ISSN: 2769-545X

Research Article

Advances in Machine Learning & Artificial Intelligence

WPA2 Security Enhancement Method through NFT in Web3 Era

Tae-Young Eun and Soo-Yong Park*

Department of Computer Science and Engineering, Sogang University, Republic of Korea.

***Corresponding Author**
Soo-Yong Park, Department of Computer Science and Engineering, Sogang University, Republic of Korea.

Submitted: 2023 Nov 02; **Accepted**: 2023 Nov 20; **Published**: 2023 Dec 01

Citation: Eun, T-Y., Park, S-Y. (2023). WPA2 Security Enhancement Method through NFT in Web3 Era. *Adv Mach Lear Art Inte, 4*(2), 94-109.

Abstract

WPA2 is the most used wireless communication protocol in the world (2023). It first appeared in 2006, and now several vulnerabilities have been identified. 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 connecting devices. We are currently living in the Web3 era. In the future society, people will have more than one NFT each. It is possible to improve the security of WPA2 by using this as an authentication means. In this paper, see the principles of WPA2 crack tools that are currently used today and suggest a way to defend against them using NFT. An experiment was carried out on the security of WPA2, which is widely used in SOHO environments, using only SBC (Single Board Computer) and NFT without expensive routers or additional authentication means. Hacking time, Internet connection delay time, download speed, etc. were compared on various PCs. In conclusion, this proposal demonstrated that representative WPA2 cracking tools can be defended without performance degradation compared to existing WPA2.

Keywords: Block Chain, Ethereum, NFT, Wireless LAN, Wi-Fi, WPA2, Hacking, Security

1. Introduction

Wireless LAN (Local Area Network) is a technology that allows users within a certain range to access a network from anywhere without a physical connection. A device called a wireless AP (Access Point, router) is used to connect a client (STA) to a remote server without a wired cable. The earliest adopted router encryption method was WEP. It was adopted as a Wi-Fi security standard in 1999, and later WEP with increased password bits (192 bits) appeared. However, even though the key capacity was increased, security problems were discovered, and in 2005, the U.S. Federal Bureau of Investigation (FBI) demonstrated the process of decrypting WEP encryption within a few minutes using free software. Today, the Wi-Fi Alliance, a non-profit Wi-Fi technology certification body, officially retired WEP in 2004. The Wi-Fi Alliance launched the wireless data protection (Wi-Fi Protected Access, WPA) method to replace WEP, and WPA was officially adopted in 2003. WPA is 256 bits (which makes it stronger than the 64 bits of the existing WEP), but TKIP, a core component of WPA, recycles the WEP method, which was again discovered as a vulnerability. In 2006, Wireless Data Protection II (Wi-Fi Protected Access II, WPA2) replaced WPA. Unlike WPA, WPA2 uses the AES algorithm as standard, and CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) replaces TKIP. However, as time passed, problems with WPA2 were also discovered. Problems and vulnerabilities of WPA2 are covered in Chapter 2.4. To solve the problems of WPA2, the Wi-Fi Association announced the WPA3 security protocol in 2018. The fundamental drawback of WPA2 is its incomplete 4-way handshake. In addition, when using PSK (Pre-Shared Key), it exposes the Wi-Fi connection to risk. WPA3 implements additional security, making it difficult to guess the encryption key during the connection process. By replacing WPA2's PSK (Pre-Shared Key) with SAE (Simultaneous Authentication of Equals), it protects against attacks from KRACK, which is WPA2's most vulnerable crack tool. However, the connecting devices, STA (station, client) and AP (access point, router), must support SAE. The latest smartphones support WPA3, but most routers (which have been used in the past) do not support WPA3. This must be upgraded in hardware, not in software. Router companies have been releasing AP's SAE functions since 2019.

According to a Wigle.net survey, WPA2 will be the most used protocol worldwide in 2023 (Figure 1). At the time in 2006, it was a protocol with excellent performance, but today, anyone can

attempt to crack it in a short time by purchasing wireless monitoring equipment that costs less than $25 [1]. Widely used WPA2 attack methods include Pixie Dust crack using WPS, Key Reinstallation Attack (KRACK), Key Derivation Attack (PMK) using expected dictionary key sharing files, and WPA2 Handshakes Capturing packet analysis. Using WPA2 in public places is dangerous. Rogue War Driving, in which hackers drive around public places such as cafes or airports and hack wireless APs from a distance to collect personal information, is in full swing [2]. In the past, the author presented a paper at an academic conference on how easy WPA2 hacking is to raise awareness of this risk [3].

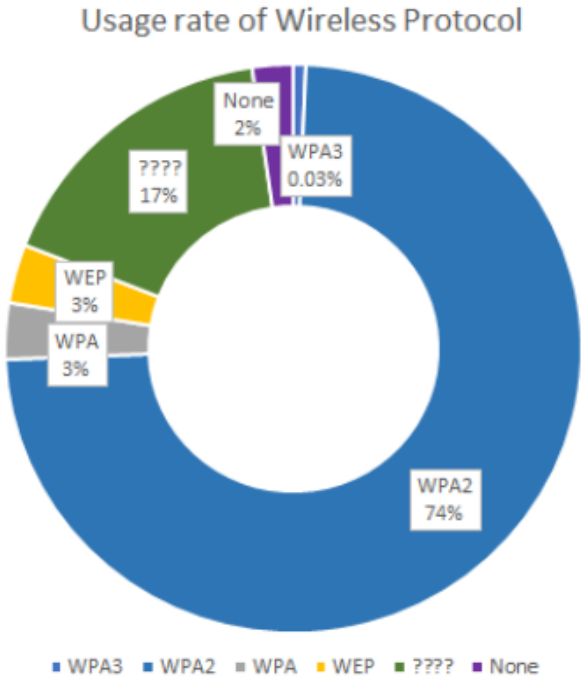

Figure 1: Usage rate of Wireless Protocol in 2023

WPA2 is still widely used today. However, it is impossible to upgrade all WPA2 routers in the world. The author proposes a methodology that makes WPA2 routers safe from crack tools without hardware upgrades. It is useful for use in SOHO environments and has no performance degradation compared to existing WPA2. The name of the proposal is N-WPA2.

1.1. Background Knowledge

1.1.1. Blockchain and NFT (Non-Fungible Token)

Blockchain is a form of distributed database managed through a P2P network. A technology stores ledgers containing transaction information on all computers connected to the blockchain rather than storing them in one central server. In 2008, S. Nakamoto's paper "Bitcoin: A Peer-to-Peer Electronic Cash System" made blockchain technology widely known to the world, and he developed blockchain to solve the problems that occurred while developing Bitcoin [4]. Transactions made by previous users are recorded in the block. This is a P2P method (Figure 2) and is distributed equally to all users, so transaction details cannot be arbitrarily modified or deleted by a single individual.

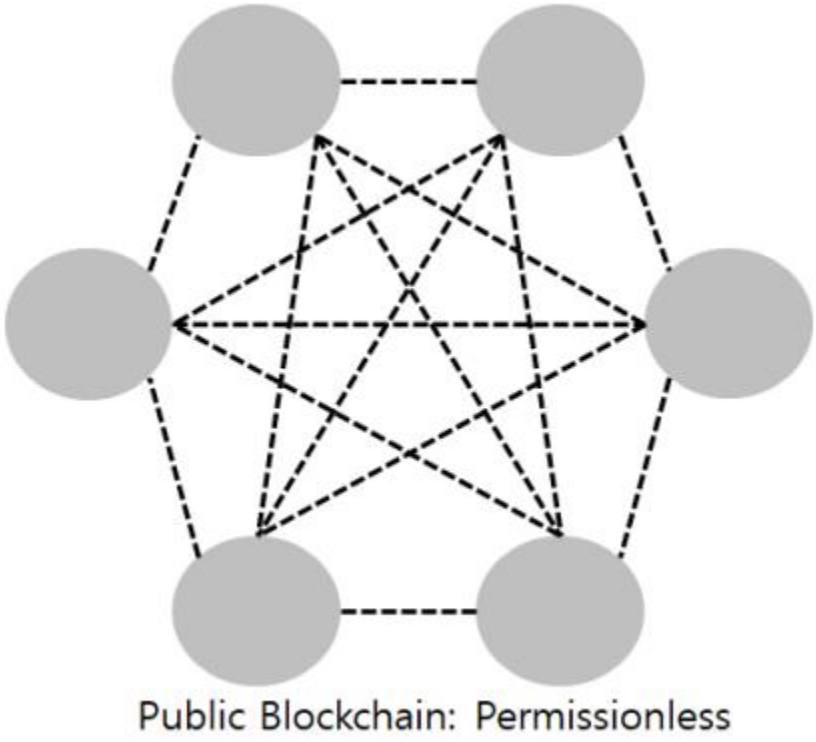

Figure 2: Network Structure of Public Blockchain

Since each block has a link to the date of discovery and the hash value of the previous block, a set of such blocks is called a blockchain. Simply put, the transaction ledger is called a block, and what connects them is called a chain, collectively called a blockchain. In public blockchains such as Bitcoin, Proof-of-Work (PoW) was introduced to verify that block transactions were not forged. PoW is the process of changing the header nonce of a block until it becomes a value below the difficulty level set by the Bitcoin network system to calculate the hash value for the next block. A block is created once every 10 minutes, and this creation 6 times is called '6 Confirm Finalizing' in Bitcoin. Once this is accomplished, the computing power when writing this paper cannot falsify the values of previous blocks.

Ethereum is another public blockchain platform and the name of the platform's currency, created by Vitalik Buterin in 2015, inspired by a 2009 Bitcoin paper [5]. The biggest difference from Bitcoin is a variety of uses due to the introduction of smart contracts. In other words, while Bitcoin focuses only on payment, and transaction-related systems, and functions as a currency, Ethereum allows anyone to create various decentralized distributed applications (DApps) such as contracts, electronic voting, and DAO as well as transactions and payments through smart contracts. Due to these differences, Bitcoin is called the first-generation blockchain, and Ethereum is called the second-generation.

NFT (ERC-721), the subject of this paper used in Ethereum, has the standard as shown in the following table (Table 1). As the coronavirus outbreak occurred in 2019, many assets were concentrated in cryptocurrency. From 2020 to 2023, several problems such as scams and hacking occurred in NFTs, but there were no problems with blockchain and NFTs themselves. Most of the problems were the leakage of users' personal information or the security status of NFTM (NFT-Market). The author intends to utilize NFT's unique originality, ownership, and function as an identification card as a means of authentication.

Function	Description
Balance Of	Returns the number of NFT Owner's owned
Owner Of	Returns the owner address of an NFT with a specific token ID
approve	Allow specific accounts to use one NFT owner's own
Get Approved	Returns whether certain NFTs have been authorized for use by other accounts
Set Approval For All	Allow specific accounts to use all NFT Owner's own
Is Approved For All	Returns whether the owner has allowed a particular account to use it for all of NFTs

Transfer Form	NFT Ownership Transfer
Safe Transfer From	Send NFT ownership after confirming that the receiving address can receive NFT

Table 1: ERC-721 Standard

1.2. Digital Signing

Currently, when accessing most well-known websites such as Google or Yahoo, the client's browser and server side use SSL authentication for encrypted communication, and what is used at this time is a digital signature. In Web3 environments such as blockchain, digital signatures, and code signing are used. A digital signature can be expressed as a signature and verification. When signing using a wallet in a blockchain (when sending data), the data and the hashed data are encrypted with the wallet's private key sent to the verifier (Figure 3).

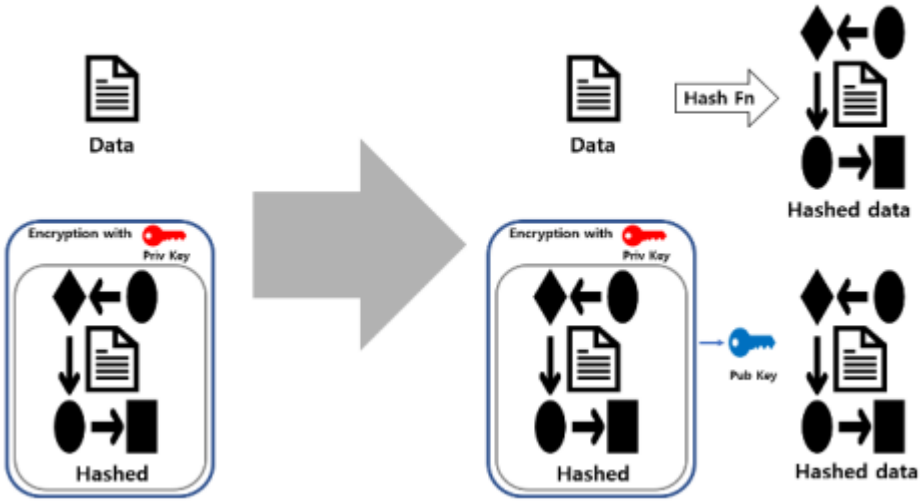

Figure 3: Signing and Verification

A verifier that receives two pieces of data verifies that the two match by running a hash function on the unencrypted data and decrypting the encrypted data with a known public key. The former called signature, and the latter called verification. Through these signatures and verifications, a qualified person can call or change (update) the value (variable) in the Smart Contract. As will be explained in Chapter 2.4, in conclusion, WPA2 causes problems because many keys and information required for communication are transmitted as signals in the air. In digital signatures, the private key that encrypts and decrypts data is not transmitted. That is the difference from WPA2 communication.

1.3. Open WRT on SBC (Single Board Computer)

Most of the security vulnerabilities that will appear in Chapter 2.4 can be solved by using expensive router equipment from security companies. Most of the well-known high-end routers from Company C cost from as little as $500 to over $3,000. According to a survey by Cyber Defense Magazine in 2023, 45% of all cyber-attacks worldwide are targeting SOHO environments. To put it simply, SOHO can be expressed as a home or small business. Homes and small businesses neither need nor can afford expensive router equipment. However, if WPA2 is used as is, it is exposed to many vulnerabilities mentioned in Chapter 2.4. In other words, expensive router equipment from security companies is not practical or reasonable in a SOHO network.

SBC is a complete computer consisting of a single circuit board with modules such as microprocessor, memory, and input/output that are essential for computer functions and is characterized by ultra-small size and low power consumption. In 2023, there are countless SBC companies, and there is a famous SBC company, Company H, in Korea. The one that will be used in this paper is the Raspberry Pi 4B model, which is the most general purpose and widely used for educational purposes. Raspberry Pi, which is widely used for educational purposes in third countries where computers are lacking, can run up to Windows 11, which is made with ARM. The Raspberry Pi 4B, launched by the Raspberry Pi Foundation in the UK, was released for $55, but due to the semiconductor shortage that has continued since 2020, it can be purchased for around $180 including shipping costs. Nevertheless, it is certainly more economical than the expensive security company equipment mentioned in Chapter 2.3. There are many other SBCs, but to use Python and Node.JS, a CPU with an aarch64 architecture or higher is required. That is Raspberry Pi 4B.

The router has firmware or embedded OS that matches the device installed. The approach of this paper is to use WPA2 after passing NFT ownership authentication but to proceed with this specially ordered process, it could not be implemented with the basic firmware of router companies. Additionally, to authenticate NFTs on a blockchain network, a small server that could communicate with an external network (www) and run at least JavaScript was

needed. Open WRT is a Linux-based open-source project that solves these process flow and equipment problems at once. Because it is Linux-based firmware, it is easy for developers to handle settings. Using this, it can be used as a wired/wireless router and at the same time a small Linux-embedded device. In summary, instead of using expensive router equipment from a security company, the author will install Open WRT on Raspberry Pi, a relatively inexpensive educational single-board computer, and use it as a wired/wireless router and NFT authentication server.

1.4. WPA2 Hacking Theory

1.4.1. EAP 4-Way Handshake

WPA2 uses a simple 4-way handshake called EAPOL stands for Extensible Authentication Protocol (EAP) over LAN. Since the IEEE standard diagram is complicated to explain, the author's simplified diagram explains the WPA2 connection process (Figure 4).

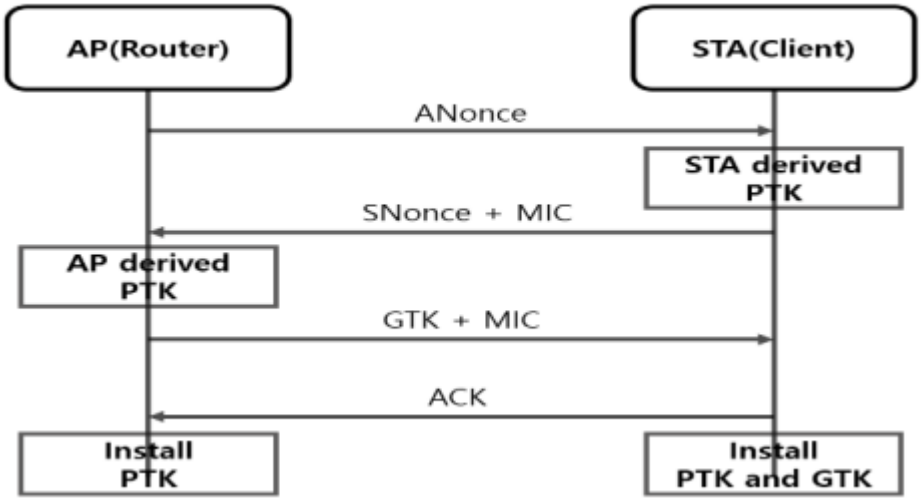

Figure 4: EAP 4-Way Handshake

Ultimately, to perform WPA2 encrypted communication, PTK (Pairwise Transient Key) must be installed on the connecting device STA (station) and AP (router). To derive this PTK, many keys are used in the process, but five key factors are required (Figure 5).

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

Figure 5: Parameter of PTK

Figure 4 explains in flow order. The device STA trying to connect receives ANonce, which is the AP's nonce, from the AP. PMK (PSK) is a pre-shared key, so both STA and AP know it. From the perspective of the STA that received ANonce, all five factors to induce PTK are in place based on Figure 5. Next, the STA sends SNonce and MIC to the AP. It stands for Message Integration Code and is a message code that proves that the STA sent it. From the perspective of AP, which received SNonce, all factors were in place to induce PTK. AP also sends STA a MIC to prove itself and a GTK (Group Temporary Key) to become a member of its group. STA checks the AP's MIC and installs PTK and GTK since it is the correct value. An ACK is sent to the AP indicating that the installation is performed correctly, and the AP installs the PTK. Now the AP and STA are ready for encrypted communication with each other. There are four representative vulnerabilities (crack tools) in this communication method, including Brutal Force.

1.4.2. PMK (PreShareKey) Attack

The first is to exploit vulnerabilities in PMK (PSK, pre-shared key) using a tool called Airmonng [6]. A pre-shared key is literally a pre-shared key. There are 3 to 4 ways to create it, and older routers can be cracked more easily if their firmware has not been patched. There is a well-made dictionary password prediction file as shown in Figure 6, and PSK can be generated by extracting 8 to 36 bits from the file and running an SSID-based key derivation function called PBKDF2. Generate PTK using this PSK and five factors and verify it to AP based on MIC. Even if the process fails, hacking can be done by repeating this process. The performance of modern hardware is amazing, so hacking can be accomplished within 20 minutes with just a well-prepared dictionary password file.

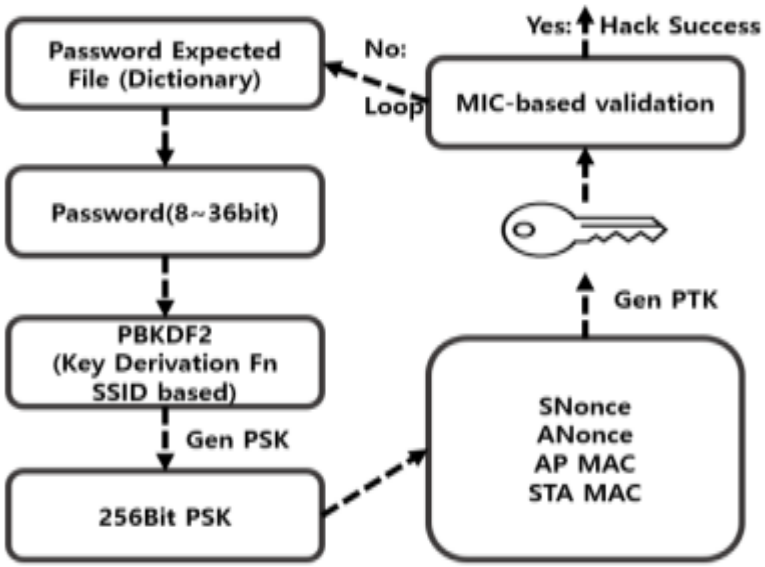

Figure 6: Principle of PMK (PSK) Crack

1.4.3. Pixie Dust (WPS) Attack

The second is the Pixie Dust Attack, which exploits WPS. The original purpose of the WPS (Wi-Fi Protected Service) button is to provide the convenience of connecting easily by simply pressing the button without entering the router's Wi-Fi password [7]. However, each router manufacturer has leaked the PIN number of the chipset built into the product, and this can be exploited to attempt more easily to crack it.

In Figure 7, there are 13 EAP-POL message transmission processes. Here, M3 and M5 answered by AP are the problem. If M2 sends an incorrect message, M3 sends a NACK indicating that it was incorrect. Through this, the number of cases that require cracking is drastically reduced. By the same principle, if M4 is incorrect, M5 sends a NACK, and the number of cases that need to be cracked is reduced once again. Lastly, if there is information about the PIN number, attacks become easier. Even if no information is leaked, the WPS PIN number is an 8-digit number. WPS attack is an intelligent Brutal Force attack, and considering the AP's response time, the hacker can receive 1 to 2 responses per second. When calculating the number of cases, 8 digits and 10 numbers per digit (0-9) result in 10^8 seconds. This value is approximately 3 years or more, but there is a rule in the PIN Number that the 8th digit is the checksum. Additionally, this 8-digit PIN Number is divided into the first 4 digits and the last 4 digits so that they can be checked independently. That is, the first four digits require 10^4 seconds, and the second half takes 10^3 seconds.

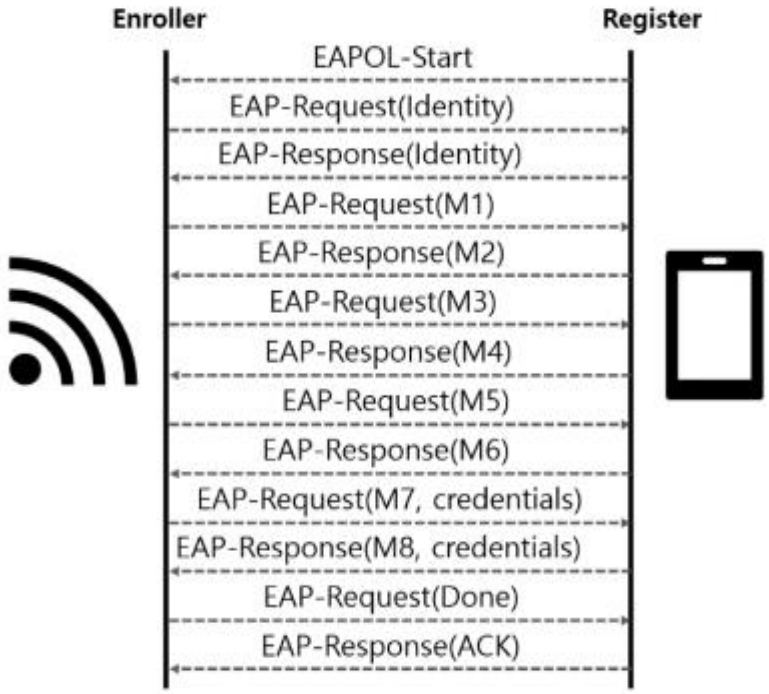

Figure 7: WPS Message Exchange

The sum of these two guesses can be expressed as Figure 8. That means 11,000 guesses are needed, and this can be compressed into about 3 hours. This method is used with a tool called Reaver or Wifite.

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

Figure 8: Number of WPS Crack Guesses

1.4.4. KRACK (Key Reinstallation) Attack

Lastly, the most infamous attack in WPA2 is KRACK (Key Reinstallation Attack). A logical flaw in the WPA2 protocol 4-way handshake causes this vulnerability, and almost all devices that support Wi-Fi (Android, Linux, Apple, Windows, MediaTek, Linksys, and OpenBSD) are affected. This vulnerability takes advantage of the fact that it is not defined when to set the negotiation key during the 4-way handshake process in 802.11i. This is an attack in which an attacker installs (reinstalls) the same key multiple times to reset the random number (Nonce) and reproduction coefficient used in the encryption protocol. The official Common Vulnerabilities and Exposures (CVE) site, which lists publicly known computer security flaws, was updated with 10 CVE entries (Table 2) in 2017. In 2018, due to these vulnerabilities, the Wi-Fi Alliance released WPA3, which blocks KRACK.

CVE	No.	Vulnerability Contents
C	13077	Reinstallation of the pairwise encryption key (PTK-TK) in the 4-way handshake.
V	13078	Reinstallation of the group key (GTK) in the 4-way handshake.
E	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.
2	13081	Reinstallation of the integrity group key (IGTK) in the group key handshake.
0	13082	Accepting a retransmitted Fast BSS Transition (FT) Re-association Request and reinstalling the pairwise encryption key (PTK-TK) while processing it.
1	13084	Reinstallation of the STK key in the Peer Key handshake.
7	13086	Reinstallation of the Tunneled Direct-Link Setup (TDLS) Peer Key (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.

Table 2: CVE List of KRACK

As shown in Figure 4, the main problem is M3, the third response from the AP during the EAP-POL 4-way handshake message exchange process. After receiving M3, the client (STA) installs a secret key and uses it to encrypt the normal data frame. However, since this is spread through the air, there is a possibility that the message may be lost in the middle. Therefore, if M4 (ACK) does not arrive within a certain period, the AP delivers M3 again. In this case, the client has the possibility of receiving M3 multiple times, and the client has the opportunity to reset the password key each time it receives M3. In addition, whenever M3 is received again, there is an opportunity to reset the data packet's Nonce and Regeneration Counter again. Therefore, if KRACK is successful, in the worst case, it can be cracked within 30 seconds, and sensitive information such as credit card and email account information can be stolen while using Wi-Fi.

1.4.5. 4-Way Handshake Capturing

There is a way to capture the WPA2 handshake transmission process using a packet detection tool such as Wireshark and crack the captured file (packet) with Brutal Force using equipment such as a graphics card.

2. Methodology

2.1. WPA2 Security Method

Other people's papers defending against WPA2 vulnerabilities are representative of changing the key algorithm built into the router and changing or separating the nonce and keys during the transmission process to make it difficult to infer [8,9]. Because there are limits to just changing the key or algorithm, a method to detect intrusions in advance is also used, which is a neural network model, such as the ReLU hidden layer model and the Sigmoid model [10]. In another paper, there is a method to prevent man-in-the-middle attacks or KRACK by using the semiconductor's unique value (PuF) as a signature for each STA (station) [11]. Another paper proposes to strengthen security through user authentication by wallet address using the Bitcoin platform among blockchains [12]. To summarize, the authors of other papers on WPA2 security change the key or algorithm in the transmission process, making it difficult to derive and infer the final key, PTK. However, these methods are solutions that will quickly break if you invest a lot of time and advanced equipment. Overall, the proposals in various papers to protect against cracks are not practical (receiving PuF values from a semiconductor factory) and are not easy (changing the encryption algorithm of WPA2) for general users to use.

2.1.1. PMK (PSK) Security Method

First, using the leaked chipset's PMK should be prohibited and continuously updated through periodic firmware patches to avoid the number of cracks in pre-encrypted expected files.

2.1.2. Pixie Dust (WPS) Security Method

The second way to prevent WPS Attack (Pixie Dust) is to simply disable the WPS push button. However, most router companies ship products with WPS turned on by default.

2.1.3. KRACK Security Method

Lastly, since KRACK exploits the logical vulnerability of WPA2, the keys that induce PTK must be prevented from reinstallation. This vulnerability can be done by changing the algorithm of WPA2 [8] or disassembling and separating the key values in the transmission process [9]. Because the conditions and number of key reinstallation cases for KRACK are very diverse, it is difficult to respond perfectly with firmware security patches alone.

2.2. WPA2 Security Improvements Using NFT (N-WPA2)

The author's ideas were discovered by doing everything in the chapters preceding the methodology text. While dealing with crack tools, thinking about defense methods, and reading various papers, I concluded that Simultaneous Authentication of Equals (SAE), especially used in the paper on WPA3, should be changed to NFT authentication [13]. The author used two main concepts to use NFT as an authentication method. First, in Solidity language, which is mainly used in smart contracts, the mapping type was declared as an array and used as a whitelist concept. There are numerous NFTs in Ethereum, and each NFT is created as a Smart Contract, so it has a Contract Address. This is to allow users of this proposal to add or remove NFT Contract Addresses in a whitelist format to specify the NFT Contract Address to be used for authentication.

The second concept used is Blockchain Event Monitoring. It receives a certain number of blocks (5 to 6) from the latest block in the blockchain and outputs changes to the contracts within those blocks. For example, if A, who has 1 NFT registered in the whitelist, authenticates and then sends the NFT to B, and A's NFT becomes zero, A's qualification must be removed. At this time, monitoring blockchain events is performed by JS code. Monitoring can confirm that A sent NFT to B. This process flow can be written in Solidity code. The name of the proposed approach is N-WPA2.

Figure 9 schematizes the approach of this paper. The AP and Auth Server, enclosed in a dotted box in the upper right corner of the picture, operate on the device (Raspberry Pi) mentioned in Chapter 2.3 The part above the gray line in the middle is the authentication part and the part below is the existing WPA2 part. In other words, NFT authentication was added to the existing WPA2. As explained in Chapter 3.1, most WPA2 cracks can be solved through correct

and secure authentication. However, in the case of Simultaneous Authentication of Equals (SAE) used in WPA3, the router device itself must support the function, so it is difficult to use in existing router devices. This proposal uses NFT as an authentication method while using existing router equipment and WPA2 as is.

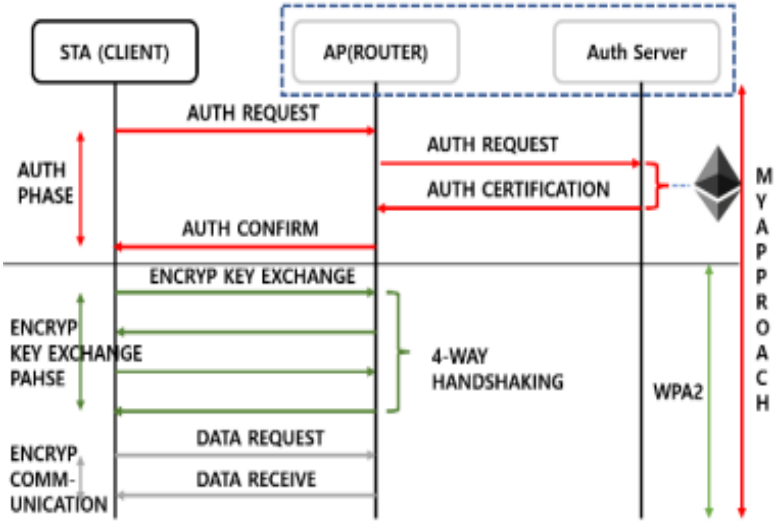

Figure 9: N-WPA2 Process Flow Chart

The implementation of this paper was based on the Windows environment where Python and JS are convenient to operate. Figure 10 is the program flow chart of this paper's approach.

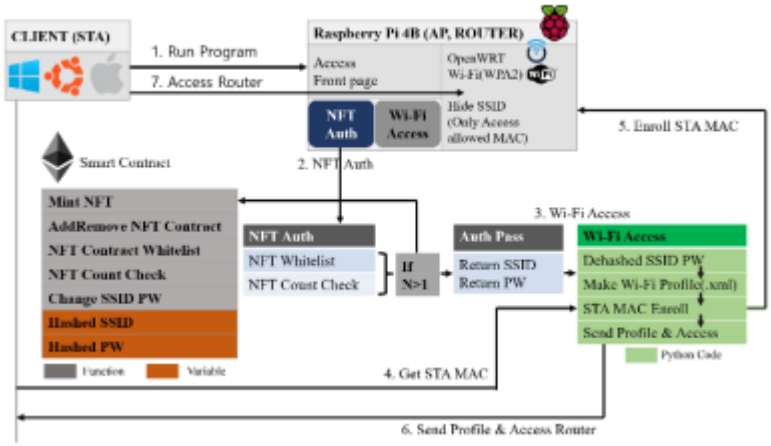

Figure 10: N-WPA2 Program Flow Chart

It is explained according to the flow of Arabic numerals.

(1) The STA device that wants to connect to Wi-Fi runs an executable file (exe) rather than searching for the SSID in the Wi-Fi Search List to connect to the AP. Pywebview, one of Python's modules, was used to create the overall frame of the program.

(2) On the AP side, Raspberry Pi's Open WRT mentioned in Chapter 2.3 is running. For NFT authentication, the server communicates with the blockchain. In this process, JS is used because it supports the most blockchain APIs.

It communicates with the blockchain using the window object, which is JS's DOM (Document Object Module). If the blockchain is Ethereum, window. Ethereum and web3.eth objects are used, and if the blockchain is a Klaytn window. Klaytn and caver objects are used. The developer document page of each blockchain platform indicates which object should be used. The author used Ethereum. First, create a Smart Contract to be used in this system. To call and operate this Smart Contract as a code on a program rather than a web page such as EtherScan, the ABI value of the Smart Contract must be extracted in JSON format. The NFT Mint function is written considering cases where someone does not want to register an NFT address, or when there is no NFT at all. Create an NFT Contract Whitelist that manages the mapping array to check whether the NFT has been registered in the whitelist. In

it, write the Add Remove NFT Contract function to register and cancel, and the NFT Count Check function to count the number of NFTs.

Lastly, the SSID and PW information hashed with SHA256 and salt, which are Wi-Fi information, are stored in a Private Variable (in blockchain, only authorized wallets can call the Private Variable). An SSID PW change function was created to change personal variables. There are two Smart Contracts loaded from the Client (STA) running JS. If it is confirmed that there is more than one registered NFT through the NFT Whitelist and NFT Count Check, the client receives the hashed SSID and PW from the Contract.

The values are passed from
(3) JS to Python through interprocess communication (IPC) between JS and Python. The Python Code side decrypts the hashed values and creates a Wi-Fi Profile.XML file based on it.
(4,5) After creating the XML file, obtain the STA's MAC Address and register it in Open WRT's wireless information.
(6) If the MAC Address is registered in OpenWRT, the generated XML file is delivered to the client and applied to the STA. The STA that received the XML file can connect to Wi-Fi through the profile. At the same time, Blockchain Event Monitoring must be running through JS on the server side. The most important things in monitoring implementation are the eth.getBlockNumber() and eth.Contract.getPastEvent() functions. The getBlockNumber function receives the location of the latest block, and after a certain period (5 seconds in this paper), getBlockNumber is called again to determine the difference in the number of blocks. The blocks are inspected for events that occurred in the contract using the getPastEvent function. Note that to monitor, an NFT must be created that has the function of calling the event of the Transfer function when the NFT is created (or an NFT that has already been implemented must be used). The smart contract in Figure 11 is used in this paper complies with Open Zeppelin's guidelines and is implemented to call events for each function within the contract. To summarize, three codes operate on the server side. 1 Python code to communicate with the client (STA), 1 JS code to authenticate and receive Wi-Fi-related values from the blockchain, and 1 Blockchain Event Monitoring JS code to detect the movement of the NFT.

Algorithm 5 Smart Contract(N-WPA2)

Require: *ERC – 721ofOpenZeppelin*
Input: *STA.Wallet_address*
Output: *Hashed SSID, Hashed PW*

```
Create Function 'Mint NFT'                          ▷ to mint NFT
Create Function 'AddRemove NFT Contract'            ▷ to manage NFT whitelist
Create Function 'NFT Whitelist'                     ▷ Whitelist mapping array
Create Function 'NFT Count Check'                   ▷ to counting STA.NFT
Create Function 'Change SSID PW'                    ▷ to change Wi-Fi Info
if NFT Count Check && NFT Whitelist > 1 then
    Return Hashed SSID, Hashed PW
end if
```

Figure 11: Pseudo Code of N-WPA2 Smart Contract

3. Experiments and Result

A testbed was configured with various device environments to evaluate the degree of security improvement of practical WPA2 through NFT authentication proposed in this paper. Since this paper's proposal operates in a Windows environment, Windows was installed on each device. Typical desktop, typical laptop, Windows 10 in Macbook (BootCamp), LattePanda (SBC).

What these devices (Table 3) have in common is their x86-based architecture. The reason why only the x86 architecture can be used is that Windows with the arm-based architecture does not yet support proper wireless network connection (as of 2023). In addition, because the results of the experiment may differ depending on the performance of the wireless adapter, the same wireless adapter was used. Since the experimental environment is, a product with specifications that were previously released, the connection speed of Wi-Fi6E, which is the latest protocol (2023), and Internet speed may differ. Lastly, in this experiment, Raspberry Pi 4B which serves as a wireless router (AP) and NFT authentication server, and an old router product (A104M model) as the AP experimental control group were prepared. The experiment is structured as follows. Whether it protects against existing crack tools, whether the performance is different from the existing WPA2 by introducing the proposal of this paper, and measures the prevention of signature forgery for NFT authentication and DoS attacks on the authentication server.

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

Table 3: Testbed Environment

3.1. Analysis of Experiment I results

3.1.1 KRACK

First, the security evaluation of KRACK in the major category I experiment in (Table 4). As a result, KRACK is impossible in the environment proposed in this paper. There are two main reasons. Because the SSID is not searched and MAC address-based filtering is performed, network devices with unauthorized MAC addresses cannot even attempt to connect. That means hackers cannot even initiate the key exchange. Figure 12 is the result of running the KRACK tool with the SSID value specified. The value shown by Company I's A104M control group represented the average value of five KRACK successes (15.732sec).

Major Category	Subcategory	Testcase Description
I. Existing Crack Tool Resistance (Evaluation of Security)	PMK(PSK inference Attack)	Assessment of security against Dictionary 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 Practically)	4-way handshake Delay Time	Dealy for 4-way handshake Time Assessment
	Total Auth time	Assessment of total connection time of Approach
	Compare Network Speed	Evaluation of network speed this proposal and existing WPA2
III. New attack methods for NFT authentication (Safety Assessment of Proposal)	Forgery of Signature	Possibility of signature forgery
	Resistance of DoS Attack	Assessment of DoS Safety for Auth Server

Table 4: Experiment Items

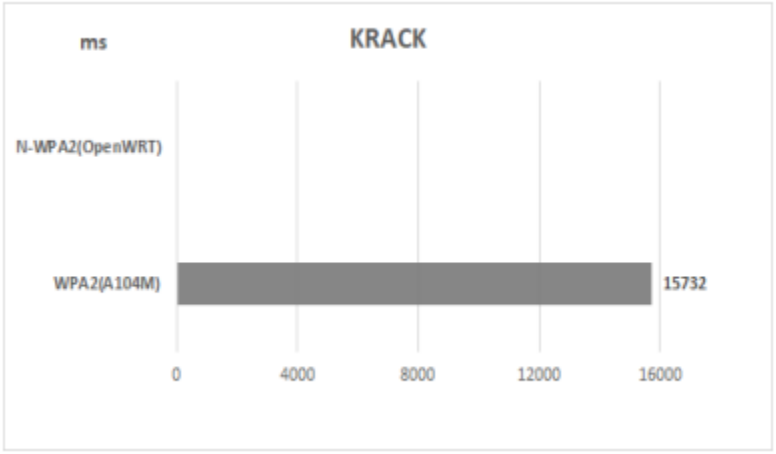

Figure 12: KRACK Time Analysis

3.1.2 WPS

Figure 13 is a security evaluation of the Pixie Dust attack. This paper proposes to turn off WPS by default in Open WRT, but for this experiment, the WPS function was turned on. Likewise, cracks do not occur. The reason is that in Open WRT, the user can change the WPS number arbitrarily, and filtering is performed based on the MAC Address. The value in Figure 13 represents the average value of 5 times. What is unusual is that it took more than 30 seconds to acquire the PIN number in the first WPS attack, but once the PIN number of the target AP was acquired, the WPS attack time was shorter than KRACK (9.245 sec).

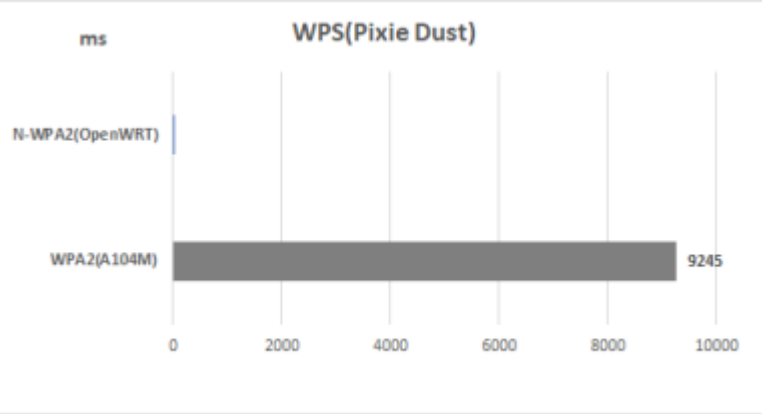

Figure 13: WPS Attack Time Analysis

3.1.3. PMK

Figure 14 is a security evaluation against dictionary password file attacks. As explained at the beginning of Chapter 2.4, five factors are needed to derive PTK. Among them, the ones that can be changed on the AP side are PMK, SSID, and ANonce. In OpenWRT, the SSID and ANonce are randomly reset for each STA connection request, and MAC Address Filtering is processed, so it is not cracked (786sec).

Experimental results show that this proposal can completely protect against three representative cracks of WPA2.

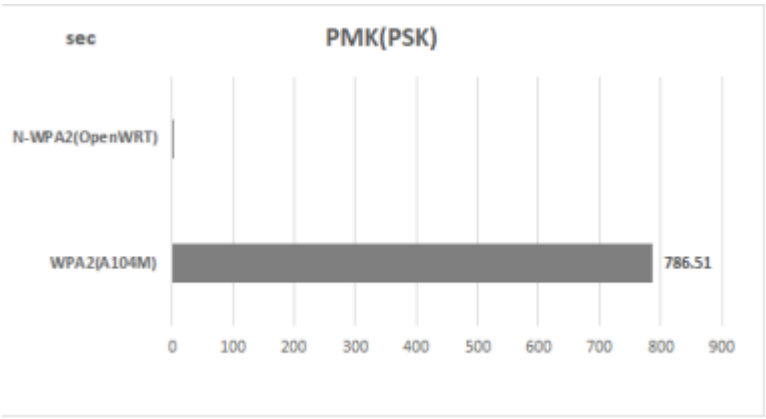

Figure 14: PMK (PSK) Attack Time Analysis

3.2. Analysis of Experiment II Results

In the major category II experiment, experiments are conducted to determine whether this proposal causes any inconvenience in actual use compared to the existing WPA2. The experimental data shows the connection time for the four devices in the above-mentioned Testbed (Table 3).

3.2.1. 4-Way Handshake Delay Time

Figure 15 measures the 4-way handshake delay time. The resulting graph compares the transmission time and installation time of the Wi-Fi Profile.XML file with a typical WPA2 connection. This paper's proposal installs PTK through an XML file, so there is no 4-way handshake, so the PTK installation time and the general 4-way handshake connection time were compared. The unit of experiment is seconds and installing the key through Wi-Fi profile transmission showed faster results overall.

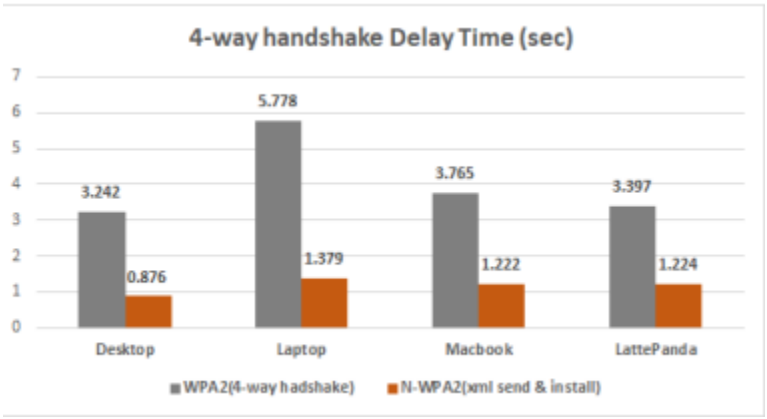

Figure 15: 4-Way Handshake Delay Time

3.2.2. Total Auth (Connection) Time

Figure 16 compares the time taken to connect to N-WPA2 after NFT authentication with Wi-Fi connection processing in Windows as a command. The reason is that even in WPA2, there is time for a person to enter the password, and NFT authentication requires a person to manually authenticate the wallet. Therefore, the comparison was made excluding the time a person inputs. It was written based on the average value of 5 times, and the overall connection time for N-WPA2 in this paper was longer than the existing WPA2 because it takes time to authenticate and call values from the contract in the Web3 environment (Ethereum).

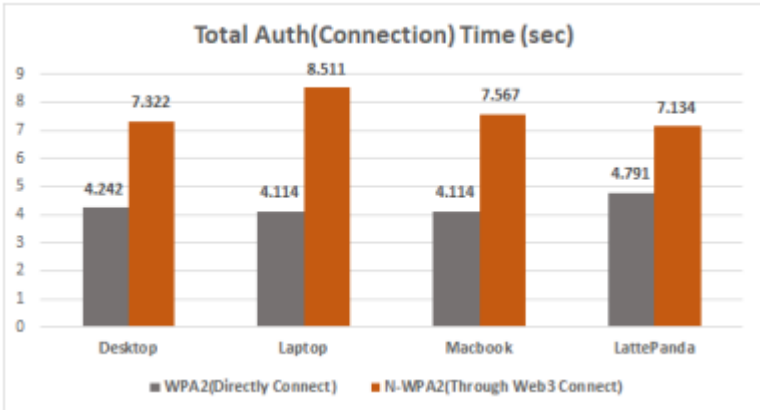

Figure 16: Total Auth(Connection) Time

3.2.3. Network Speed Comparison

Figure 17 compares download time and upload time using an Internet speed measurement tool while connected to Wi-Fi. The Internet experiment environment is a 100Mbps line from Hello Vision, a subsidiary of LG U+ in Korea. Although the Internet environment is poor, all testbed devices in (Table 3) were tested in the same environment. As a result, there is little difference between N-WPA2 and the existing WPA2, or there is a slight increase in network speed thanks to the performance of Open WRT.

The experiment in Chapter 4.2 showed that although the total connection time for N-WPA2 is slightly longer, there is also a reduced time for the 4-way handshake, and there is no significant problem in actual use in terms of Internet speed.

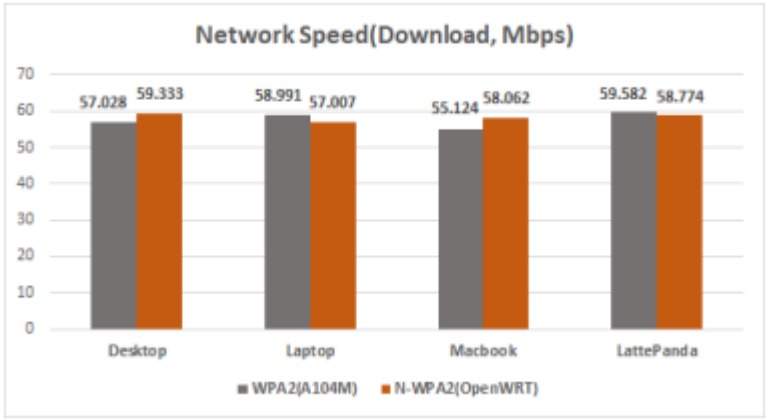

Figure 17: Network Speed Comparison

3.3. Analysis of Experiment III Results

3.3.1. Forgery of Signature

In the major category III experiment, the experiment to evaluate the possibility of signature content forgery is to test whether signing is allowed with the signature part changed in the developer mode of the commonly used Chrome browser.

Figure 18 is a pseudocode that describes how to divide and transmit a wallet signature into V, R, and S, which are used as three components of an ECDSA digital signature. When implementing the authentication page, divide the signature part into V, R, and S as shown in Figure 18. In that case, even if a malicious user enters developer mode in the Chrome browser modifies the signature part, and transmits it, the signature will not be established if either the wallet address or the signature content is incorrect on the server side that receives the signature. After applying the V, R, and S split code on the authentication server web page (HTML), it was confirmed that NFT authentication was not possible with a forged signature.

Algorithm 7 Signature VRS apply

Require: *web3 JS API from cdnjs*

Input: *STA.Wallet_Signature*

Output: *V, R, S of STA.Wallet_Signature*

```
sig = web3.sign(STA.Wallet_Signature)
const v = '0x' + sig.substring(2).substring(128,130)        ▷ sig is cdnjs API
const r = '0x' + sig.substring(2).substring(0,64)
const s = '0x' + sig.substring(2).substring(64,128)
signature = [v,r,s]
Send signature to AUTH server
```

Figure 18: Ways of V, R, and S Signature Coding

3.3.2. Authentication server DoS resistance

As a final experiment, this is a graph comparing how well Company I's product and Open WRT can withstand the same DoS attack Figure 19. As the DoS attack equipment, Xerosploit was used using three Intel Xeon E5-2609 2.4 Ghz Quad-Core, DDR3 8Gb RAM equipment. Xerosploit is a DoS tool that attacks by sending a large amount of meaningless ICMP pings at high speed (Hping). It may vary depending on the attack experiment equipment, but in this experiment, the router using Raspberry Pi showed approximately twice the resistance on average (363 min).

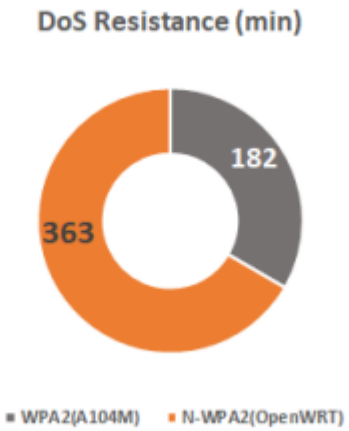

Figure 19: DoS Resistance of Auth Server and Control Group AP

Through the experiment in Chapter 4.3, the risk of signature forgery and falsification of this proposal was evaluated. This paper's proposal was also tested against DoS attacks, which are typical of cyber terrorism, and was shown to be more stable than using products from general router companies.

4. Conclusion

The era is heading towards the era of Web3. Now, people must be able to prove themselves in the Internet space, and in the process, proof of ownership and compensation (incentive) are becoming important. Soon, everyone will be able to prove themselves with NFT or SBT [14]. With the emergence of the term hyper-connected society, most IT devices support Wireless Networks, and the role of Wi-Fi routers for those devices to access the Internet is becoming more important. However, WPA2 is still widely used to universally connect many older devices, and to solve the vulnerabilities of WPA2 mentioned in Chapter 2.4, this paper proposes an NFT authentication WPA2 connection method based on a Windows environment.

This paper was proposed assuming that in the future society; everyone will have at least one NFT. In this paper, the author wanted to emphasize practicality while solving the problems of the existing WPA2. The practicality that the author refers to includes the advantage of practical use of not having to memorize passwords but also excludes the inconvenience of having to prepare something more (upgrading equipment). In this paper, an experiment was conducted: a proposal that proves oneself with an individual's NFT, uses it as an authentication method, defends against crack tools, and has no problems in actual use even if the existing WPA2 is used as is.

In future research, the author plans to implement this system to operate on mobile devices. Currently, mobile devices can be broadly divided into Android and Apple groups. The two coding methods are different, and the method of obtaining Wi-Fi permissions from the mobile OS internal policy is different, so first, the code was written based on Windows, which is easy to write. A means will be provided to create an N-WPA2 app for mobile devices and link it with the Metamask app.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (RS-2023-00259099) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

1. Fehér, D. J., & Sandor, B. (2018, September). Effects of the wpa2 krack attack in real environment. In 2018 IEEE 16th international symposium on intelligent systems and informatics (SISY) (pp. 000239-000242). IEEE.
2. Etta, V. O., Sari, A., Imoize, A. L., Shukla, P. K., & Alhassan, M. (2022). Assessment and Test-case Study of Wi-Fi Security through the Wardriving Technique. Mobile Information Systems, 2022.
3. Eun, T., A. Saad., Park, S. (2022). Introduction to attack methods of locally accessible private blockchain. The Korea Institute of Information Scientists and Engineers.
4. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.
5. Buterin, V. (2015). A Next Generation Smart Contract & Decentralized Application Platform. Ethereum White Paper,@ inproceedings, 1-36.
6. Baray, E., & Ojha, N. K. (2021, April). WLAN security protocols and WPA3 security approach measurement through aircrack-ng technique. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 23-30). IEEE.
7. Nikolov, L. G. (2018). Wireless network vulnerabilities estimation. Security & Future, 2(2), 80-82.
8. Omorog, C. D., Gerardo, B. D., & Medina, R. P. (2018, September). The performance of blum-blum-shub elliptic curve Pseudorandom Number Generator as WiFi protected access 2 security key generator. In Proceedings of the 2nd

International Conference on Business and Information Management (pp. 23-28).
9. Guo, J., Wang, M., Zhang, H., & Zhang, Y. (2020, May). A Secure Session Key Negotiation Scheme in WPA2-PSK Networks. In 2020 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.
10. Thing, V. L. (2017, March). IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In 2017 IEEE wireless communications and networking conference (WCNC) (pp. 1-6). IEEE.
11. Chatterjee, U., Sadhukhan, R., Mukhopadhyay, D., Subhra Chakraborty, R., Mahata, D., & M. Prabhu, M. (2020, April). Stupify: a hardware countermeasure of kracks in wpa2 using physically unclonable functions. In Companion Proceedings of the Web Conference 2020 (pp. 217-221).
12. Niu, Y., Wei, L., Zhang, C., Liu, J., & Fang, Y. (2017, October). An anonymous and accountable authentication scheme for Wi-Fi hotspot access with the Bitcoin blockchain. In 2017 IEEE/CIC International Conference on Communications in China (ICCC) (pp. 1-6). IEEE.
13. Wi-Fi Alliance. (2018). WPA3 AND ENHANCED OPEN: NEXT GENERATION WI-FI SECURITY. ARUBA.
14. Weyl, E. G., Ohlhaver, P., & Buterin, V. (2022). Decentralized society: Finding web3's soul. Available at SSRN 4105763.