즈베리파이 재단에서 출시한 라즈베리파이 4B는 $55로 출시되었지만 2020년부터 계속된 반도체 부족사태 때문에 현재 국내에서는 배송비 포함 20만원 안팎에 구매할 수 있다. 그럼에도 불구하고 제2.2.3-1 장에 언급한 고가의 보안 회사 장비들 보다는 당연히 경제적이다. 더 저가의 다른 SBC도 있지만 저자가 논문에서 사용할 Python과 Node.JS를 사용하기 위해선 aarch64 아키텍처 이상의 성능을 가진 마이크로프로세서가 탑재된 SBC가 필요했기에 라즈베리파이 4B를 선택했다.

3) OpenWRT(Open Wireless RouTer)

공유기나 라우터 장비에는 그 기기에 맞는 Firmware 나 Embedded OS가 설치되어 있다. 각각의 제품 회사들 마다 고유의 Firmware가 기본적으로 설치되어 있고 해당 Firmware의 관리자 모드에 들어가서 Wi-Fi의 비밀번호를 바꾸거나 인터넷 이용시간 제한 등의 기능 설정을 할 수 있다. 저자의 논문은 NFT의 소유 인증을 통과한 후 WPA2를 이용하게 하는 것인데, 이렇게 특별한 순서가 정해진 Process를 진행하기 위해선 라우터 회사들의 기본 Firmware로는 구현할 수 없었다. 또, 블록체인 네트워크에서 NFT 인증을 하려면 최소한 외부와 통신 가능하며 JavaScript(JS) 정도는 돌아가는 소형 서버가 필요했는데 이러한 Process Flow와 장비추가의 문제를 한번에 해결해 주는 것이 바로 리눅스 기반의 오픈소스 프로젝트인 OpenWRT이다. 리눅스 기반의 Firmware라서 개발자가 설정을 다루기에도 용이하다. 용도가 공유기로만 제한되지는 않으며, 라즈베리파이, 포고플러그 등 각종 임베디드 장비 혹은 x86 머신에도 설치가 가능하다. 즉, 이것을 이용하면 유무선 공유기가 되면서 동시에 소형 리눅스 임베디드 장치가 될 수 있다. 정리하자면 저자는 보안 회사의 고가의 라우터 장비를 사용하는 것 대신, 상대적으로 저렴한 교육용 Single Board Computer인 라즈베리파이에 OpenWRT를 설치하여 이것을 유무선 공유기이자 NFT 인증 서버로 사용할 것이다.

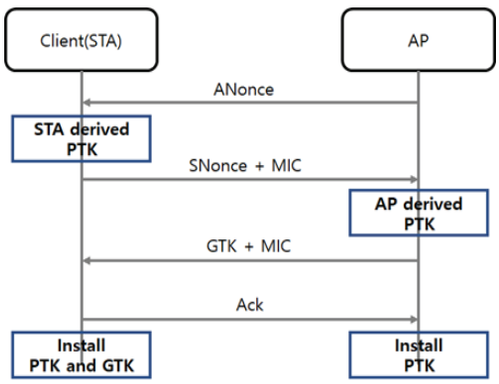

Fig. 4. EAP 4-way Handshake

$$(1)\ PTK \leftarrow PMK(PSK)+ANonce+SNonce+MAC(AP)+MAC(STA)$$

Fig. 5. Parameter for PTK

2.4 WPA2 해킹 이론

1) EAP 4-way handshake

WPA2는 EAPOL stands for Extensible Authentication Protocol(EAP) over LAN 이라는 간단한 4-way handshake를 사용한다. IEEE의 표준 그림이 설명하기에 복잡하므로 저자의 간략화된 그림으로 WPA2의 연결과정을 설명한다(Fig. 4).

최종적으로 WPA2의 암호화된 통신을 하기 위해선 접속하는 기기 STA(스테이션)과 AP(공유기)에 PTK(Pairwise Transient Key)가 설치되어야 한다. 이 PTK를 derive(유도)하기 위해선 그 과정에서 많은 Key가 사용되지만 핵심적으로 5개의 인자가 필요하다(Fig. 5).

Fig. 4를 순서대로 파악해보면 접속하려는 기기 STA은 AP로부터 AP의 Nonce인 ANonce를 받는다. PMK(PSK)는 사전에 공유된 키이므로 STA, AP 둘 다 알고 있다. ANonce를 받은 STA 입장에서는 Fig. 5에 의거해 PTK를 유도할 5가지 인자가 다 갖추어졌다. 다음으로는 STA에서 AP에게 SNonce와 MIC를 보낸다. Message Integration Code의 약자로써 STA이 자신이 보냈다는 것을 증명하는 메시지 코드이다. SNonce를 받은 AP 입장에서도 PTK를 유도할 인자가 다 갖추어졌다. AP도 STA에게 자신이 보냈다는 것을 증명하는 MIC와 자신의 그룹 일원이 되라는 GTK(Group Temporary Key)를 보낸다. STA은 AP의 MIC를 확인하고 올바른 값이므로 PTK와 GTK를 install한다. install 이후 올바르게 수행되었다는 ACK를 AP에게 보냄으로써 AP도 PTK를 install하여 서로 간에 암호화 통신을 할 준비가 되었다. 이 통신 방법에는 대표적으로 3가지 크랙 Tool이 존재하고, Brutal Force까지 이용하면 크게 4가지의 취약점이 존재한다.

2) PMK(PreShareKey) Attack

첫 번째로 Airmon-ng[6]라는 Tool을 이용해 PMK(PSK, 사전 공유키)의 취약점을 이용하는 것이다. PreSharedKey는 말그대로 사전 공유키 인데, 만드는 방법은 3~4 가지가

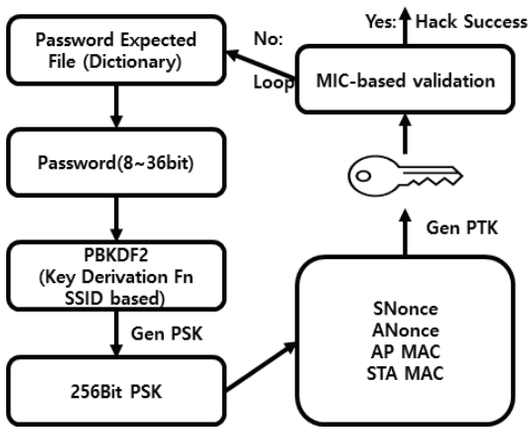

Fig. 6. Principle of PMK(PSK) Crack

존재하며 구형 공유기일수록 펌웨어 패치가 안 되어 있다면 더더욱 쉽게 크랙이 가능하다.

Fig. 6과 같이 잘 만들어진 사전 암호 예상 파일이 있고, 그 파일에서 8~36bit를 추출해서 PBKDF2라는 SSID 기반 키유도 함수를 돌리면 PSK를 Generate 할 수 있다. 이것을 토대로 다시 PMK를 만들기 위해 5가지 인자를 모두 넣어서 PTK를 Generate하고 MIC 기반으로 AP에게 검증을 한다. 과정이 실패해도 다시 Loop를 돌려서 될 때까지 하면 된다. 2022년 현재 하드웨어의 성능은 굉장하기 때문에 사전 암호 파일만 잘 갖추어 진다면 크랙 하는데 20분 이내로 가능하다.

3) Pixie Dust(WPS) Attack

두 번째로는 WPS를 악용한 Pixie Dust Attack[7]이 있다. WPS(Wi-Fi Protected Service) 버튼의 원래 목적은 공유기에 설정된 별도의 Wi-Fi 비밀번호를 입력하지 않고 간단하게 버튼을 누름으로써 손쉽게 연결할 수 있는 편리함을 제공하는 기능이다. 하지만 공유기 제조사 마다 제품에 내장된 칩셋의 PIN Number가 유출되었고 이것을 악용하여 더 쉽게 크랙을 시도할 수 있다.

Fig. 7과 같이 총 13번의 EAP-POL 메시지 전송과정에서 AP가 답변하는 M3와 M5가 문제이다. M2에서 잘못된 메시지를 보내면 M3에서 그것이 틀렸다는 NACK를 보내준다. 이를 통해서 크랙 해야 할 경우의 수가 급격하게 줄어든다. 또, M4에서도 틀려도 M5에서 NACK를 보내주기 때문에 크랙 해야 할 경우의 수가 두 번이나 급격하게 줄어들기 때문에 PIN Number에 대한 정보까지 있다면 공격이 매우 쉽다. 유출된 정보가 없더라도 WPS의 PIN 번호는 8자리 숫자이다. WPS attack은 기본적으로 지능형 Brutal Force 공격인데 AP가 응답하는 시간을 고려하면 1초당 1~2개의 응답을 받을 수 있다. Brutal Force로 진행하면 8자리와 자리당 10개의 숫자(0-9)는 10^8 초라는 계산을 가져온다. 그냥 시도하면 약 3년 이상의 시간이 걸리지만, PIN Number에는 8번째 자리가 Checksum이라는 규칙이 있다. 또한 이 8자리는 사실 앞 4자리와 마지막 4자리를 독립적으로 확인할 수 있도록

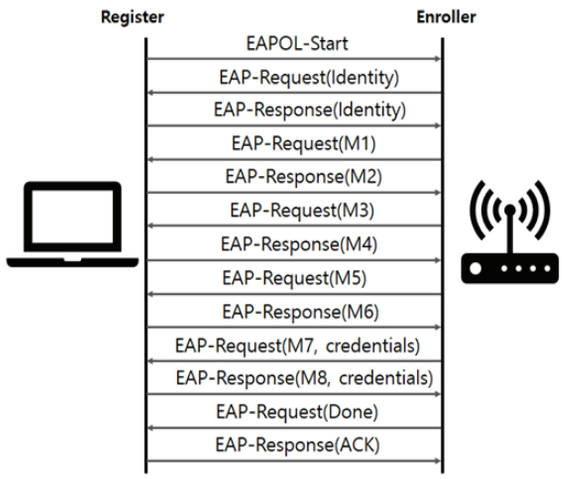

Fig. 7. WPS Message Exchange

$$(2)\quad Number\ of\ WPS\ guesses = 10^4 + 10^3$$

Fig. 8. Number of WPS Crack Guesses

나누어져 있다. 즉 기본적으로 전반부 4자리는 10^4 초의 시간이 필요하고 후반부는 10^3 초가 소요된다.

이 두개의 추측 횟수 합은 Fig. 8로 나타낼 수 있다. 즉 11,000의 추측이 필요하고 이것은 약 3시간 이내로 압축할 수 있다. 대표적으로 reaver와 Wifite라는 Tool을 이용해 크랙 한다.

4) KRACK(Key Reinstallation Attack) Attack

마지막으로는 WPA2에서 가장 악명높은 공격인 KRACK이다. 이 취약점은 WPA2 프로토콜 4-way handshake의 논리적 결함으로 인해 발생하는 것으로 Wi-Fi를 지원하는 거의 모든 디바이스들(Android, Linux, Apple, Windows, Mediatek, Linksys, OpenBSD)이 영향을 받는다. 이 취약점은 주로 802.11i 중의 4-way handshake 과정 중 언제 협상 키를 설정해야 하는지 정의되어 있지 않아, 공격자가 동일한 키를 여러 번 설치하여(재설치) 암호화 프로토콜이 사용되는 난수(Nonce) 및 재생계수를 재설정하는 공격이다. 이 취약점에 대하여 공개적으로 알려진 컴퓨터 보안 결함 목록을 다루는 CVE(Common Vulnerabilities and Exposures) 공식 사이트에 2017년에만 10개의 CVE 항목(Table 2)이 업데이트 되었다. 이러한 취약성들로 인해서 2018년에 Wi-Fi 연합은 KRACK을 막는 WPA3를 발표하였다.

Fig. 4와 같이 EAP-POL 4-way handshake 메시지 교환 과정에서 AP가 답변하는 3 번째 답변인 M3가 주로 문제이다. 클라이언트(STA)는 M3를 수신 후 비밀 Key를 설치하여 정상 데이터 프레임을 암호화하는데 사용한다. 그런데 Message가 중간에 소실될 가능성이 있기 때문에 일정 시간내에 M4(Ack)가 오지 않는다면 AP는 M3를 다시 전달하게 된다. 이렇게 되면 클라이언트는 여러 번의 M3를 받을 가능성이 있고 클라

Table 2. CVE List about KRACK

CVE	No.	Vulnerabilities
CVE-2017-	13077	Reinstallation of the pairwise encryption key(PTK-TK) in the 4-way handshake.
	13078	Reinstallation of the group key(GTK) in the 4-way handshake.
	13079	Reinstallation of the integrity group key(IGTK) in the 4-way handshake.
	13080	Reinstallation of the group key(GTK) in the group key handshake.
	13081	Reinstallation of the integrity group key(IGTK) in the group key handshake.
	13082	Accepting a retransmitted Fast BSS Transition(FT) Reassociation Request and reinstalling the pairwise encryption key(PTK-TK) while processing it.
	13084	Reinstallation of the STK key in the PeerKey handshake.
	13086	Reinstallation of the Tunneled Direct-Link Setup(TDLS) PeerKey(TPK) key in the TDLS handshake.
	13087	Reinstallation of the group key(GTK) when processing a Wireless Network Management(WNM) Sleep Mode Response frame.
	13088	Reinstallation of the integrity group key(IGTK) when processing a Wireless Network Management(WNM) Sleep Mode Response frame.

이언트는 매번 M3를 받을 때 마다 암호 Key를 재설정할 기회가 생긴다. 또한, M3를 다시 받을 때마다 데이터 패킷의 Nonce와 수신한 재생 Counter도 다시 재설정할 기회가 생긴다. 때문에 KRACK이 성공한다면 최악의 경우 30초 이내로 크랙이 가능하고 Wi-Fi 사용 중에 신용카드, 이메일 계정 정보 같은 다양한 정보들을 탈취당할 수 있다.

5) 4-way handshake Capturing

추가적으로 Wireshark 같은 패킷 탐지 툴을 이용해서 WPA2 Handshake의 전송과정을 Capture하고 그 Capture한 파일(4패킷)을 그래픽카드 같은 장비를 이용해 Brutal Force로 크랙 하는 방법이 있다.

2.5 WPA2 보안 방법

위에 언급한 WPA2의 취약성에 대해 방어하는 다른 사람들의 논문들은 라우터에 내장된 키 알고리즘을 변경하고[8] 전달 과정에서의 난수(Nonce)와 Key들을 추론하기 어렵게 하도록 변경 또는 분리하는 방법[9]이 대표적이다. Key나 알고리즘의 변경만으로는 한계가 있어서 신경망 모델인 ReLU 은닉층 모델과 Sigmoid 모델 등을 통해 침입을 미리 탐지하는 방법[14]도 사용한다. 추가적인 논문으로는 STA마다 반도체의 고유한 값(PuF, Physical Unclonable Function)을 서명으로 사용해서[15] 중간자 공격이나 KRACK을 방지하는 방법도 있으며 블록체인 중 비트코인 플랫폼을 이용하여 지갑 주소별 사용자 인증을 통해 보안을 강화[16]하는 제안이 있다. 이러한 WPA2 보안에 대한 다른 논문 저자들의 내용은 요약해보면 전송과정의 Key나 알고리즘을 바꾸어 최종 Key인 PTK의 유도와 유추를 어렵게 한다. 그러나 이 방법은 하드웨어 성능이 진보한 현 시점에서는 시간과 장비를 많이 추가한다면 금방 Break될 Solution이다. 그리고 크랙으로부터 보호하기 위해 인증을 거친 후 통신을 하는 방법을 제시하는 논문들에서도 반도체 공장에서 받아와야하는 PuF값이라든지, WPA2의 암호화 알고리즘을 변경하는 것은 실용적이지 않고 일반 사용자들이 손쉽게 쓸 수 있는 방법은 아니다.

1) PMK(PSK) 보안 방법

첫 번째로 유출된 칩셋의 PMK의 사용을 금지하고 주기적인 Firmware 패치를 통해 지속적으로 업데이트해서 사전 암호화된 예상파일의 크랙 경우의 수에서 벗어나야 한다.

2) Pixeie Dust(WPS) 보안 방법

두 번째 방법인 Pixie Dust을 방지하는 방법은 아주 간단하게는 WPS 푸시 버튼을 쓰지 않게 설정하면 된다. 하지만 대부분의 라우터 회사에서 출고되는 제품에는 기본적으로 WPS가 On이 된 상태를 Default 값으로 제품을 출시한다

3) KRACK 보안 방법

마지막으로 KRACK은 WPA2의 논리적 취약점을 이용한 것이기 때문에 WPA2의 알고리즘을 바꾸거나[8] 전송과정의 Key 값을 분해, 분리[9] 등을 통해 바꿔주는 형식으로 PTK를 유도하는 Key들이 Reinstallation이 안 되도록 해야한다. KRACK은 조건(Condition)과 Key 재설치 경우의 수가 매우 다양하기 때문에 Firmware 보안 패치만으로는 완벽하게 대응하기 어렵다.

3. 본론 : NFT 인증을 통한 WPA2 사용

저자의 아이디어는 본론 이전의 장에 나온 모든 것을 직접 해보면서 발견했다. 크랙 툴들을 다뤄보면서 방어법을 생각하고 다양한 논문을 읽어 보던 도중, 특히 WPA3[13]의 논문에서 사용된 장치 간 동시 인증(SAE)을 NFT 인증으로 바꿔야겠다는 결론에 도달했다.

3.1 NFT 인증과 제안 방법 소개

저자는 NFT를 인증 수단으로 사용하기 위해 크게 2가지 개념을 사용하였다. 첫 번째로 Smart Contract에서 주로 사용하는 Solidity 언어에서 mapping 타입을 array로 선언하여 그것을 Whitelist 개념으로 사용하였다. 이더리움에는 수

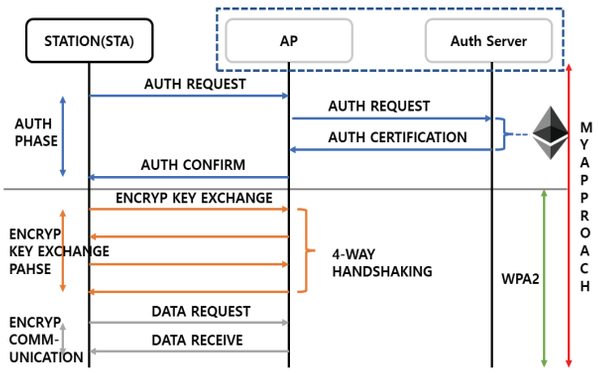

Fig. 9. N-WPA2 Process Flow

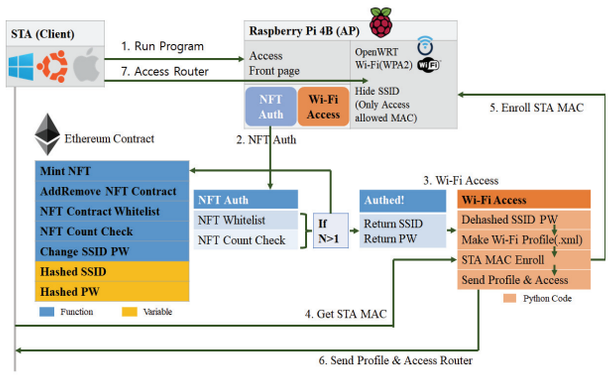

Fig. 10. N-WPA2 Program Flow

많은 NFT가 있고 각각의 NFT는 Smart Contract이므로 Contract Address가 있다. 이 제안을 사용하는 자가 인증에 사용될 NFT Contract Address를 지정하기 위해선 Whitelist 형식으로 NFT Contract Address를 추가하거나 제거할 수 있도록 하기 위해서이다. 두 번째로 사용한 개념은 Blockchain Event Monitoring이다. 이것은 모니터링 하려는 블록체인의 특정 블록으로부터 5~6개씩 최신 블록을 수신하여 그 안의 블록에서 해당 Contract의 변화가 있으면 그 변화를 출력한다. 예를 들면, Whitelist에 등록된 NFT를 1개 가진 A가 인증을 한 후, B에게 NFT를 전송하여 NFT가 0개가 되었다면 A의 자격을 회수해야한다. 이때 블록체인 이벤트 모니터링을 하고 있다면 A가 B에게 NFT를 전송한 것을 알 수 있고, 그러한 전송이 이루어져서 A의 NFT가 0개가 되었다는 이벤트가 확인되면 A의 권한을 회수하는 코드를 작성하면 된다. 제안하는 Approach의 이름은 N-WPA2이다. Fig. 9는 본인 Approach의 Flow Sequence이다. 그림에서 우측 상단에 점선 박스로 묶어둔 AP와 Auth Server는 제 2.2.3-2 장에서 언급한 한 개의 기기에서 같이 동작한다. 가운데 회색 선으로 반을 나눠서 위에 부분은 인증 Part이고 아래 부분은 기존 WPA2 Part이다. 즉, 기존 WPA2에 NFT 인증을 추가하였다. 제2.2.5 장에서 설명했듯이 대부분의 WPA2 크랙은 올바르고 안전한 인증을 거치게 되면 많은 부분이 해결될 수 있다. 다만 WPA3에서 사용하는 장치 간 동시 인증 같은 경우는 라우터 장비 자체가 그 기능을 지원해야 하므로 기존 라우터 장비에서 사용하기 어려운 경우가 많다. 본 제안은 기존 라우터 장비와 WPA2는 그대로 사용하면서 인증 수단으로 NFT를 사용한 것이다. 그리고 이 제안은 Wi-Fi를 Hidden SSID로 설정하기 때문에 Wi-Fi 검색 창에서 뜨는 List에서 골라서 접속하는 것이 아닌, 저자가 제작한 실행 프로그램(exe)로 접속하는 것이므로 Wi-Fi Search List에 검색되지 않는다.

3.2 프로그램 구현 방법 소개

본 논문의 구현은 Python과 JS의 동작이 편리한 Window 환경을 기반으로 하였다.

Fig. 10은 본 논문 Approach의 프로그램 흐름이다. 아라비아 숫자 flow 대로 설명한다. (1)Wi-Fi에 접속하려는 STA은 AP에 접속하기 위해 Wi-Fi Search List에서 SSID를 찾는 것이 아닌, 저자가 만든 실행파일(exe)을 실행한다. 프로그램의 전체적인 Frame을 만들기 위해 이때 사용한 것은 Python의 모듈 중 Pywebview를 사용하였다. (2)접속하는 AP에는 제2.2.3 장에서 언급한 라즈베리파이 OpenWRT가 동작하고 있다. NFT 인증을 위해서는 블록체인과 통신해야 하는데 JS를 사용하는 것이 지원하는 API가 많기 때문에 가장 편리하다. JS에서 제공하는 DOM(Document Object Module)인 window 객체를 이용해서 사용하는 블록체인이 이더리움이라면 window.ethereum과 web3.eth 객체를 사용하고 클레이튼이라면 window.klayth과 caver 객체를 사용한다. 플랫폼마다 사용해야할 객체와 라이브러리는 다르며 해당 플랫폼의 개발자 Document 페이지에서 무엇이 사용되는지 알 수 있다. 저자는 이더리움을 사용하였다. 먼저 이 시스템에 사용할 Smart Contract를 작성한다. 이 작성한 Smart Contract를 EtherScan 같은 웹페이지가 아닌 프로그램 상에서 Code로써 호출하고 작동하려면 Smart Contract의 abi 값을 추출해서 JSON 형태로 저장해두어야 Contract를 호출하기 편하다. 미래사회에는 누구나 1개 이상의 NFT를 가지고 있을 것을 전제로 논문을 작성하지만, 이 시스템에 NFT 주소를 등록하기 싫은 경우, 혹은 NFT가 1개도 없는 경우를 생각해서 NFT Mint 함수를 작성한다. 인증을 하기 위해 인증하려는 NFT가 Whitelist에 등록이 되었는지 확인하는 mapping array를 관리하는 NFT Contract Whitelist, 거기에 등록과 해제를 하는 AddRemove NFT Contract 함수, NFT의 개수를 세는 NFT Count Check 함수를 작성한다.

마지막으로 Wi-Fi의 정보인 SHA256과 salt로 해시화된 SSID와 PW의 정보를 담는 Private Variable(Solidity에서 Private Variable은 Contract에서 허가된 Wallet만 호출할 수 있다)를 만들고 그것을 변경하는 Change SSID PW 함수를 작성한다. 이렇게 작성된 Contract에서 JS쪽에서 불러오는 Contract는 2개이다. NFT Whitelist와 NFT Count Check를 통해 등록된 NFT가 1개 이상인 것이 확인되면 Contract로부터 해시화된 SSID와 PW를 받아온다. 그 값들

Algorithm 5 Smart Contract(N-WPA2)

Require: *ERC − 721ofOpenZeppelin*
Input: *STA.Wallet_address*
Output: *Hashed SSID, Hashed PW*
1: Create Function 'Mint NFT' ▷ to mint NFT
2: Create Function 'AddRemove NFT Contract' ▷ to manage NFT whitelist
3: Create Function 'NFT Whitelist' ▷ Whitelist mapping array
4: Create Function 'NFT Count Check' ▷ to counting STA.NFT
5: Create Function 'Change SSID PW' ▷ to change Wi-Fi Info
6: **if** *NFT Count Check && NFT Whitelist* $>$ 1 **then**
7: Return Hashed SSID, Hashed PW
8: **end if**

Fig. 11. Pseudo Code of N-WPA2 Smart Contract

은 JS와 Python의 프로세스 간 통신(IPC)를 통해 (3)JS에서 Python으로 전달된다. Python Code 측에서 해시화된 값들을 복호화하고 그것을 토대로 Wi-Fi Profile.XML파일을 생성한다. (4,5)XML 파일을 생성한 후에 STA의 MAC Address를 가져와서 OpenWRT의 무선 정보에 등록한다. (6)MAC Address까지 OpenWRT에 등록이 되었으면 생성한 XML파일을 STA에게 전달해주고 AP에도 적용한다. XML 파일을 전달받은 STA은 그 Profile을 통해 키를 install하고 Wi-Fi 접속을 할 수 있다. 마지막으로 Server Side에선 JS로 Blockchain Event Monitoring이 동작하고 있어야 한다. 모니터링 구현에서 가장 중요한 것은 eth.getBlockNumber()와 eth.Contract.getPastEvent() 함수이다. getBlockNumber로 현재시점의 최신 블록의 위치를 수신하고, 일정 시간 후에(저자는 5초) 다시 getBlockNumber를 하여 그 차이만큼의 블록들을 getPastEvent로 Contract에서 발생한 이벤트들을 검사한다. 이때, 이벤트를 수신할 NFT가 Transfer함수의 event를 call할 수 있도록 처음부터 해당 NFT Smart Contract에 구현이 되어있어야 한다. 본 논문에서 사용되는 스마트 Contract(Fig. 11)는 OpenZeppelin의 가이드라인을 준수하였고 Contract 안의 각 함수별 Event를 Call할 수 있도록 구현되어있다. 해당 장을 정리하자면, Server Side에는 STA과 통신하는 Python 코드, 블록체인으로부터 인증을 하고 Wi-Fi 관련 값을 받아오기 위한 JS코드, NFT의 움직임을 감지하기 위한 Blockchain Event Monitoring JS코드까지 총 3개가 작동해야 한다.

4. 실험 설계 및 결과 분석

4.1 실험 환경 설계

본 논문에서 제안하는 NFT 인증을 통한 키 교환 없는 실용적인 WPA2의 성능을 다양한 디바이스 환경에서 평가하기 위해 여러 개의 Testbed 디바이스 환경을 구성하였다. 본 논문의 제안은 Window 환경에서 동작하기 때문에 다양한 디바이스들로 각각의 Window 환경을 구축하였다. 일반적인 데스크톱, 일반적인 노트북, Window 10 in Macbook (BootCamp), LattePanda(SBC)에 Window 환경을 구축하였다.

이 디바이스들(Table 3)의 공통점은 x86 기반 아키텍처이다. 이유는 arm 기반 아키텍처의 Window에서는 2022년 11월 아직까지는 제대로된 Wireless Network 연결을 지원하지 않기 때문이다. 그리고 Testbed 환경도 중요하지만 Wireless Adaptor의 성능에 따라 실험의 결과가 차이날 수 있기 때문에 동일한 Wireless Adaptor를 사용하였다. 예전에 출시된 스펙의 제품이라 2022년 시점의 최신 Wi-Fi6 연결 속도와 인터넷 속도는 다소 차이가 날 수 있다. 마지막으로 본 실험에서 무선 공유기(AP)이자 NFT 인증 서버 역할을 하는 라즈베리파이 4B와 저자의 집에서 사용 중인 I사의 구형 제품을 AP 실험 대조군(A104M 모델)으로 준비하였다. 실험은 크게 기존 크랙 툴들로부터 방어가 되는지, 그리고 이러한 시스템을 도입함으로써 기존 WPA2와의 성능비교를 해서 실사용에 큰 지장이 없는지, 마지막으로 NFT 인증에 대한 새로운 공격을 대비하여 서명 방지와 인증 서버에 대한 DoS 공격으로 실험 대분류를 구성하였다.

Table 3. Testbed Environment

	CPU	RAM	OS
Desktop	i5 9600K	DDR4 16Gb	Window 10
Laptop	i3 5005U	LPDDR4 8Gb	Window 10
Macbook	i5 6360U	LPDDR3 8Gb	Window 10(BootCamp)
Latte Panda(SBC)	Celeron N5105	LPDDR4 8Gb	Window 11

4.2 실험 결과 분석 I

1) KRACK

첫 번째로 Table 4의 대분류 I 실험에서 KRACK에 대한 보안성 평가이다. 결과적으로 제안하는 방식을 이용하면 KRACK은 불가능하다. 이유는 크게 2가지이다. SSID가 검색되지 않고, 접속 시도가 되어야 키 교환을 진행하는데 MAC Address 기반 Filtering 처리를 하였기 때문에 허용되지 않은 MAC Address를 가진 Network 기기들은 접속을 시도조차 할 수 없다. 즉, 키 교환을 시작조차 할 수 없다. Fig. 12는 SSID가 검색되지 않지만 직접 입력해주고 KRACK 툴을 구동한 결과이다. I사의 A104M 대조군은 5번의 KRACK 성공 평균값을 나타내었다(15.732sec).

2) WPS

Fig. 13은 Pixie Dust 공격에 대한 보안성 평가이다. 본 논문의 제안은 OpenWRT에서 기본적으로 WPS를 꺼놓는 제안이지만, 해당 실험을 위해 WPS 기능을 On한 상태로 진행하였다. 마찬가지로 크랙은 이루어지지 않는다. OpenWRT에서는 WPS 번호를 사용자가 임의로 변경할 수 있고 MAC Address 기반 Filtering을 하였기 때문이다. 마찬가지로 5번의 평균값을 나타내었다. 그런데 처음 WPS 공격에서 PIN Number를 획득하는데 시간이 30초 이상 소요되었지만 한 번 공격 대상 AP의 PIN Number를 획득하고 나서는 크랙 시간이 KRACK보다 적게 나오는 결과를 보였다(9.245sec).

Table 4. Experiment Items

Major Classification	Subcategory	TestCase Description
I. Existing Crack Tool Resistance (Evaluation of Security)	PMK(PSK inference Attack)	Assessment of security against Dictonary attacks
	Pixie Dust(WPS Attack)	Assessment of security against WPS attacks
	KRACK(Key Reinstallation Attack)	Assessment of security against Key re-injection
II. Usability compared to existing WPA2 (Evaluation of Practicality)	4-way handshake Delay Time	Delay for 4-way handshake Time Assessment
	Total Auth time	Assessment of total connection time of my Approach
	Compare Network Speed	Evaluation of network speed this proposal and existing WPA2
III. Review of new attack methods for NFT authentication (Safety Assessment of Proposal)	Forgery of Signature	possibility of signature forgery
	Resistance to DoS Attack	Assessment of DoS Safety for Auth Server

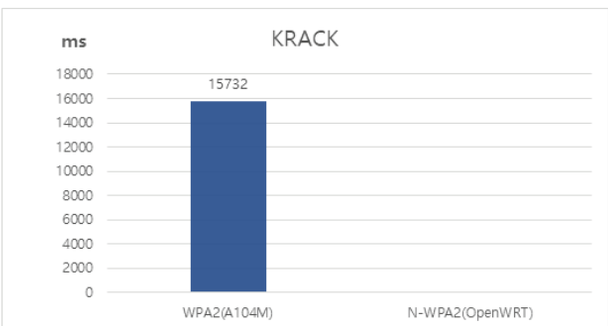

Fig. 12. KRACK Time Analysis

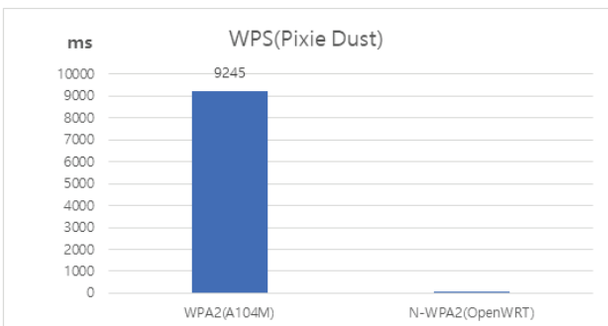

Fig. 13. WPS Attack Time Analysis

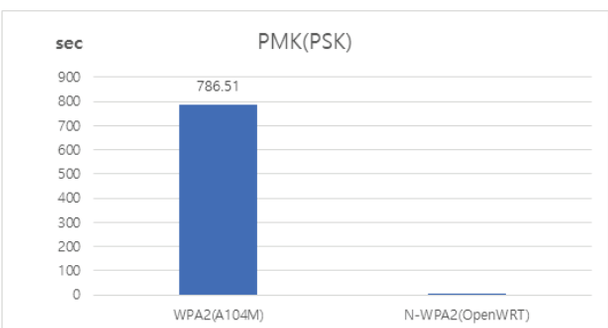

Fig. 14. PMK(PSK) Attack Time Analysis

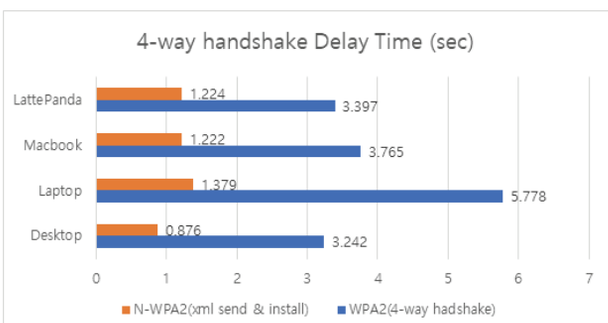

Fig. 15. 4-way handshake Delay Time

3) PMK

Fig. 14는 사전암호파일 공격에 대한 보안성 평가이다. 제 2.2.4 장 처음 부분에서 설명했듯이 PTK를 derive하기 위해선 5가지 인자가 필요하다고 설명했다. 그 중에서 AP측면에서 바꿀 수 있는 것은 PMK, SSID와 ANonce이다. OpenWRT에서 STA의 연결 요청마다 SSID와 ANonce를 랜덤 재설정되도록 하였고 마찬가지로 MAC Address Filtering 처리가 되어 있어서 크랙되지 않는다(786sec).

실험 결과들을 통해 본 제안은 WPA2의 대표적인 3가지 크랙으로부터 완전히 방어할 수 있음을 보였다.

4.3 실험 결과 분석 II

대분류 II 실험에서는 기존 WPA2와 비교하여 실사용에 불편함이 없는지 실험한다. 위에 언급한 Testbed(Table 3)의 4가지 기기들의 연결 소요시간을 측정하였다.

1) 4-way handshake Delay Time

Fig. 15는 4-way handshake Delay Time을 측정한 것이다. 측정 요소는 Wi-Fi Profile.XML 파일의 전송시간과 Install시간을 일반적인 WPA2 연결과 비교한 것인데, 본 논문의 제안은 XML 파일을 통해 PTK를 install 하므로 4-way handshake가 없기 때문에 PTK install 시간과 일반적인 4-way handshake 연결 시간을 비교하였다. 실험 단위는 초(sec)이며 Wi-Fi Profile 전송을 통해 Key를 설치하는 것이 전체적으로 더 빠른 결과를 나타냈다.

2) Total Auth(Connection) Time

Fig. 16은 Window에서 Wi-Fi 접속을 명령어로 처리한

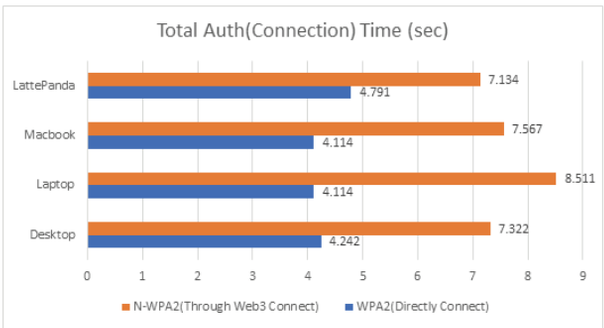

Fig. 16. Total Auth(Connection) Time

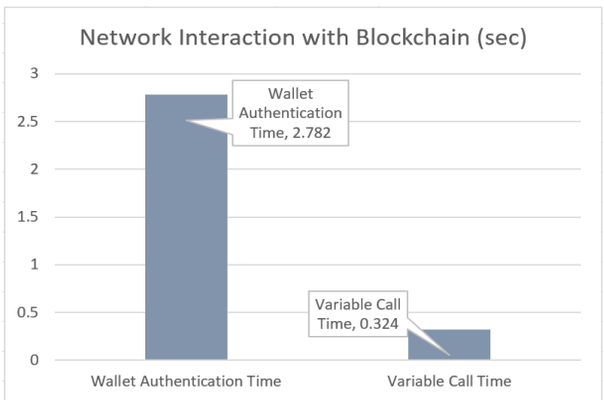

Fig. 17. Network Interaction with Blockchain

$$(3)\quad WPA2:\ O(n) = 4n + \alpha$$
$$N\text{-}WPA2:\ O(a+b+c) = a + b + c + \alpha$$

Fig. 18. Time Complexity of N-WPA2 and WPA2

것과 NFT 인증 이후 N-WPA2 연결까지 걸리는 시간을 비교하였다. NFT 인증 이후로 한 이유는 WPA2 에서도 비밀번호를 사람이 입력하는 시간이 존재하고, NFT 인증도 사람이 직접 지갑 인증해야 하므로 사람이 입력하는 시간을 제외한 것을 비교하였다. Fig. 16의 전체 인증 과정을 세부적으로 나타내기 위해 Fig. 17과 Fig. 18을 추가하였다.

Fig. 17은 전체 인증 과정 중, 지갑 인증 시간과 지갑 인증 후 블록체인으로부터 값을 받아 오는 시간을 측정하였다. 이 실험 데이터에서 유의해야할 부분은 Web3 환경(본 논문에서는 이더리움 테스트넷)의 네트워크 상태에 따라 속도가 다소 차이날 수 있다. 본 실험에서는 지갑 인증 부분에서 2.782sec라는 값이 나왔지만 예를 들어, 전 세계인이 갑작스럽게 테스트넷을 이용하는 상황이라면 이 값은 많이 증가할 수도 있다.

Fig. 18은 기존 WPA2와 N-WPA2의 시간복잡도를 Big O 기법으로 표현하였다. 두 식에서 공통적으로 표현된 α(알파, 상수값)는 키 install 시간이다. WPA2의 경우 4-way handshake의 4번 송수신 과정을 1차 항이 1개인 4n의 식으로 나타내어 O(n)으로 나타낼 수 있다. N-WPA2는 4-way handshake는 없고 지갑 인증 시간(a), 블록체인으로부터 값

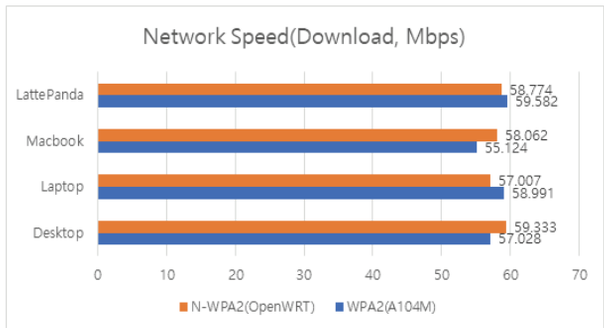

Fig. 19. Network Speed Comparison

을 받아오는 시간(b), 키를 STA와 AP에 전달하는 시간(c)로 1차 항이 3개인 a+b+c의 식으로 나타낼 수 있다.

Fig. 16과 Fig. 17 모두 5번의 평균값을 토대로 작성하였고 최종적으로 N-WPA2 전체 연결 시간은 Web3 환경(이더리움 테스트넷)에서 지갑 인증 및 Contract로부터 값을 호출하는 시간이 Web3 네트워크 환경에 따라 속도가 달라지기 때문에 기존의 WPA2보다 소요 시간이 다소 길게 나타났다.

3) Network Speed 비교

Fig. 19는 Wi-Fi 접속이 된 상태에서 인터넷 속도 측정 툴로 다운로드 속도를 비교하였다. 인터넷 실험 환경은 한국의 LG U+ 자회사 Hello Vision의 100Mbps 회선이다. 인터넷 환경은 열악하지만 Table 3의 Testbed 장치 모두 동일한 환경에서 실험하였으므로 N-WPA2와 기존의 WPA2의 차이가 거의 없거나 OpenWRT의 성능 덕에 소폭 증가한 부분도 있음을 알 수 있다.

4.4.3 장의 실험을 통해 N-WPA2의 전체 연결 시간이 조금 더 소요되긴 하지만 4-way handshake의 줄어든 시간도 있고, 인터넷 속도 측면에서 보았을 때 실사용에 큰 무리가 없음을 보였다.

4.4 실험 결과 분석 III

1) 서명 위조

대분류 III 실험에서 서명 내용 위변조 가능성 평가 실험은 일반적으로 사용하는 Chrome 브라우저의 개발자 모드에서 Signature 부분을 변경한 채로 Signing이 허용되는지 실험하는 것이다.

Fig. 20은 지갑으로 서명한 것을 ECDSA 디지털 서명의 세 가지 구성 요소로 사용되는 V,R,S로 나누어서 전송하는

Algorithm 7 Signature VRS apply
Require: *web3 JS API from cdnjs*
Input: *STA.Wallet_Signature*
Output: *V, R, S of STA.Wallet_Signature*
1: sig = web3.sign(STA.Wallet_Signature)
2: const v = '0x' + sig.substring(2).substring(128,130) ▷ sig is cdnjs API
3: const r = '0x' + sig.substring(2).substring(0,64)
4: const s = '0x' + sig.substring(2).substring(64,128)
5: *signature* = [v,r,s]
6: Send *signature* to AUTH server

Fig. 20. Way of V, R, S Signature Coding

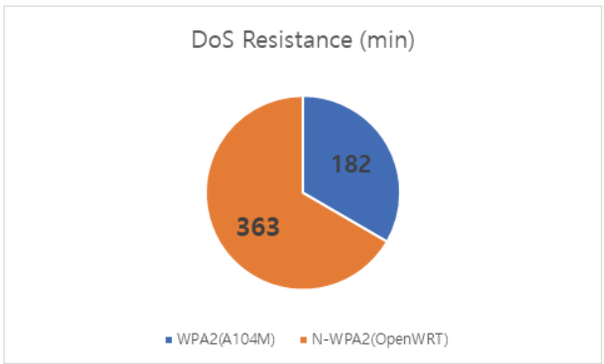

Fig. 21. DoS Resistance of Auth Server and Control group AP

방법을 기술한 수도코드이다. 인증페이지를 구현할 때 Fig. 20과 같이 서명 부분을 V,R,S로 나누게 되면 Chrome 브라우저에서 개발자 모드로 진입하여 서명 부분을 변조해서 전송해도 서명을 받는 Server Side에서 Wallet 주소나 Signature 내용 중 하나만 틀려도 서명이 성립되지 않는다. 인증 서버 웹페이지(html)에 V,R,S 코드 적용 후 변조된 서명으로 NFT 인증이 되지 않음을 확인하였다.

2) 인증서버 DoS 저항성

마지막 실험으로 I사의 제품과 OpenWRT가 서로 동일한 DoS 공격에서 얼마나 버티는지 비교한 그래프이다(Fig. 21). DoS Attack 장비로는 가상 머신이 아닌 실제 Intel Xeon E5-2609 2.4Ghz Quad-Core, DDR3 8Gb RAM 장비 3대를 이용하여 Xerosploit을 이용했다. Xerosploit은 무의미한 다량의 ICMP Ping을 고속으로 보내서(Hping) 공격하는 DoS 툴이다. 더 최신의 장비로 DoS 실험을 하면 시간이 더 적게 나오겠지만, 평균적으로 라즈베리파이를 이용한 라우터가 약 2배 정도의 저항성을 보였다(363min).

4.4.4 장의 실험을 통해 본 논문의 제안이 가질 수 있는 서명 위변조의 위험성에 대해 평가해보았고, 사이버테러의 대표적인 DoS 공격에 대해서도 실험하여 일반적인 라우터 회사의 제품을 사용하는 것보다 안정적임을 보였다.

5. 결론 및 향후 연구

시대는 Web3의 시대로 향하고 있다. 이제 사람들은 인터넷 공간에서 자기 자신을 증명, 증빙할 수 있어야 하며 그 과정에서 소유권 증명과 보상(Incentive)에 대한 증빙이 중요해지고 있다. 곧, 그런 것들을 모든 사람들이 NFT나 SBT[14]로 자기 자신을 증명하게 될 것이다. 초연결사회라는 단어가 등장하며 대부분의 모든 IT 기기들은 Wireless Network를 지원하며 그 기기들이 인터넷에 접속하기 위해서 Wi-Fi Router의 역할은 더욱 중요해지고 있다. 하지만 아직까지도 구형의 많은 기기들을 범용적으로 연결하기 위해 현재까지도 WPA2가 많이 사용되고 있고 제2.2.4 장에서 언급한 WPA2의 취약성을 해결하기 위해 윈도우 환경 기반의 NFT 인증 WPA2 연결 방식을 제안하였다.

본 논문에서는 미래사회에는 개인이 NFT를 1개 이상은 가지고 있다고 가정하고 이러한 논문의 제안을 하였다. 본 논문에서는 기존 WPA2의 문제점은 해결하면서도 실용성을 강조하고 싶었다. 저자가 말하는 실용성이란 패스워드를 암기하지 않아도 되는 실사용의 장점도 포함되지만, 장비를 업그레이드하지 않고 추가적으로 무언가를 더 준비해야하는 부수적인 불편함을 제외하는 것도 포함한다. 즉, 본 논문에서는 특별한 준비 없이 개인이 가지고 있는 NFT로 자신을 증명하고 그것을 인증수단으로 사용하며 크랙 툴들은 방어하며 기존 WPA2를 그대로 사용하여도 실사용에 무리가 없는 제안을 실험으로 보였다.

향후 연구로는 모바일 기기에서 본 시스템을 작동할 수 있도록 구현할 예정이다. 현재 모바일 기기는 크게 안드로이드, 애플 군으로 나눌 수 있는데 두개의 코딩 방식이 다르며 모바일 OS 내부 정책에서 Wi-Fi 권한을 획득하는 것도 방법이 다르기에 먼저 쉽게 작성할 수 있는 윈도우 기반으로 코드를 작성하였다. 모바일 기기용 N-WPA2 앱을 만들고 Metamask 앱과 연동하여 편리한 모바일 연결수단을 제공할 계획이다.

References

[1] D. J. Fehér and B. Sandor, "Effects of the WPA2 KRACK attack in real environment," *IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. 2018.

[2] V. O. Etta, A. Sari, A. L. Imoize, P. K. Shukla, and M. Alhassan, "Assessment and Test-case Study of Wi-Fi Security through the Wardriving Technique," *Article in Mobile Information Systems*. 2022.

[3] T. Eun and S. Park, "Introduction to attack methods of locally accessible private blockchain," *Korea Computer Congress 2022 (KCC 2023)*, pp.1276-1278, 2022.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," [Internet], https://assets.pubpub.org/d8wct41f/31611263538139.pdf, 2008.

[5] V. Buterin. "A next generation smart contract & decentralized application platform," [Internet], https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf, 2013.

[6] E. Baray and N. Kimar Ojha. "WLAN security protocols and WPA3 security approach measurement through aircrack-ng technique," *International Conference on Computing Methodologies and Communication (ICCMC)*. 2021.

[7] L. G. Nikolov, "Wireless network vulnerabilities estimation," *Security & Future*, Vol.2, No.2, pp.80-82, 2018.

[8] C. D. Omorog, B. D. Gerardo, and R. P. Medina. "The performance of blum-blum-shub elliptic curve pseudorandom number generator as WiFi protected access 2 security key generator," *Proceedings of the 2nd International Conference on Business and Information Management (ICBIM)*, 2018.

[9] J. Guo, M. Wang, H. Zhang, and Y. Zhang, "A secure session key negotiation scheme in WPA2-PSK networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2020.

[10] V. L. L. Thing, "IEEE 802.11 network anomaly detection and attack classification: A deep learning approach," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2017.

[11] U. Chatterjee, R. Sadhukhan, D. Mukhopadhyay, R. Chakraborty, D. Mahata, and M. Pranhu, "Stupify: A hardware countermeasure of KRACKs in WPA2 using physically unclonable functions," *Companion Proceedings of the Web Conference 2020*, 2020.

[12] Y. Niu, L. Wei, C. Zhang, J. Liu, and Y. Fang, "An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain," *IEEE/CIC International Conference on Communications in China (ICCC)*, 2017.

[13] Wi-Fi Alliance, "WPA3 AND ENHANCED OPEN: NEXT GENERATION WI-FI SECURITY," ARUBA [Internet], https://www.arubanetworks.com/assets/wp/WP_WPA3-Enhanced-Open.pdf, 2018.

[14] Weyl, Eric Glen and Ohlhaver, Puja and Buterin, Vitalik. "Decentralized Society: Finding Web3's Soul," Social Science Research Network (SSRN) [Internet], https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4105763, 2022.

은 태 영

https://orcid.org/0000-0003-3282-8235
e-mail : tyeun7@sogang.ac.kr
2016년 한국외국어대학교 컴퓨터공학과 (학사)
2022년 서강대학교 컴퓨터공학과 석사과정
관심분야 : 블록체인, 네트워크, 보안

Alshihri Saad

https://orcid.org/0000-0001-8651-1597
e-mail : saaad77.sa@gmail.com
2022년 서강대학교 컴퓨터공학과 박사과정
관심분야 : 블록체인, 보안

박 수 용

https://orcid.org/0000-0002-3979-0586
e-mail : sypark@sogang.ac.kr
1986년 서강대학교 컴퓨터공학과(학사)
1988년 Florida State University, Computer and Information Science(석사)
1995년 George Mason University, Information Technology(박사)
현 재 서강대학교 컴퓨터공학부 교수 및 지능형 블록체인센터장
관심분야 : 소프트웨어공학, 블록체인