KAPASA MAKASA UNIVERSITY

School of Applied Sciences and Education

Bachelor of Science Cyber Security

ICT Department

Details

STUDENT NAME	**:** James Soko
STUDENT NUMBER	**:** 20231392
YEAR OF STUDY	**:** 3rd
COURSE NAME	**:** Wireless Security
COURSE CODE	**:** CYS331
LECTURER	**:** Mr. Zilani Kaluba
TASK	**:** Assignment 1
DUE DATE	**:** 23rd June, 2025

QUESTION

Design a simulation program for a 4-way handshake process (AP and Client) with the inputs

a) SSID
b) Password
c) Client Mac Address [Validate the format of the physical address] Should be in the format (8C:04:BA:14:5B:A3)
d) AP Physical Address (8C:04:BA:14:5B:A3) [Validate the format of the physical address] Should be in the format (8C:04:BA:14:5B:A3)

Clearly display the message keys exchanged and give sample Keys and show the keys which match for the handshake to be a success.

Table of Contents

Table of Figures

Introduction

The 4-way handshake is the process of exchanging 4 messages between an access point (authenticator) and the client device (supplicant) to generate some encryption keys which can be used to encrypt actual data sent over Wireless medium.[1]

To help us understand this process will utilize the tool called sokoFy developed by Soko James. It is written in C/C++ Programming and you can access all the files on the GitHub page including the source codes.

sokoFy

Is a lightweight educational simulator that visually demonstrates the WPA2 Wi-Fi 4-way handshake process. Designed for cybersecurity and networking students, it helps you understand how authentication and key exchange work between a client (STA) and an access point (AP) in a secure wireless network.

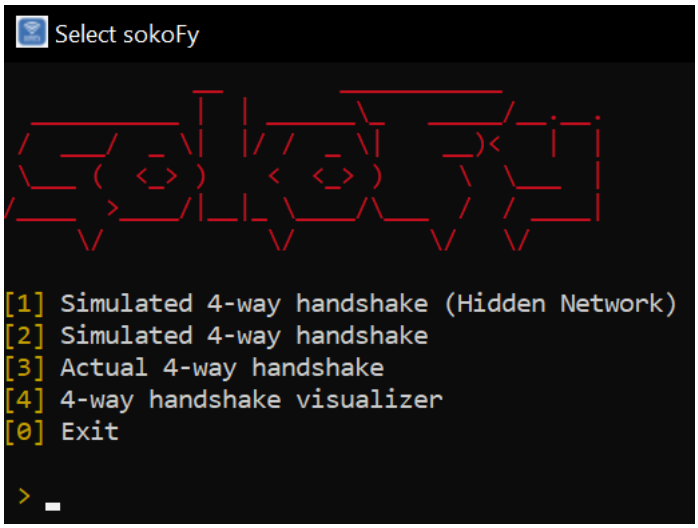


Figure 1: sokoFy tool menu

The 4-Way Handshake Phase

The 4-way handshake is essential in the IEEE 802.11iprotocol, aiming to verify that the access point is legitimate to generate the PMK.[2]

1. Authenticator Nonce

   The authenticator (Access Point) generates a fresh nonce, called the **ANonce**, and together with a replay counter (i.e., a counter used to protect the receiver against replay attacks) sends it to the supplicant (Client).[3]

---

[1] WiFi Professionals , 2019. 4-Way Handshake. [Online]
Available at: https://www.wifi-professionals.com/2019/01/4-way-handshake
[Accessed 21 June 2025].

[2] Alabdulatif, A., Ma, X. & Nolle, L., 2013. Analysing and Attacking the 4-Way Handshake of IEEE 802.11i Standard. s.l.:s.n.

[3] Cremers, . C., Kiesl, B. & Medinger, . N., 2018. A Formal Analysis of IEEE 802.11's WPA2: Countering the Kracks Caused by Cracking the Counters. s.l.:s.n.

2. Supplicant Nonce

The supplicant generates its own fresh nonce, the **SNonce**, and uses a key derivation function to derive the Pairwise Transient Key (PTK) from the pre-shared secret (PMK) and the two nonces. Then, the supplicant sends the SNonce and the replay counter it received in message 1 to the authenticator. Additionally, to allow the authenticator to verify the integrity of the message, it appends a message integrity code (MIC) computed with the PTK.[3]

3. Temporal Key (GTK) + MIC

After receiving message 2, the authenticator also derives the PTK and checks its message integrity code. It then encrypts the **GTK** and together with an incremented replay counter and a MIC (also computed with the PTK) sends it to the supplicant.[3]

4. Acknowledgment

When the supplicant receives message 3, it checks the message integrity code. In case the check is successful, it installs the GTK and the PTK. To confirm to the authenticator that the installation was successful, the supplicant uses the PTK to compute a MIC for the replay counter of message 3 and sends both the replay counter and the MIC back to the authenticator. At this point, the authenticator also installs the pairwise transient key and the handshake is complete.[3] We can visualize this using option 4 in sokoFy as shown below;

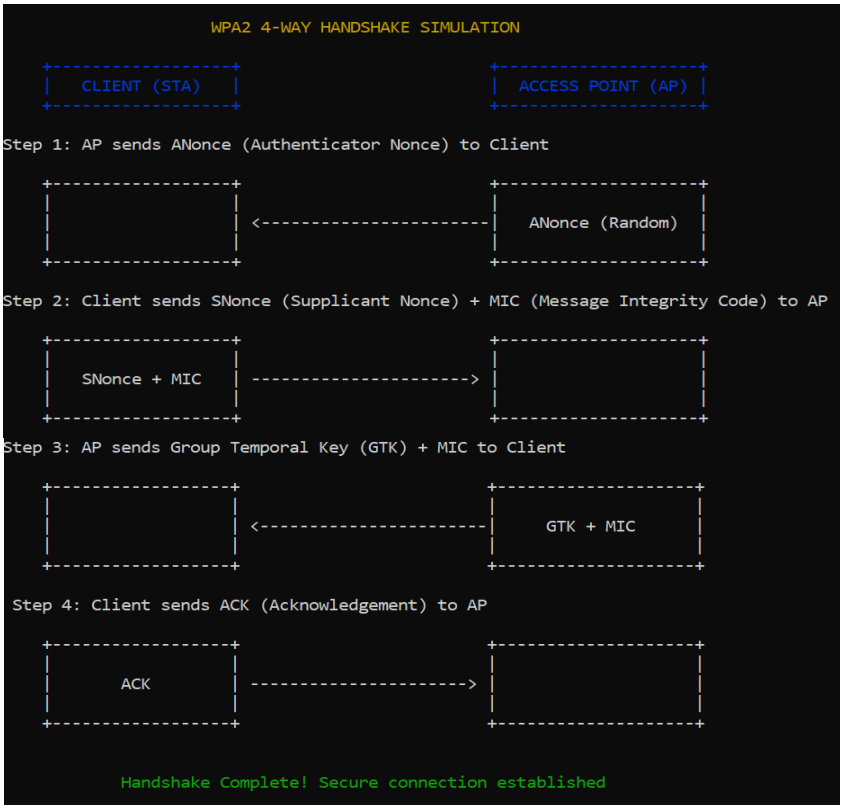


Figure 2: 4-way handshake

Simulated 4-way handshake (Allowing User Input)

```
> 1

[>] Enter SSID: sokojames
[>] Enter Password: 20231392
[>] Enter Client MAC Address (format XX:XX:XX:XX:XX:XX): 8C:04:BA:14:5B:A3
[>] Enter AP MAC Address (format XX:XX:XX:XX:XX:XX): CC:2F:71:D2:85:42

                         4-WAY HANDSHAKE SIMULATION

SSID                      sokojames
Client MAC                8C:04:BA:14:5B:A3
AP MAC                    CC:2F:71:D2:85:42
Step 1 AP sends ANonce    D76D89A67A177A53AA2F533898181A44
Step 2 Client sends SNonce EB37CF694A3A4CA8E65BCA877139D5BE
Derived PMK               PMK_20231392_sokojames
Derived PTK               PTK_PMK_20231392_sokojames_D76D89A67A177A53AA2F533898181A44_EB37CF694A3A4CA8E65BCA877139D5BE
AP derived PTK            PTK_PMK_20231392_sokojames_D76D89A67A177A53AA2F533898181A44_EB37CF694A3A4CA8E65BCA877139D5BE
Handshake Success         PTK Match!
```

Figure 3: Simulated 4-way handshake with user inputs

```
> 1

[>] Enter SSID: sokojames
[>] Enter Password: 20231392
[>] Enter Client MAC Address (format XX:XX:XX:XX:XX:XX): 8C:04:BA:14:5B:A3
[>] Enter AP MAC Address (format XX:XX:XX:XX:XX:XX): CC-2F-71-D2-85-42
[x] Invalid MAC address format.

O_O Thank you for using sokoFy.

[?] Do you want to return to the main menu? (y/n):
```

Figure 4: Validated format of a physical address

Network Adapters and Available Wi-Fi Networks

```
YOUR NETWORK ADAPTERS (MAC ADDRESSES)
Wireless LAN adapter Wi-Fi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-AC 8265
   Physical Address. . . . . . . . . : CC-2F-71-D2-85-3E
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

AVAILABLE WI-FI NETWORKS

1. KMU-WIFI (e4:7d:eb:a2:1a:e5) [Open]
2. TECNO POP 10C (b4:fb:e4:c7:a4:52) [Secured]
3. sokonalysis (c2:e7:41:fb:25:55) [Secured]
```

Figure 5: Scanned wireless networks

Simulated 4-way handshake

```
[>] Select the Wi-Fi network to connect to (number): 3
[>] Selected AP MAC: C2:E7:41:FB:25:55
[>] Enter your Client MAC address (format XX:XX:XX:XX:XX:XX): CC-2F-71-D2-85-3E
[>] Enter Wi-Fi key (8-63 chars): sokonalysiscryptotool

                                4-WAY HANDSHAKE
------------------------------------------------------------------------------------------

AP              Generated ANonce 5204f6abe0c598b6ede7f49c0000f1f8
Message 1       AP sends ANonce to Client
Client          Generated SNonce f337497e93b962458c52105788b58d09
Message 2       Client sends SNonce and MIC to AP
Derived PTK     Results 81546f2f2f04bedf2577316464632543c5ba5c7e2bdf545b405198aa9746afae
Message 3       AP sends install PTK message to Client
Message 4       Client confirms install
------------------------------------------------------------------------------------------
```

Figure 6: Simulated 4-way handshake with generated keys

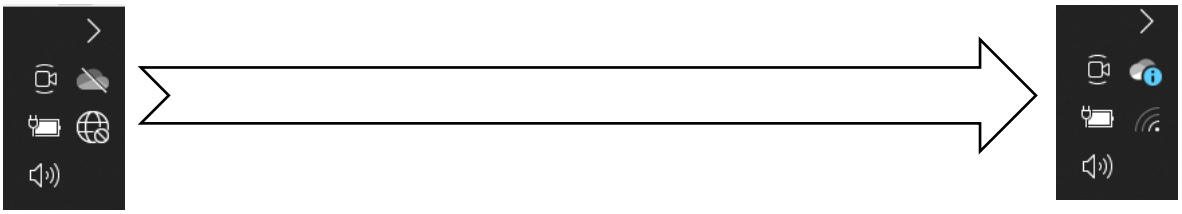
Actual 4-way handshake

1. New device with wrong password

Figure 7: Failed authentication

2. New device with correct password

   Notice that the device is not connected to the internet but after a successful 4-way handshake the device is connect, this applies to a secured network with a correct password and open Wi-Fi network.



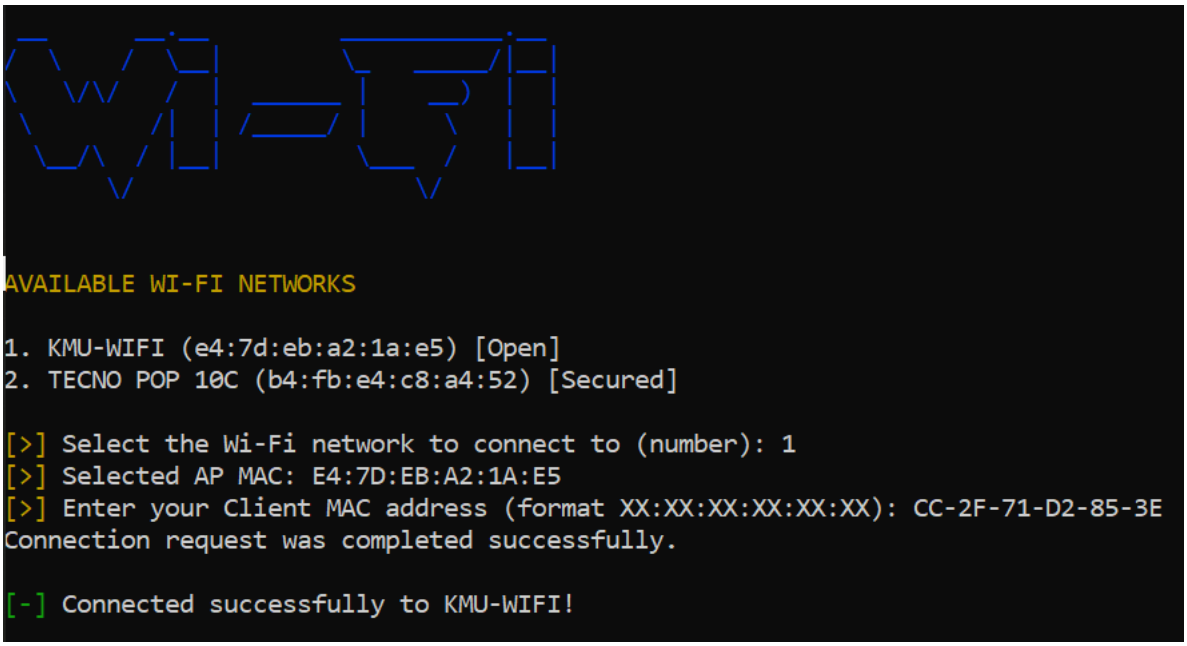


Figure 8: Successful authentication

NOTICE:

Was unable to provide the source codes in this documentation because it will consume a lot of pages, visit this GitHub repository for sokoFy to view the C (.h) and C++ (.cpp) files used to compile this program.

Conclusion

The 4-way handshake is a key part of securing Wi-Fi networks. Using the sokoFy simulator, we can better understand how the client and access point exchange keys to complete authentication. The simulation shows each step clearly and allows input validation for MAC addresses and credentials. This tool helps simplify a complex process and supports learning in wireless security.