American Journal of Networks and Communications
2024, Vol. 13, No. 1, pp. 44-62
https://doi.org/10.11648/j.ajnc.20241301.14

Research Article

Wifi Pentesting Roadmap for Classic-Future Attacks and Defenses

Ramafiarisona Hajasoa Malalatiana, Rakotondramanana Radiarisainana Sitraka*

Telecommunication-Automatic-Signal-Image-Research, Laboratory, Doctoral School in Science and Technology of Engineering and Innovation, University of Antananarivo, Antananarivo, Madagascar

Abstract

The most advanced attack on the Wireless Fidelity (WIFI) network uses social engineering. The hacker makes portal captive and forces the victim for disconnecting to internet instead of entering the real password of the WIFI. In normal actions, asking WIFI password on the web interface is not the real process, but sometimes the victim is not experience enough on security and thinks that it is a technical problem. Also, the victim didn't have internet connection due to the hard deauthentication and the select open access, which is not his WIFI network. The future generation of WIFI could be use a secure deauthentication. So, this article proposed how the actual attack will be processed, how is the secure deauthentication and how hacker could use this same attack with more secure network. Like conclusion, solutions to resolve this problem will be proposed. New hacking arsenal for replacing the deauthentication is the smart-jamming. With the secure deauthentication, reforging the packet for telling the victim to deauthenticate to the network will not be possible anymore. The smart-jamming select the frequency of the access point of the victim and jam only this specific frequency by sending a noise. In this scenario, the same effect of the first attack is still possible. For the best security of network, two solutions will be proposed: secure deauthentication and hopping frequency. A defensive proposition about secure deauthentication will be found in this article by using cryptographic key exchange like Diffie Hellman (DH), Elliptic Curve Diffie Hellman (ECDH) and Super Isogenies Diffie Hellman (CSIDH).

Keywords

CSIDH, Deauthentication, ECDH, Smart-Jamming, Wifi

1. Introduction

The classic attack on WIFI network is the Wired Equivalent Privacy (WEP) cracking password [1-7] and Personal Identification Number (PIN) cracking using Wi-Fi Protected Setup (WPS). The two options are not used in most case on the WIFI network. The security uses actually Wireless Application Protocol (WAP) privacy. The pentesting on WAP consists of the word list password attack. The hacker captures the four-way handshake and use offline cracking to test all possibilities of password on this word list. If the password of the victim is secure enough and the victim changes his password regularly, this attack is not interesting. Actually, the advanced Denied of service over WIFI attack is possible by reforging a packet for telling the victim to deauthenticate to the network. The Institute of Electrical and Electronics Engineers (IEEE) norm doesn't specify a secure deauthenticate and doesn't verify if it's the real network or a fake network

*Corresponding author: radiarisainanasitraka@yahoo.fr (Rakotondramanana Radiarisainana Sitraka)

Received: 20 February 2024; **Accepted:** 5 March 2024; **Published:** 20 March 2024

which sent this reforge packet, and it tells the victim for disconnecting to the WIFI network. Actually, the modern attack uses the two scenarios: the first step is to capture the four-way handshake to verify if password that the victim entered is the real one and use deauthenticate and fake open access point having a similar name for phishing the victim. Indeed, the hard deauthenticate will be repeated to the victim on the real network for having a denied of service.

If the victim selects the fake access point, a web interface asks the victim to enter the password and the four-way handshake permit to verify if the password captured is correct or not. After that, the connection to internet of the victim will be established by stopping the denied of service and the hacker captures this WIFI password.

2. Methodology and Scenario Attack

2.1. Hardware Setup

In these experiments, phone is used as access point (Android Phone access point) and one victim which is connected to this real access point. A physical machine contains Windows 10 as a victim and the hacker machine on virtual machine VMWARE with Kali 2022. The physical machine and hacker machine have a specification on the Table 1.

Table 1. *Computers specifications.*

Processor	**Intel (R) core (TM) i5-10300H CPU @ 2.50Ghz**
RAM	32Go
Operating System	Windows 10
Virtual Machine VM Ware	Workstation Pro 16.2.3 build-19376536
RAM of hacker virtual devices	4Go
Processor of hacker virtual devices	4Cores (2 Processors + 2 Threads)
Hard Drive of hacker virtual devices	120Go

For the classic scenario of deauthentication, two WIFI networks cards which could make injection packet mode attack will be used.

Table 2. *Network card specifications.*

WIFI Card	**RT5572**
WIFI band	2.4Ghz – 5Ghz – 5.8Ghz
Injection mode	Yes
Antenna	6dbi
Multiple Input Multiple Output (MIMO)	2T2R

For the future scenario, Software Defined Radio (SDR) also used as a programmable radio frequency hacking. The victim and hacker are not anymore on Windows, but on a real operating system using DragonOS R26 public. The DragonOn has a preinstalled Gnuradio 3.8 with the script of the smart jamming attack.

2.2. Classic Attack

This classic attack will be demonstrated on the studies of Michael et al. (2020) and Arockiam (2010) [8, 9]. This attack will divide on two steps:

1. Capturing the four-way handshake
2. Capturing the password by using the evil twin

2.2.1. Capturing the Four-way Handshake

WPA/WPA2 uses a pre-shared key named Pairwise Master Key (PMK). The four way-handshake permits having a mutual authentication generated with PMK and create for each

session a Pairwise Transent key (PTK). The exchange between the supplicant (client) and Authenticator (Access Points) has four messages named message 1,2,3,4 on the Figure 1. Using a hashing function named Key Derivation Function (KDF), the equation (1) permits us to have the PMK.

$$PMK = KDF(Passphrase, SSID, SSIDLen, RepetitionKDF, Keylength) \quad (1)$$

Passphrase: Password between 8 until 63 characters

SSID: Service Set IDentifier (SSID) of the authenticator

SSIDLen: Length of the SSID

RepetitionKDF: Number of the iteration of the hash function

Keylength: the length of the output of the hash function (in our case 256bits)

The PTK could be evaluated using the equation (2):

$$PTK = KDF(PMK, Authenticator\ Nonce\ or\ ANonce, Supplicant\ Nounce\ or\ SNonce, Authenticator\ Media\ Access\ Control\ (MAC), Supplicant\ MAC) \quad (2)$$

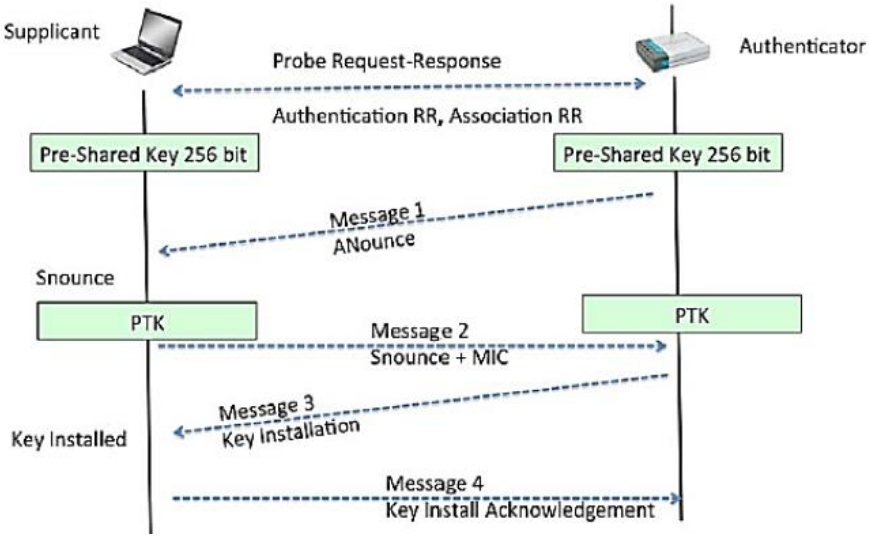

Figure 1. *Four-way handshake.*

Figure 2. *Presentation of Fluxion's requirement.*

For capturing the four-way handshake and realizing all classic attack, the Kali 2022 will be install by using software like Fluxion illustrated in the Figure 2.

For the first steps of the attack, the handshake will be captured for verifying if the password captured by the victim is true or not by the captive portal. For this, the second option handshake snooper should be selected like on the Figure 3.

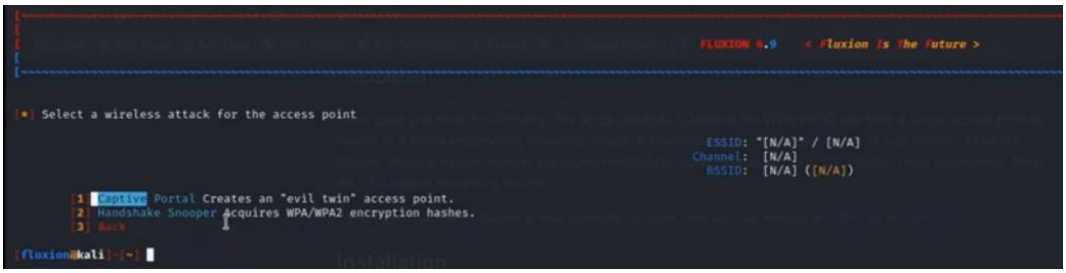

Figure 3. *Presentation of Fluxion's attacks.*

After the scanning, the access point should have a victim connected to this. For the previous selection, it's possible to scan all band or one specific band for having a good result for any specific channel, like on the Figure 4.

Figure 4. *Scanning of network.*

For having the four-way handshake, the victim should be deauthenticated on the WIFI network. In this scenario, air-play tool permits the deauthentication. It is possible to use cowpatty verification. If the handshake is not obtained for the first 30 second, it will be resent until it is being obtained. With low budget, hacker should use synchronous options instead, it's possible to use asynchronous budget. If the hacker obtained the handshake, the attack should be success on the Figure 5:

Figure 5. *Success captured handshake.*

2.2.2. Capturing the Password by Using the Evil Twin

This step will be divided in four steps like on this Figure 6:

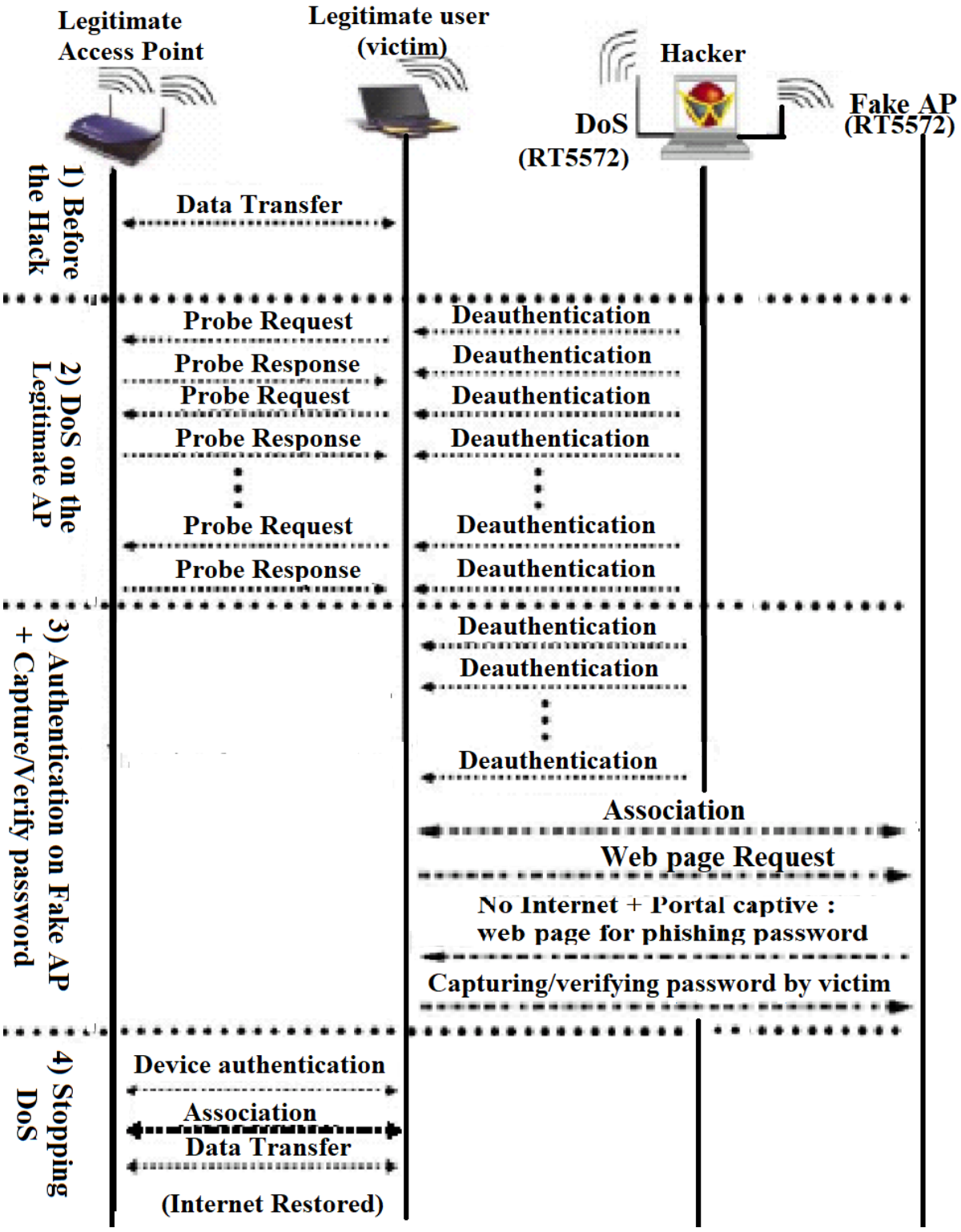

Figure 6. *Classic attack flow.*

Step 1: For this classic attack hacker, the victim connects to the legitimate access point. The hacker uses two RT5572 WIFI cards: one for deauthentication and on another for the fake access point.

Step 2: The hacker sends deauthentication, to the legitimate as a victim user and accepts this deauthentication. The victim doesn't have internet connection and will search WIFI network. With more attempt, due to the deauthentication, the victim could never have an internet connection and think that it's a problem on the network and select the another one (which is open and without internet connection).

Step 3: The victim associate to the fake access point and send a web page request. The fake access point with web responds with portal captive and ask the victim to enter the login and password for restoring internet connect (the victim shouldn't do it).

Due to the captured four-way handshake, it is possible to verify if the password entered by the victim is correct or not.

Step 4: the victim stops the deauthentication and re-establishes his internet connection.

2.3. Future Attack

On the future plan, the norm of WIFI should mitigate on the secure deauthentication. Like on authentication, a similar process as the four-way handshake should be done. In this, the legitimate user doesn't deauthenticate to the legitimate access point if only, if the hacker has a password of the legitimate access point. Even, the hacker send denied of the service, the legitimate user doesn't accept it because hacker doesn't have the password [10-14].

Also, the hacker couldn't have also the four-way handshake and couldn't verify if the password entered is correct or not. The Denied of service is also possible. Some Mikrotik WIFI enterprise client still has this option to protect to deauthentication. To enable this on Mikrotik client, the legitimate user should activate "Management Protection" like on the Figure 7.

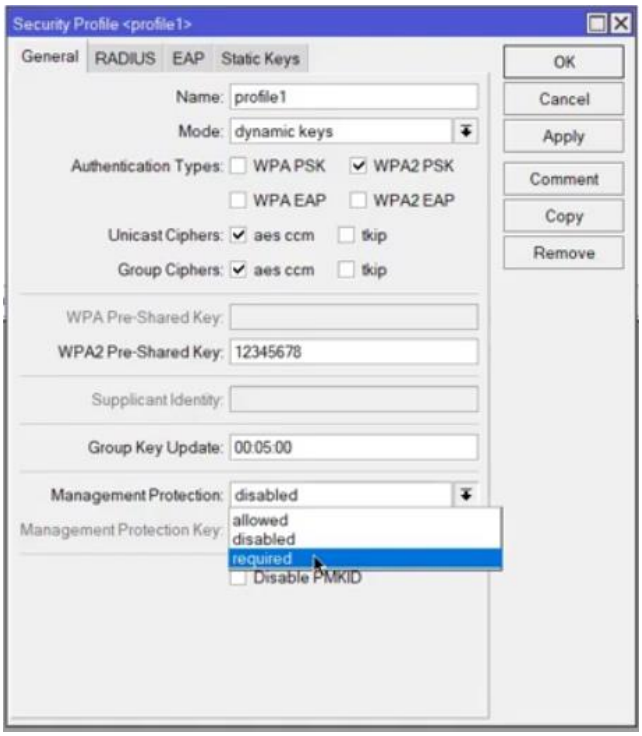

Figure 7. *Deauthentication protection on Mikrotik.*

Table 3. *WIFI channel specification.*

Channel	Frequency (GHz)
1	2.412
2	2.417
3	2.422
4	2.427
5	2.432
6	2.437
7	2.442
8	2.447
9	2.452
10	2.457
11	2.462
12	2.484

The WIFI needs radio frequency for transmitting information, and it is still possible to jam by sending random message on this frequency. In this, the hacker should transform the channel to the specific frequency like on this Table 3:

This table show only the frequency on 2.4 GHz, but a specific Table is possible for the 5GHz and the WIFI 6.

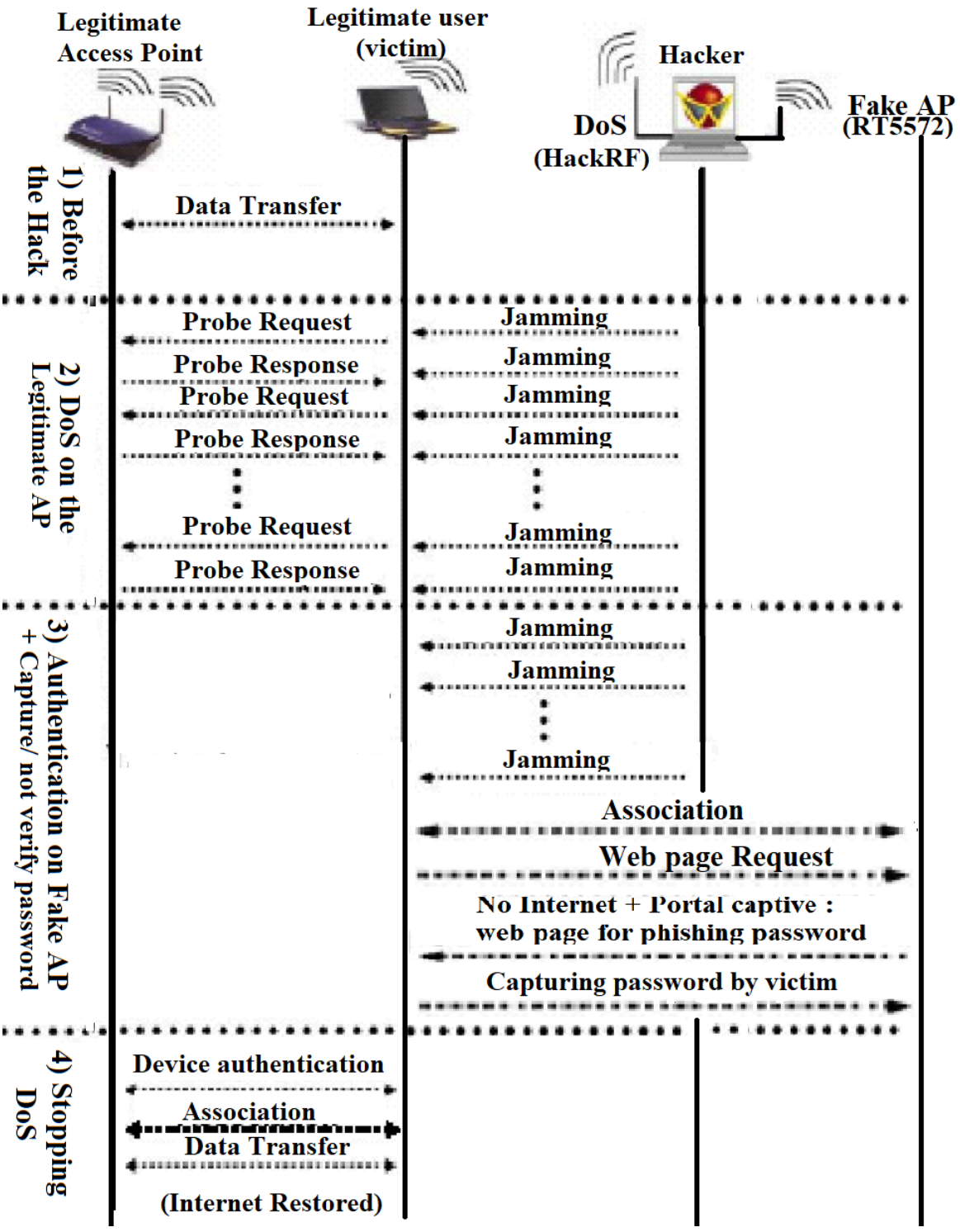

Figure 8. *Future attack flow.*

Step 1: For this future attack by hacker, the victim connects to the legitimate access point. The hacker uses one Hackrf for deauthentication and one RT5572 for the fake access point.

Step 2: The denied of service is realized with the HackRF. Using Gnuradio flow graph like the Figure 9, it is possible to jam the channel with bandwidth chosen. The denied of service here is more aggressive because, it jams all channel and could jam also another SSID than the victim.

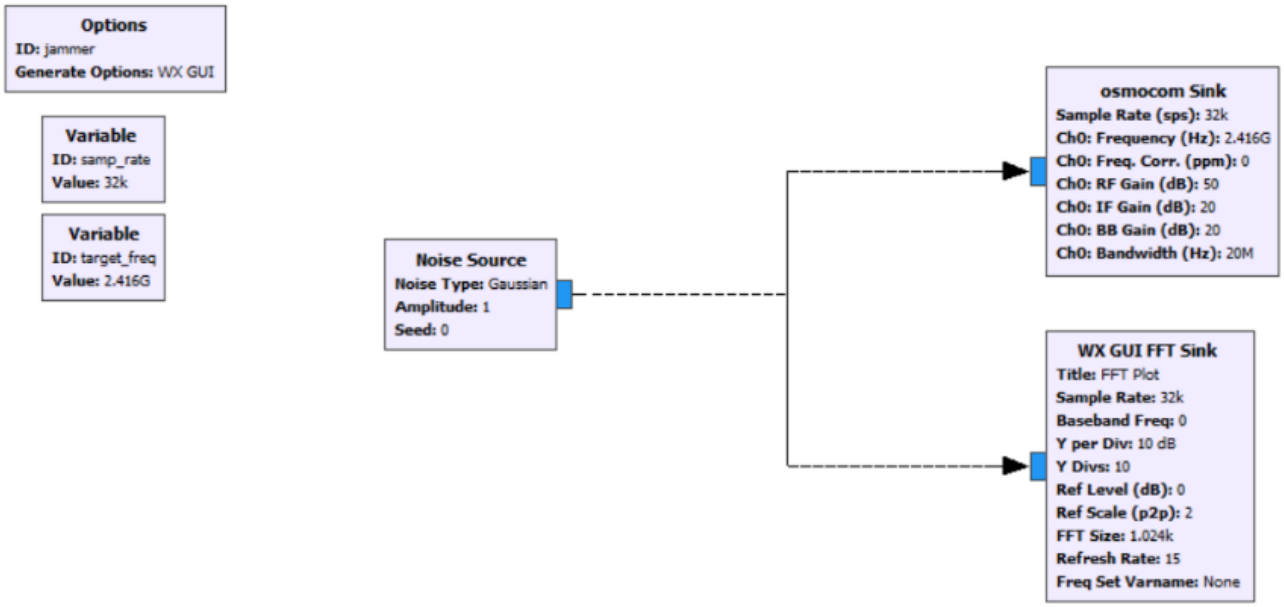

Figure 9. *Schema bloc of jamming by Gnuradio.*

Step 3: As same as the classic attack, the difference is that it's not possible anymore to make verification of the victim password

Step 4: As same as the classic attack, the victim restored his connection to internet by stopping the denied of service attack by the jamming using SDR.

As the classic attack, it is possible to capture the password without verification but the deauthentication will be replaced by the jamming using software defined radio.

2.4. Classic Defense

The classic defense actually uses Wi-Fi Protected Access version 3 (WPA3) uses cryptographic key exchange as Elliptic Curve Diffie–Hellman (ECDH) on the four-way handshake.

Diffie-Hellman Protocol

The Key exchange research begins by the fundamentals of discrete logarithm problem found by Whitfield Diffie and Martin Hellman name Diffie-Hellman (DH) in1976. This protocol is based on the Figure 10.

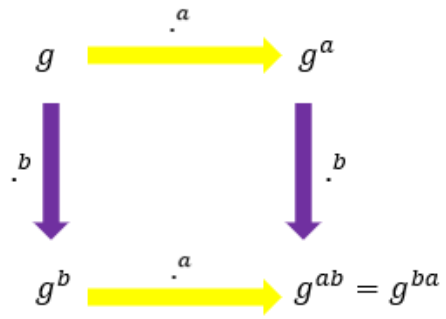

Figure 10. *DH protocol.*

In this figure, the yellow arrow is the computation done by Alice.

The purple arrow is the computation done by Bob.

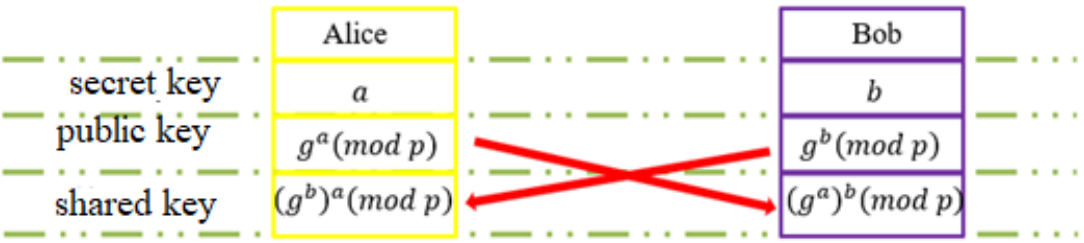

Figure 11. *DH key exchange.*

Before beginning the key exchange as like as on Figure 11, the two parts Alice and Bob start with a public parameter as the number prime p and the root primitive g which is very big. The secret part of Alice and Bob will be the number a and b; and the public part of Alice and Bob will be $g^a\ (mod\ p)$ and $g^b\ (mod\ p)$.

The shared key of Alice will be calculated by the private key of Alice and public key of Bob: $K = [g^b\ mod\ p]^a mod\ p$

The shared key of Bob will be calculated by the private key of Bob and public key of Alice: $K = [g^a\ mod\ p]^b mod\ p$

This protocol could be explained by the Schreier Graph. This graph is often used in the context of the Schreier-Sims algorithm to find a Schreier basis for a given group.

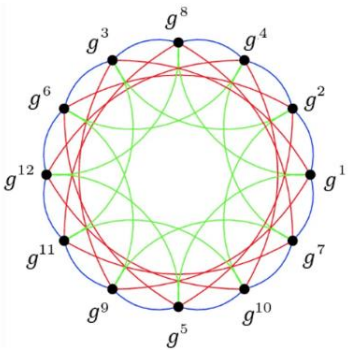

Figure 12. *Example of Schreier Graph for p = 13.*

The schreier Graph non oriented at $\langle g \rangle \{1\}$ where $g^{13} = 1$, act in $(\mathbb{Z}/\mathbb{Z}13)^*$ generated by $S = \{2, 3, 5\}$

If Alice takes for example a = 2 as secret key, her public key is g^2 and Bob takes for example b = 5 as secret key and his public key is g^5 with $p = 13$.

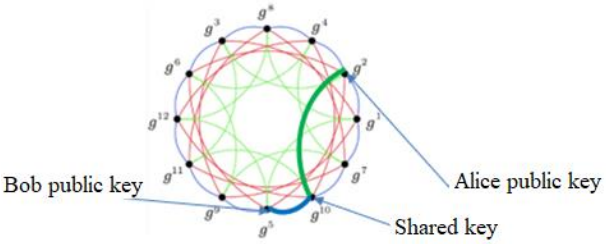

Figure 13. *Computation in Schreier Graph for a= 2; b = 5 and p = 13.*

There are two types of keys used during: secret key encryption and public key. After the key exchange, Alice and Bob compute the same shared key like as Figure 13.

Elliptic Curve Diffie-Hellman (ECDH)

The Protocol Diffie-Hellman will be performed using Elliptic curve defined by:

$$y^2 = x^3 + ax + b \tag{3}$$

Elliptic curve calculations, especially in the context of elliptic curve cryptography (ECC), include several essential operations.

Here are some calculation operations performed on elliptic curves: addition, multiplication by scalar, neutral element and morphism

The addition of two points P and Q on an elliptic curve produces a third point R. The addition of points is the fundamental operation which makes it possible to construct an Abelian group on the elliptic curve. The law of composition determines how to do this, and it can vary depending on the specific representation of the curve.

The arithmetic concerning the addition of two distinct points $P: (x_1; y_1)$ and $Q: (x_2; y_2)$ on this elliptic curve is defined as follows [15-19]:

The slope of the secant line passing through the two points is calculated by: $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$

The coordinates of the point R= P+Q are calculated using the equation (4):

$$x_R = \lambda^2 - (x_1 + x_2)\ and\ y_R = \lambda(x_1 - x_R) - y_p \tag{4}$$

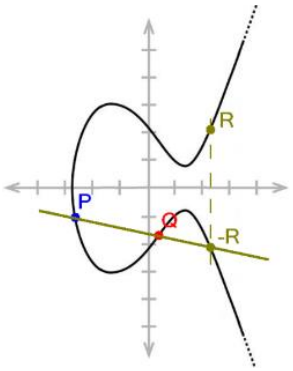

Figure 14. *Addition on Elliptic Curve P+Q=R.*

Graphically: Here are the steps to add two points P and Q:

Find the line passing through the two points P and Q

Find the point R, different from P and Q, intersecting the curve E

The result of the sum is the symmetric of R with respect to the abscissa axis.

Firstly, this law allows all the elements of the group to be added together, and to fall back on an element of the body, it is therefore a law of internal composition. Then, the curve is symmetrical with respect to the abscissa axis. The third step therefore necessarily brings us back to a point on the curve.

The doubling operation is a special operation which consists of adding a point to itself. For example, 2P is the result of the doubling operation of P. The doubling operation is

commonly used to speed up point addition calculations.

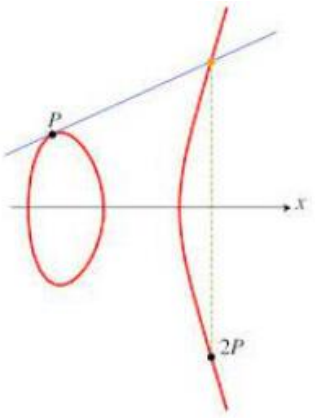

Figure 15. *Doubling point 2P.*

The slope of the secant line passing through the two points is calculated as follows: $\lambda = \frac{3-y_1}{x_2-x_1}$

The coordinates of the point R = 2P are obtained by the equation (5):

$$x_R = \lambda^2 - 2x_P \; and \; y_R = \lambda(x_P - x_R) - y_P \quad (5)$$

Graphically to add the sum of two identical points, P=Q is obtained by take the tangent of the curve on this point, and by carrying out the same addition operations. That is to say, the point R is the intersection between the curve and the tangent, such that different of R. The result is the symmetry of R with respect to the abscissa axis.

Scalar multiplication consists of multiplying a point P by an integer k to obtain a new point Q = kP. This operation is commonly used in ECC to generate public and private keys, as well as to encrypt and decrypt data. The result of scalar multiplication is usually another point on the elliptic curve, and this point is used in applications such as key generation in elliptic curve cryptography. The number of iterations depends on the value of the integer you use for multiplication. The larger this integer, the further the final result will be from the starting point along the curve.

A neutral element must be defined in order to be able to deal with a group, it is the neutral element of this group, according to the group law. This element, in the case of elliptic curves, is named ∞. This point can be found graphically as the sum of a point on the curve and its symmetry with respect to the abscissa axis. Clearly, this is one of the two cases where the curve passing through two points of the curve does not intersect it at a third point.

This neutral element is therefore the point that this curve crosses at infinity, whether on the positive or negative side. The two elements are noted by +∞ and -∞ are considered as one and the same point. Another method makes it possible to obtain the neutral element as a result of the group law between two points, by adding the same point twice, if it is on the abscissa axis.

Morphism between two elliptic curves E_1 and E_2 is an application defined by the equation (6):

$$\varphi: E_1 \to E_2 \text{ and } \varphi(P+Q) = \varphi(P) + \varphi(Q) \; where: \varphi(P+Q) \; is \; in \; E_1 \; and \; \varphi(P) + \varphi(Q) \; is \; in \; E_2 \quad (6)$$

The exchange key will be like on the Figure 16 and Figure 17. Let an elliptic curve E over $\mathbb{F}_p$ where p is a prime and $P \in E$ like a public parameter. As like on DH, ECDH uses secret key and public key to calculate shared key.

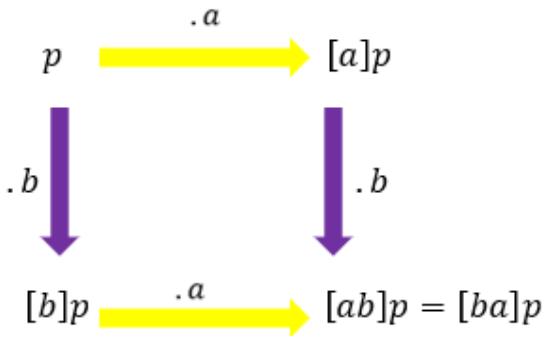

Figure 16. *Group commutative ECDH.*

The yellow arrow is the computation done by Alice.
The purple arrow is the computation done by Bob.

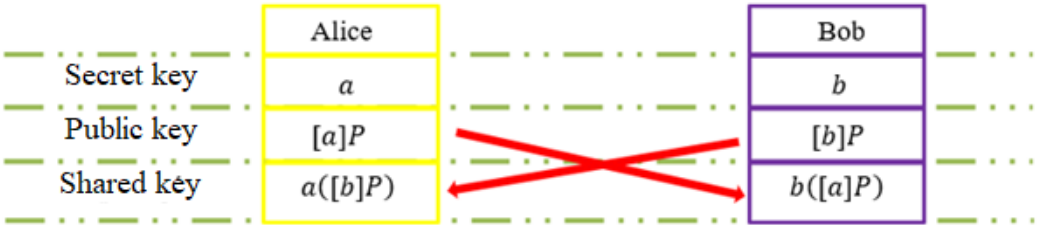

Figure 17. *Exchange key ECDH.*

The security of this key exchange protocol relies on the difficulty of two computational problems. The elliptic discrete logarithm problem requires finding the integer a when P and [a]P are given. The Diffie-Hellman computational problem requires finding [ab]P when P,[a]P and [b]P are given.

The example of Schreier Graph of ECDH is illustrated on Figure 18.

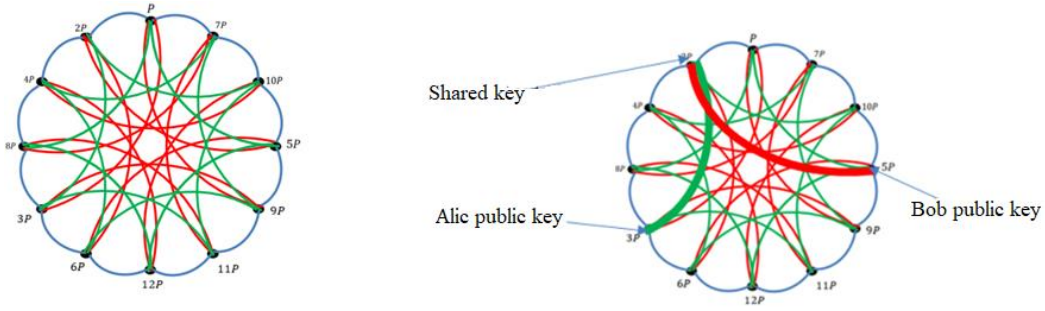

Figure 18. *Example of Schreier Graph of ECDH with a =3; b= 5 and p = 13.*

On this Schreier Graph, $\mathbb{F}_p = 13$ and $S = \{2, 3, 5\}$

Let choose the secret key of Alice is a = 3 and his public key is $[3]P$ and for Bob the secret key is b = 5 and the public key is $[5]P$. After computation of shared key, Alice and Bob will be in the same point [2]P

2.4. Post quantum defense

Commutative Super Isogenies Diffie Hellmann (CSIDH)is an isogeny-based Diffie-Hellman protocol using supersingular curves defined on $\mathbb{F}_P$ and commutative group action, which is quantum resistant key exchange. Isogenies are a morphism of elliptic curve.

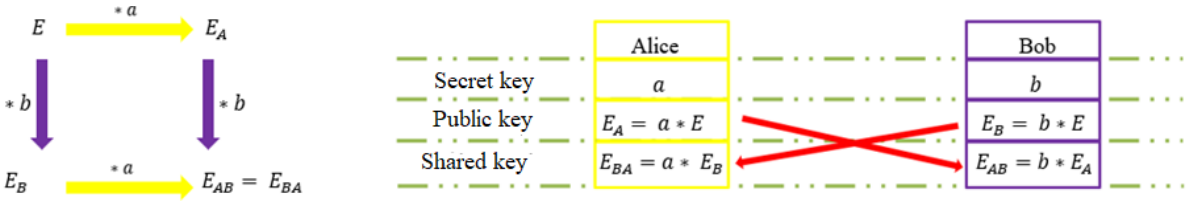

Figure 19. *CSIDH key exchange.*

First, Alice and Bob take a secret subgroup a and b of the Elliptic curve E

Second, Alice calculates groups act as multiple isogenies $* a: E \to E_A$ and Bob calculates group act $* b: E \to E_B$

Afterward, Alice and Bob exchange the two public keys E_A and E_B

After receiving the two keys, Alice calculates E_{BA} while Bob calculates e E_{AB}

Let gives example of Schreier Graph CSIDH illustrate on Figure 20.

In the Figure 21, E_A is a supersingular elliptic curve defined by: $y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$

The edges denote $\{3, 5, 7\}$ −isogenies, and given (clockwise) by the action of $h_3, h_5, h_7 \in$ H.

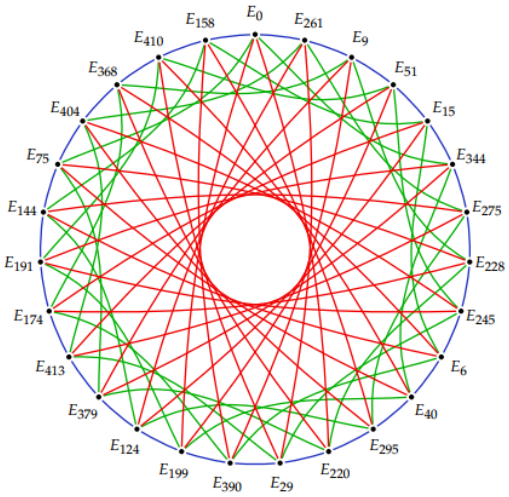

Figure 20. *Schreier Graph for CSIDH.*

The good mixing properties of the underlying isogeny graph are relevant for the security of isogeny-based cryptosystems. The graph above which serves as an example is a Schreier graph associated with our class-group action and the chosen generators. One view on this is that one can move quickly between two elliptic curves by using isogenies properties in the subgraph corresponding to one generator by switching to the subgraph corresponding to another generator. This thus replaces the square-and-multiply algorithm in cryptosystems based on exponentiation.

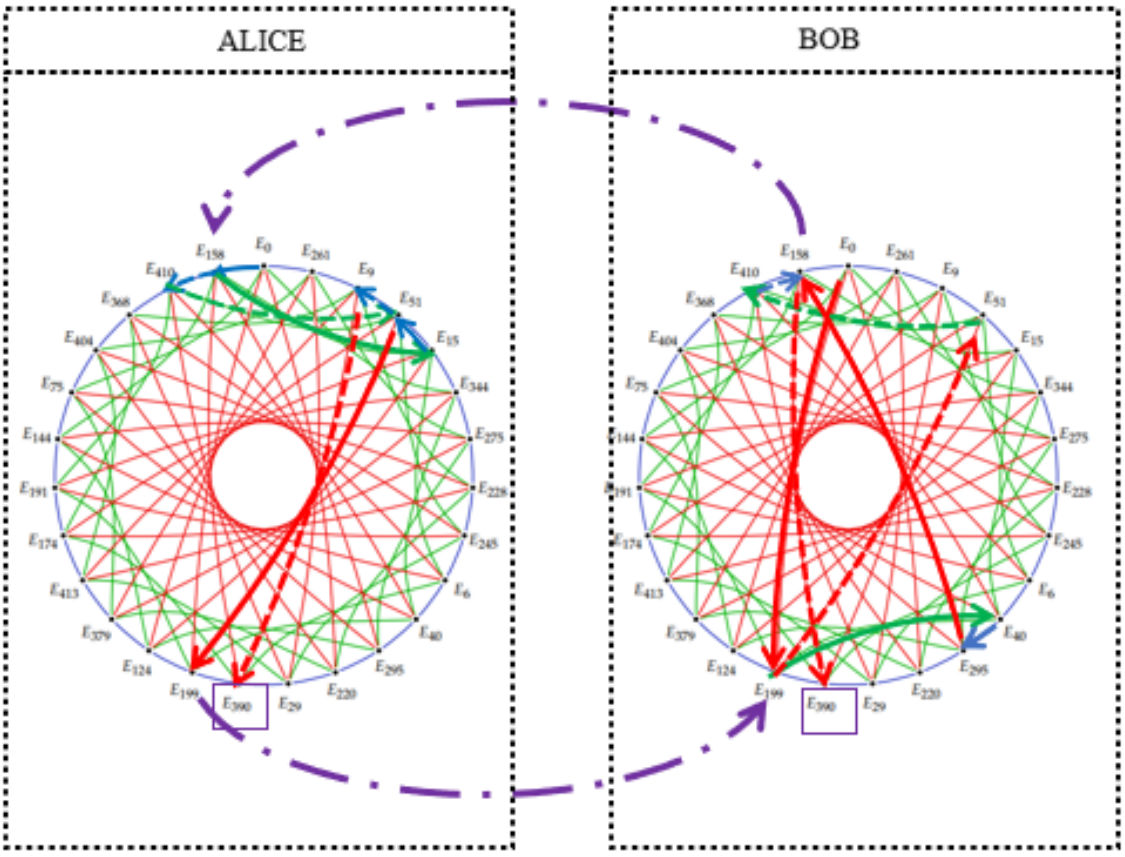

Figure 21. *Example of Group action with CSIDH.*

Where:

: first calculation by Alice and Bob

: public key exchange between Alice and Bob

: second calculation after exchange of public keys

$$E_A = [+,-,+,-] = (h_5^{-1} * h_3 * h_7^{-1} * h_3) * E_O$$

$$E_B = [+,+,-,+] = (h_5 * h_3^{-1} * h_7 * h_5) * E_O$$

$$E_{AB} = E_{BA} = (h_5^{-1} * h_3 * h_7^{-1} * h_3 * h_5 * h_3^{-1} * h_7 * h_5) * E_0 = E_{390}$$

The parameters of the example above are chosen, an elliptic curve E_0 on a finite body $\mathbb{F}_{419}$.

The participants in the key exchange are called Alice and Bob. They each choose an integer which are respectively $a = [+,-,+,-]$ and $b = [+,+,-,+]$ and which are used as private keys. Alice calculates and publishes $E_A = a * E_O$ as the public key. Bob proceeds in the same way and calculates his public key$E_B = b * E_O$. After public key exchange, due to the commutativity of the action of elliptic curve groups, both parties can now calculate $E_{AB} = E_{BA} = a * E_B = b * E_A$, which is then used as a shared secret.

3. Results and Discussions

3.1. Classic Attack

After launching the attack, the victim couldn't access to internet connection due to the deauthentication and the new fake access point with the same name will be created. [20-22] The Figure 22 shows the result of this.

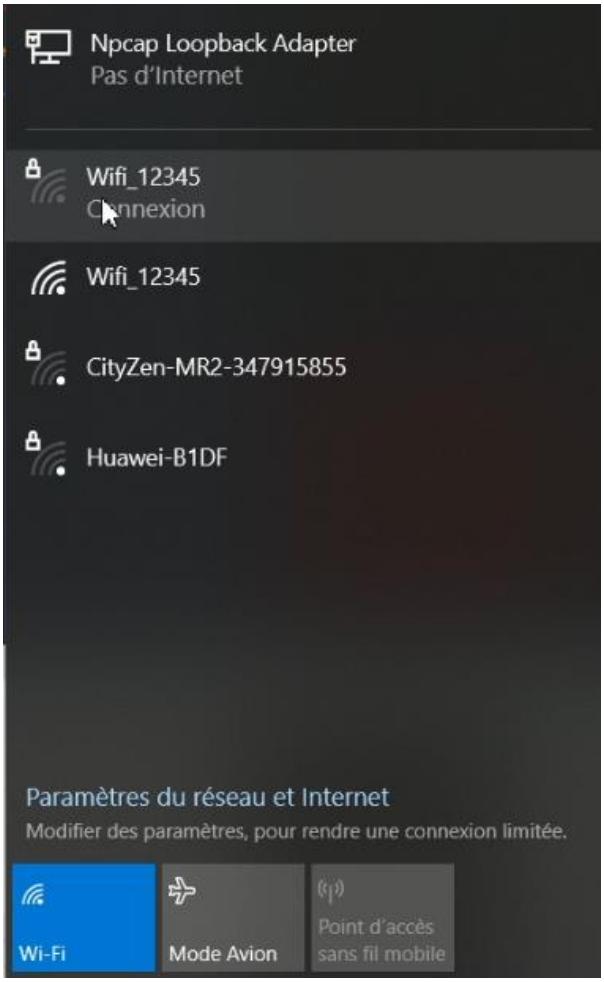

Figure 22. *Fake access point.*

If the victim connects to the fake access point due to aggressive denied of service on the real network, the web portal captive will be appearing for asking users to restore the internet connection like on Figure 23.

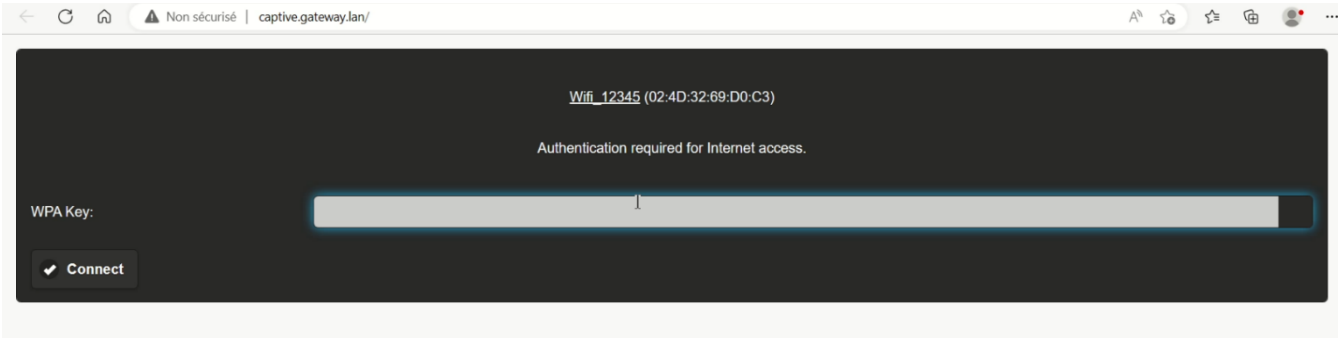

Figure 23. *Deauthentication protection on Mikrotik.*

Due to the handshake captured, the web could verify if the password entered by the victim is corrected or not. The two web interfaces Figure 24 and Figure 25 show this. When the user enters a real password, the fake access point will be stopped and the victim will connect to the real network directly.

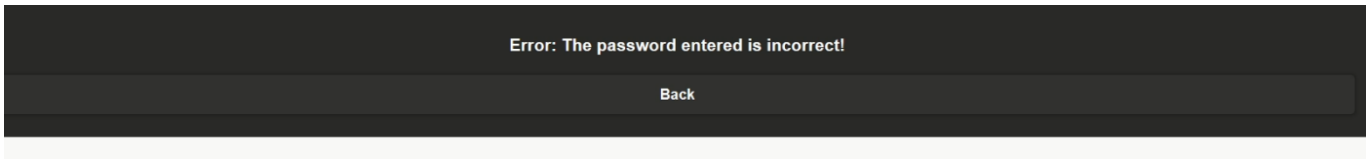

Figure 24. *Password error entered.*

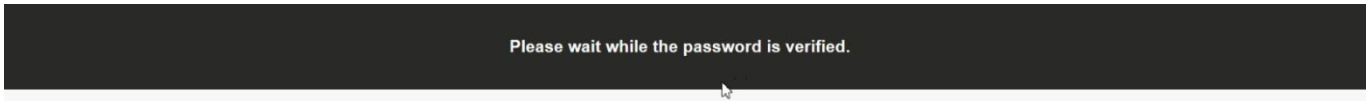

Figure 25. *Password success entered.*

The final step is to see the password obtained by the victim will be restored on the path attacks/captive portals/name_wifi like on this Figure 26.

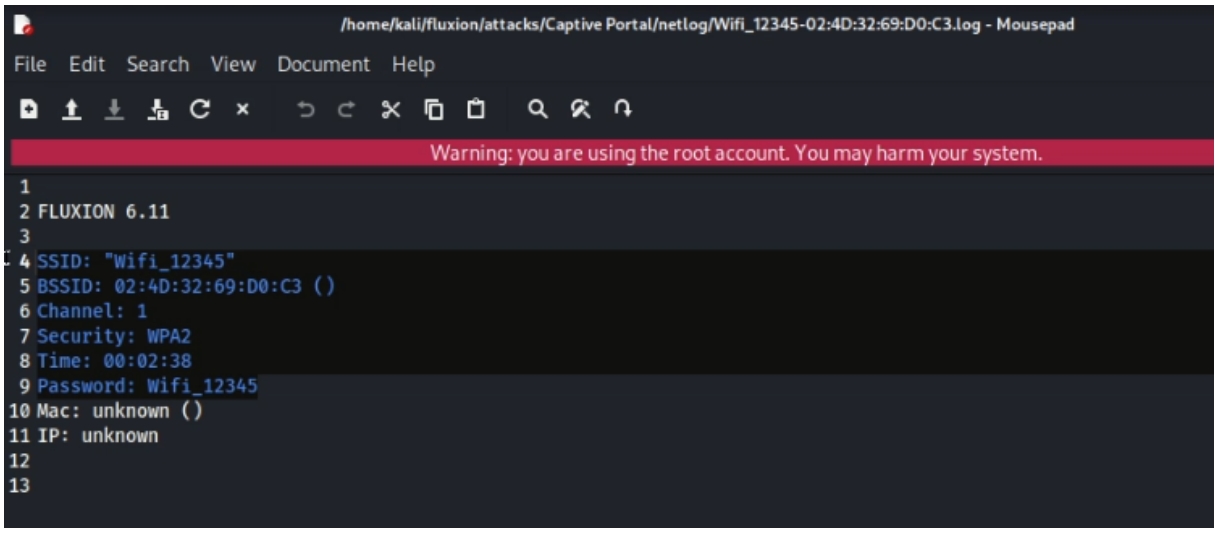

Figure 26. *Log captured password.*

3.2. Future Attack

As a classic attack, the future attack uses the same process but replaces the deauthentication by the jamming [23-24]. This section presents the results of the jamming. By using the jamming with SDR and HackRF like devices, the victim doesn't have connection Internet anymore, like on the Figure 27.

After stopping the attack, the victim reestablished internet connection.

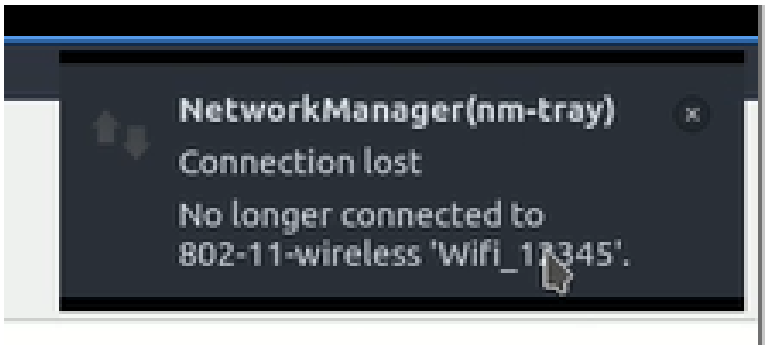

Figure 27. *Jamming effect with SDR.*

3.3. Classic Comparing to Future Attack

Table 4. *Comparisons between classic and future attack over WIFI.*

	Classic attack	Future attack
Feasibility	Simple deauthentication	Secure deauthentication
Dos Effect	Selective jamming (only target) No disruption with other SSID	Jamming all targets Disruption with other SSID having the same channel
Cost	Cheap	Expensive
Complexity	Simple	Complex
Verification of password captured on phishing	Yes	No

The Table 4 shows the difference between classic attack and future attack. The simple deauthentication is not possible for the secure deauthentication by using jamming. So, this future attack is more disrupt and not possible to make a selective denied of service. The classic attack is more simple and cheaper, it's possible to make the attack in a simple device like ESP8266 or ESP32XX, but the future attack needs Software Defined Radio which is more expensive.

3.4. Classic and Future Defense

The Table 5, 6 and 7 show the different key exchange which is possible to use over WIFI. The demonstration and installation is provided in the link at the references [25-26].

Table 5. *DH key exchange.*

Alice private key: alice.priv

0xb0572e96131d2b3bccaca9ee5e95feb1d497acfc78968416c0ea5a4c695a17a43bab5b6596cd4789442b371f0c4090131f0f40fce3f01328c60c33b05ddfb156a286c971cc6884ffee01da30181a865dbc73526ff2c18911812dc641f7e68098b6f528b5d1d4f6ccd4608ebe074a5608739d627dacd1e04dbede6d652f119fdbe801e389b6f36f6a238c851ca62214f1bda350f6eb7c2f7e58f40d288099e543cb548d1a15d81df70ea0ca8c48a72643c23db4df6d72171449c69e4dcfbdd03196787814e5802d36be9402b71f7c84299f36eaaa2940f71113c7f3eb34eab3fa7b181ca03c318f47982b82e8980c82d983f8f5bf6377de7a1f6b6c64fa71a54d

Alice public key: alice.pub

0xb98325655307e9d2a48494a434ba62ce8c3a58d5902bc7c238dc3b1ff93f3ff1542fa833f3944f86fe477fec8a29201afbe6f5270d0fe2de4b224785aa2506e05f01b1158dad4094e4100958289ba56a86f7c82b8488e6aff913ba574f82a45592d1638060d97b19767725a6adbb78e7f5c1acbdc28991483ceef1719bc7dac0a7bbf1e1218f7cfc326f72cfda58be3caf556f4f458bf5c78e1ba127a24a004b6ef8cce759d46a3c3d5f4b9d5d241b97514513e33b500ab7f24bbacf3a6974a008567ffaef7da5428b6d465820b15f59f0878dac65d826129007a5790c5d18a0cc305f25f530998075f4fc792609ede1d3d7cfd0638036690c750e0cc498764

Bob private key: bob.priv

0x43928225562c57ffa837ea0764eedd3545bf8fe08700889e116e27a3e566b5ccde35b09ae9dafa38946507c4241b392fcb3699fb37aab1dd91d38cc9254f8d83cc5ced06a02e7eb99e910d01bb6a0a5c1f023b7b1051e73a20c6a18d39feff2655c4847fc10cab250365ff2c53d207fb7b819f12623884997aa30d0ac05ea4ff98464c102c10fa9f6472ed85ca1d3fef1523e2ed3bd226b1db8f79af88f406371564651d338674f31ffc5a65a878bfb3ce6e4d4cf5223c86f17ab1c3c69e1535db7298f86e7db934ca00bcc1a5f17d5f3604bb984914fc705e603b63b81ab1b5fdb64b1fb192b964972b45fab34e01605d616c401a115666d0cdfea688eaf1ad

Bob public key: bob.pub

0x56eb84dd55efd976520d6e59ffe870495def898432005033af759e078491faa750a1688fe091fdd2aa0aa04fb5524e068b8a163a0f77696d77b0bf45889d974e390eef5fc78228f95f839d2aa42273fd4d9448b90bb568e402e90b89f906bab353ad25258a34b512d7532d827a88c5962542325faf27e24b7c30b56e8453f19ed576539ea2b5c9bc1abb58c3720ebe04b8b439a7ce4663ca997178053050475155105 7fc5305fb1cd2a8335e17017d978ddae7048c96633520d8f13833cc0537d6ff510f27a465bb83b89a0173fda81a39c69e1e491c5ea9199802bb951c11725071e994d8f5dd3ec744e05cc2ecfa29b2addeb5f3cf219a44e6eefd74a7ec4b

Shared key: key1.key and key2.key

0xd0aa14e6a41177850b170ab13413c46aae620426fb80399c944480113dc99fef3ae28134c410b159ab4080b77aeb0cab4d9116377c91e21a36787775958fd06d3e83f3afcf1d9ac27c05dba5eaa70a883fe73fbe4218edb3b1ff5a5da1ff4506e17949220c8352f6c54238fe311c086137867462de930520c4851d41892ad46e14c9744f3c4e6c2cbb8725931b22179d763e64de22dbfcce39ffa5a4ad52a030b4f21ff88af4e7054dfbf84fb7c86a16400567d6ff6f136c31fb4ab4bbd03bb6b25a3c6b4e14b31c78a80dd1a2f740da43e5eb7f69b08c95439e673abac3dd7bb0f21e9bce8e4de9eb17482b22829fc1f499a848f9272a6ebba3247096b80baf

After Alice generates private key then public key named alice.priv then alice.pub by using command keygen_dh.py -n alice; and does the same operation for Bob to have bob.priv then bob.pub. The Table 5 resumes all key generated by the algorithm DH. By using the command sharedkey_dh.py, the Figure 28 shows that shared key is equals for alice and bob part.

```
root@ubuntu:/home/cryptoandwifi/SRC# python3 shared_dh.py -p alice.pub -s bob.priv -k key1.key
shared key
0xd0aa14e6a41177850b170ab13413c46aae620426fb80399c944480113dc99fef3ae28134c410b159ab4080b77aeb0cab4d9116377c91e21a36787775958fd06d3e83f3afcf1d9ac27c05dba5eaa70a883fe73fbe4218edb3b1ff5a5da1ff4506e17949220c8352f6c54238fe311c086137867462de930520c4851d41892ad46e14c9744f3c4e6c2cbb8725931b22179d763e64de22dbfcce39ffa5a4ad52a030b4f21ff88af4e7054dfbf84fb7c86a16400567d6ff6f136c31fb4ab4bbd03bb6b25a3c6b4e14b31c78a80dd1a2f740da43e5eb7f69b08c95439e673abac3dd7bb0f21e9bce8e4de9eb17482b22829fc1f499a848f9272a6ebba3247096b80baf
root@ubuntu:/home/cryptoandwifi/SRC# python3 shared_dh.py -p bob.pub -s alice.priv -k key2.key
shared key
0xd0aa14e6a41177850b170ab13413c46aae620426fb80399c944480113dc99fef3ae28134c410b159ab4080b77aeb0cab4d9116377c91e21a36787775958fd06d3e83f3afcf1d9ac27c05dba5eaa70a883fe73fbe4218edb3b1ff5a5da1ff4506e17949220c8352f6c54238fe311c086137867462de930520c4851d41892ad46e14c9744f3c4e6c2cbb8725931b22179d763e64de22dbfcce39ffa5a4ad52a030b4f21ff88af4e7054dfbf84fb7c86a16400567d6ff6f136c31fb4ab4bbd03bb6b25a3c6b4e14b31c78a80dd1a2f740da43e5eb7f69b08c95439e673abac3dd7bb0f21e9bce8e4de9eb17482b22829fc1f499a848f9272a6ebba3247096b80baf
root@ubuntu:/home/cryptoandwifi/SRC#
```

Figure 28. *Evaluating shared key using DH.*

Table 6. *ECDH key exchange.*

Alice private key: alice.priv

0x12de0040d1efa8e14a088fa09e864fd9f4f51aa8a3ca7f87a0028e5043f6d966

Alice public key: alice.pub

0x3d5703e7d124f265c109181a6b630f03f9f76a1b009d652884861ec6a49969ca68d30e1f109bdbd55d029f6b82a3eb77d1935e483bb40f11b11c1e9a3b5b6b2

Bob private key: bob.priv

0x8f6c63f70ed708f32ed796ecb4013dffb2aafc6a0dc2b6d0ac51548684900f1bpriv

Bob public key: bob.pub

0x7fa870533fac1b47a6afe9cb9bb7f277f6580731332eda8842ff4001d11445853a3fe1a366d724b904296f69bc9e2b6889cb32e8ac4f814d046d472627d41e71

Shared key: key1.key and key2.key

0x7f3ec1795e6fbc3a43c2a4b09c34087389168dcd93de8fb967658359a5a2aeeb0x50bcc0fa7e801e2402810ce6abb8bfb8766102ca5a5cf4cd20647d95d2db80b7

After Alice generates private key then public key named alice.priv then alice.pub by using command keygen_ecdh.py -n alice; and does the same operation for Bob to have bob.priv then bob.pub. The Table 6 resumes all key generated by the algorithm ECDH. By using sharedkey_ecdh.py, the Figure 29 shows that the shared key is equals for alice and bob part.

Figure 29. *Evaluating shared key using ECDH.*

Table 7. *CSIDH key exchange.*

Alice private key: alice.priv

0x03fe02ef010a12000800fa0f02141303f3fff301efff17f9f80e02090406fd02ff0b09f9fbf70906080408fffe0606fa040702fcfcfafdfbfb0301fd01010301fcfe04fe0004020400fd

Alice public key: alice.pub

0x50f89ef9a81a2d3bdfe849b74d82cb1eb6287a8cba63c8dd77b8011f2777f97a1adee779bf5263ac0992a24a9152e5d4c2c63706a8ef6689ef83c3b2ea786413

Bob private key: bob.priv

0xff0600f9f100fc02fe08f6010a080b0bf50dff030df911f9f406f80df4f6fffe03f7ff01fdfd09060604fe01fa0600000607fe06fa0203fb03030501fd01fdfd02fefcfc04fe04fc0401

Bob public key: bob.pub

0xa7189f8303d91997ec4d313c07e6826512c2c570475c74616fd27d1cde0a4e6b2fdfda787231941500d7f59b4c056da0d6a0f898487efa5315fb02ff7d9e844d

Shared key: key1.key and key2.key

0x0e230be22d7c1bee26f984ad50ef6602af2760dc69ec9912a2b0635dc7f20a6168e3d8301a75b0ff2f97c9b8203e8f561cbe014eadd45cab45dfc3cbd6ad6d44

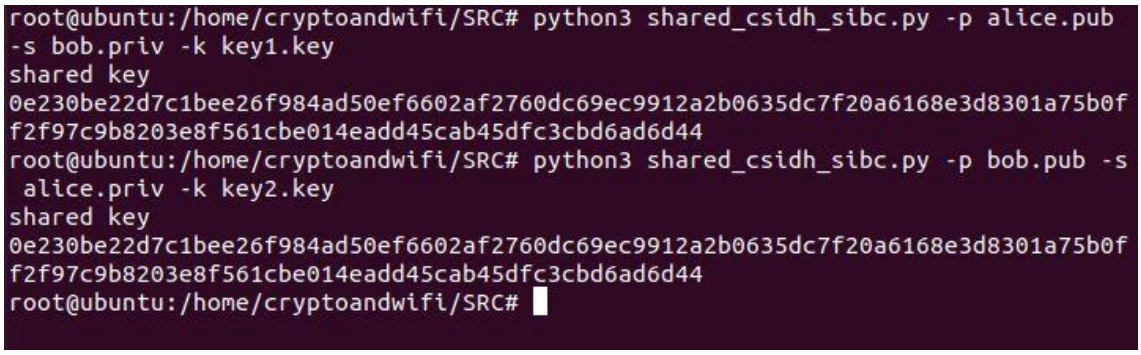

Figure 30. *Evaluating shared key using CSIDH.*

For the post-quantum part, API SIBC like on the reference is used [27]. After Alice generates private key then public key named alice.priv then alice.pub by using command keygen_csidh_sibc.py -n alice; and does the same operation for Bob to have bob.priv then bob.pub. The Table 7 resumes all key generated by the algorithm CSIDH using API SIBC. By using sharedkey_csidh_sibc.py, the Figure 30 shows that shared key are equals for alice and bob part.

4. Conclusion

The WIFI is vulnerable by the classic attack which is cheap, simple, and selective denied of service and the future attack using SDR which is more disrupt, expensive and more complex. Indeed, these two attacks are more dangerous combined with portal captive. The feasibility of the attack of WPA3 and

the security of this problem are demonstrated. The cryptographic solution consists of using key exchange like DH, ECDH and CSIDH. This paper shows how to install and simulate the DH and ECDH algorithm. In the future, it's more secure to use CSIDH algorithm, which is quantum safe. This cryptographic solution protects to the classic attack by using secure deauthentication and the hopping frequency protects to the future attack based on jamming. As defenses, CSIDH cryptography should be implemented on real WIFI card in the future for improved direction by patching code at the driver.

Abbreviations

CSIDH: Commutative Super Isogenies Diffie Hellman
DH: Diffie Hellman
ECC: Elliptic Curve Cryptography
ECDH: Elliptic Curve Diffie Hellman
IEEE: Institute of Electrical and Electronics Engineers
KDF: Key Derivation Function
MAC: Media Access Control
MIMO: Multiple Input Multiple Outpu
PIN: Personal Identification Number
PMK: Pairwise Master Key
PTK: Pairwise Transent key
SDR: Software Defined Radio
SSID: Service Set Identifier
WAP: Wireless Application Protocol
WEP: Wired Equivalent Privacy
WiFi: Wireless Fidelity
WPS: Wi-Fi Protected Setup

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S Vinjosh Reddy, K Sai Ramani, K Rijutha, Sk Mohammad Ali, CH. Pradeep Reddy (2010), Wireless hacking: a WiFi hack by cracking WEP,. *IEEE - International Conference on Education Technology and Computer, ICETC,* *https://doi.org/10.1109/ICETC.2010.5529269*

[2] Lee Barken. Eric Bermel, John Eder, Matthew Fanady, Michael Mee, Marc Palumbo, Alan Koebrick, (2004) Wireless Hacking: Projects for Wi-Fi Enthusiasts. *Syngress,* ISBN: 978-1-931836-37-1.

[3] He-Jun Lu & Yang Yu (2021). Research on WiFi Penetration Testing with Kali Linux, *Hindawi,* https://doi.org/10.1155/2021/5570001

[4] Matthew Denis, Carlos Zena, Thaier Hayajneh. (2016). Penetration testing: Concepts, attack methods, and defense strategies. *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*). https://doi.org/10.1109/LISAT.2016.7494156

[5] Adrian Dabrowski Georg Merzdovnik, Nikolaus Kommenda, Edgar Weippl.(2016). Browser History Stealing with Captive Wi-Fi Portals. *IEEE Symposium on Security and Privacy Workshops (SPW)*. https://doi.org/10.1109/SPW.2016.42

[6] Pragati Shrivastava, Mohd Saalim Jamal, Kotaro Kataoka,.(2017). EvilScout: Detection and Mitigation of Evil Twin Attack in SDN Enabled WiFi. *IEEE Transactions on Network and Service Management.* https://doi.org/10.1109/TNSM.2020.2972774

[7] Kevin Bauer, Harold Gonzales, Damon McCoy. (2008). Mitigating Evil Twin Attacks in 802.11. *IEEE International Conference on Performance, Computing and Communications (IPCCC)*. https://doi.org/10.1109/PCCC.2008.4745081

[8] Michael Kyei Kissi, Michael Asante (2020). Penetration Testing of IEEE 802.11 Encryption Protocols using Kali Linux Hacking. *International Journal of Computer Applications.*

[9] L. Arockiam Lawrence, Vani. B L.(2010). A Survey of Denial of Service Attacks and it's Countermeasures on Wireless Network. *International Journal on Computer Science and Engineering.*

[10] Lakshmi R, Aanchal Sharma, Bhuvan S, Chinmay B (2022). Comparative Analysis of Security and Privacy Protocols in Wireless Communication. *International Journal of Computer Trends and Technology*. https://doi.org/10.14445/22312803/IJCTT-V70I10P102

[11] Nirmal S Selvarathinam, Amit Kumar Dhar, Santosh Biswas. (2019). Evil Twin Attack Detection using Discrete Event Systems in IEEE 802.11 Wi-Fi Networks. *Mediterranean Conference on Control and Automation (MED)*. https://doi.org/10.1109/MED.2019.8798568

[12] Sandesh Jain, Sarthak Pruthi, Vivek Yadav, Kapil Sharma. (2022). Penetration Testing of Wireless EncryptionProtocols. *IInternational Conference on Computing Methodologies and Communication (ICCMC).* https://doi.org/10.1109/ICCMC53470.2022.9754042

[13] Jean Pierre, (2021), Mikrotik prevent deauthentication attacks With English subtitles, https://www.youtube.com/watch?v=QDsSekMIHOw

[14] Ivan Palam, Francesco Gringoli, Giuseppe Bianchi, Nicola Blefari Melazzi. (2022). The diverse and variegated reactions of different cellular devices to IMSI catching attacks. *WiNTECH'20: Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*. https://doi.org/10.1145/3411276.3412191

[15] N. Mehibel, A new approach of elliptic curve Diffie-Hellman key exchange, *Conférence internationale sur le génie électrique - Boumerdes (ICEE-B) Octobre 2017.*

[16] D.Hankerson,S.Vanstone, Guide to elliptic curve cryptography, *Springer 2004.*

[17] L.C.Washington Elliptic curves number theory and cryptography, *Chapman & Hall /CRC, 2003.*

[18] W. Castrck T. Lange C.Martindale L.Panny J.enes, CSIDH: An Efficient Post-Quantum Commutative Group Action, *Oxford PQC Workshop, 22 March 2019.*

[19] M.Meyer S.Reith, A faster way to the CSIDH, *University of Applied Sciences Wiesbaden, Germany 2018*.

[20] Sitraka Rakotondramanana, Malalatiana Ramafiarisona, https://drive.google.com/file/d/16M6iERCNrvr0hoqmWh3NuHvZMLr-SoEy

[21] S. Atluri and R. Rallabandi, (2021). Deciphering WEP, WPA, and WPA2 preshared keys using fluxion, *in Smart Computing Techniques and Applications. Singapore: Springer, 2021, pp. 377–385*, https://doi.org/10.1007/978-981- 16-0878-0_37

[22] Duc Tran Le, Thong Trung Tran,, Khanh Quoc Dang,Reem Alkanhel, Ammar Muthanna, (2022), Malware Spreading Model for Routers in Wi-Fi Networks, *IEEE Access*, https://doi.org/0.1109/ACCESS.2022.3182243

[23] Sitraka Rakotondramanana, Malalatiana Ramafiarisona, https://drive.google.com/file/d/16_w3aUbm0CcDcRJnvYbs1iqcdMGWIdMb

[24] George Chatzisofroniou, Panayiotis Kotzanikolaou. (2019). Association Attacks in IEEE 802.11: Exploiting WiFi Usability Features. *Socio-Technical Aspects in Security and Trust: 9th International Workshop, STAST 2019, Luxembourg City*. https://doi.org/10.1007/978-3-030-55958-8_6.

[25] Francisco Rodr ́guez-Henr ́quez, SIBC: A Python-3 library for designing and implementing efficient isogeny-based protocols, *isogenyschool, September 17, 2021*, *https://isogenyschool2020.co.uk/schedule/is_FRH.pdf*

[26] Sitraka Rakotondramanana, Malalatiana Ramafiarisona, https://drive.google.com/file/d/1_2yZoJXqHAPeWei-oo3aA71wCmrOWwIu

[27] Sitraka Rakotondramanana, https://github.com/SitrakaResearchAndPOC/cryptodome