Analysis of Multiple PSK in the context of WPA3

Prepared By **Lee Wright**

Senior Network Enginner

Redway Networks

Contents

Introduction

The ability to use multiple passphrases on a single Service Set Identifier (SSID) has been in place for more than a decade; since that time, this method of access has grown in popularity and has seen increasing adoption from different Wireless Local Area Network (WLAN) Vendors.

Wi-Fi Protected Access 2 (WPA2) underpins most of these existing solutions. However, weaknesses with WPA2, coupled with the inability of WPA2 to operate on the 6GHz frequency, necessitate alternative solutions.

This paper examines both standards-based implementations of WPA2 and proprietary implementations of Multiple Pre-Shared Key (PSK) solutions using WPA2. Afterwards, the newer successor to WPA2, Wi-Fi Protected Access 3 (WPA3), is examined, assessing why it is much more challenging to implement a multiple PSK solution using WPA3.

Lastly, this paper introduces an alternative solution to multiple PSK that provides security improvements over WPA2 and is compatible with 6GHz.

WPA2

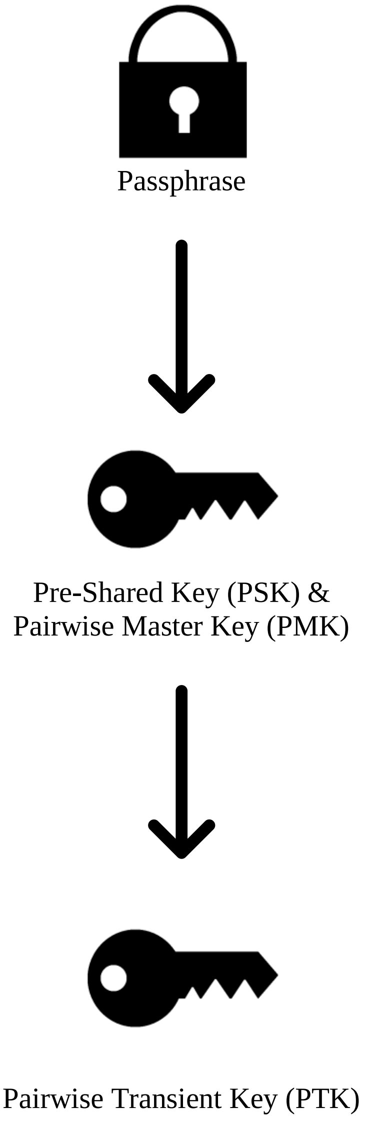
WPA2 – Standards-Based

WPA2 is a certification created by the Wi-Fi Alliance, which aligns with the Institute of Electrical and Electronics Engineers (IEEE) 802.11i standard, ratified in 2004. WPA2 has seen widescale adoption from both Wi-Fi infrastructure and Wi-Fi clients.

WPA2-Personal (also known as WPA2-PSK) is a method of securing an SSID with a single passphrase. From the user's perspective, they are prompted for a passphrase once they have selected the SSID. If the correct passphrase is input, the device reaches a connected state, and authentication is complete.

The passphrase goes through a specific process defined in the 802.11i standard to derive the keys used for encryption. Figure 1 explains the journey from a passphrase to a Pairwise Transient Key (PTK):

Figure 1



The PSK is derived from the passphrase using the following formula:

PSK = PBKDF2(PassPhrase, ssid, ssidLength, 4096, 256)

The same PSK will always be derived if the SSID and passphrase remain unchanged.

The PMK is obtained directly from the PSK.

The Four-way handshake derives the PTK.

This handshake uses two random numbers used only once, called the SNonce and ANonce, generated by the Supplicant (client) and Authenticator (infrastructure), respectively. These random numbers mean the PTK generated by every 4-way handshake is unique.

The formula used for deriving the PTK is:

PTK ← PRF-X(PMK, "Pairwise key expansion", Min(AA,SPA) || Max(AA,SPA) || Min(ANonce,SNonce) || Max(ANonce,SNonce))

The PTK is split into as many as five keys, used for encrypting traffic over the air. The exact details of this process are outside the scope of this paper and not relevant to its topic. The 802.11i protocol details more information on this process.

For this paper's topic of multiple PSK, the key takeaway is that the PSK is a predictable value that is always the same. Only the PTK is unpredictable, using the randomly generated ANonce and SNonce, making each PTK unique.

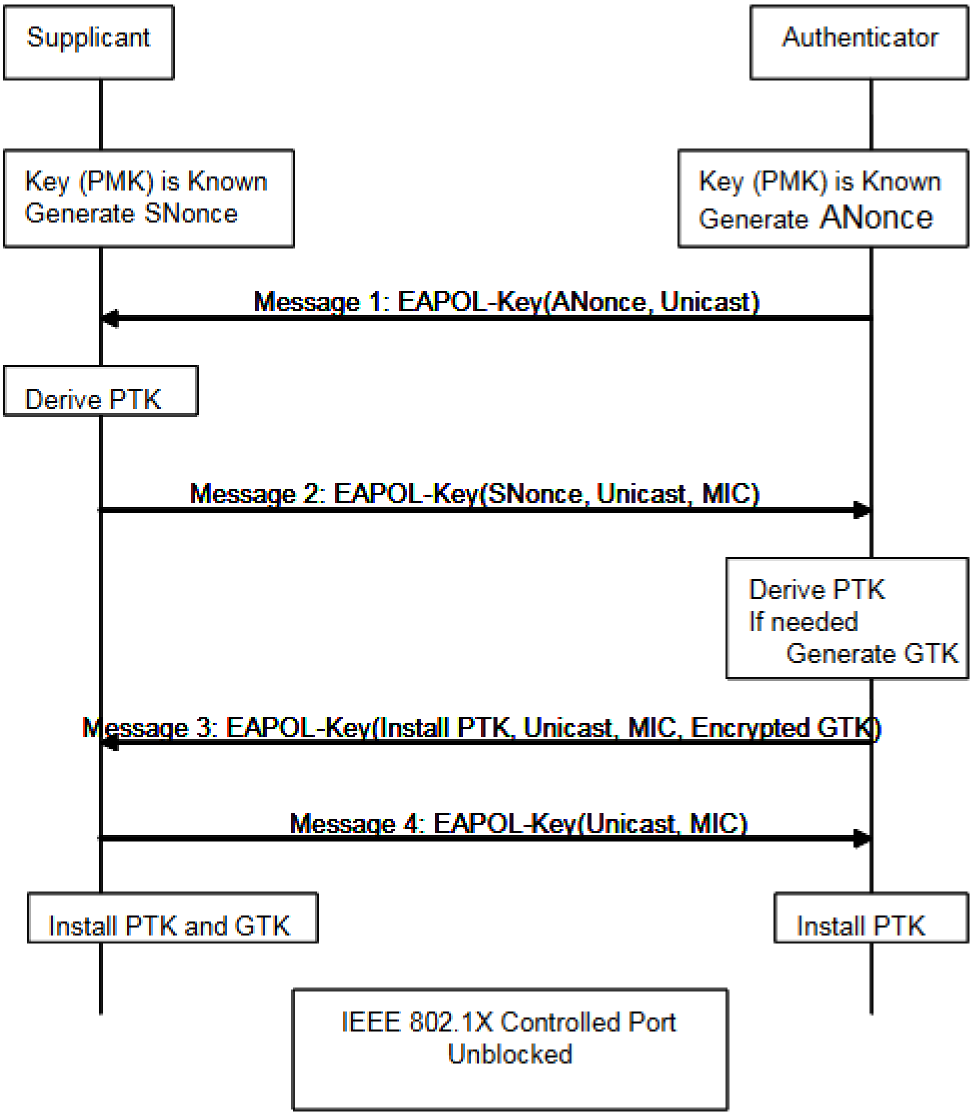
4-Way Handshake

The 4-Way handshake, which, as established above, is used to derive the PTK from the PMK.

WPA2, the standards-based WPA2 PSK implementation, and the proprietary WPA2 multiple PSK use the same 4-way handshake.

The below image details the frames exchanged in the 4-Way handshake:

Figure 2 (from IEEE 802.11i standard)


Supplicant
Authenticator
Key (PMK) is Known
Generate SNonce
Key (PMK) is Known
Generate ANonce
Message 1: EAPOL-Key(ANonce, Unicast)
Derive PTK
Message 2: EAPOL-Key(SNonce, Unicast, MIC)
Derive PTK
If needed
Generate GTK
Message 3: EAPOL-Key(Install PTK, Unicast, MIC, Encrypted GTK)
Message 4: EAPOL-Key(Unicast, MIC)
Install PTK and GTK
Install PTK
IEEE 802.1X Controlled Port
Unblocked

The first two messages from the 4-Way handshake are needed to derive the PTK.

The first message from the WLAN infrastructure (Authenticator) includes the ANonce, and the second from the client device (Supplicant) includes the SNonce and MIC.

The message integrity check (MIC) establishes whether the PSK derived on the supplicant and authenticator had the same value. If successful, the 4-way handshake can continue, and the PTK is installed, as per Figure 2.

WPA2 - Multiple PSK

The core concept of multiple PSK is allowing an SSID to support multiple passphrases.

One important distinction is that this is a proprietary technology and, as a result, is called many different names. Below are some examples from current vendors:

- Cisco (iPSK)
- Extreme (PPSK)
- Huawei (PPSK)
- Ruckus (DPSK)
- TIP OpenWiFi (M-PSK)
- Aruba (MPSK)
- Mist (Multi PSK)
- Cambium (ePSK)

These solutions differ, but all share the same core concept of allowing multiple passphrases on a single SSID.

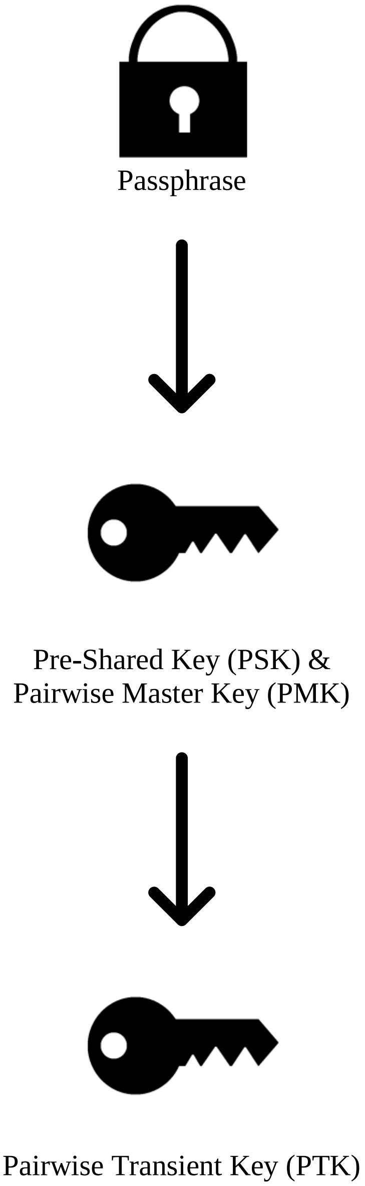
Unless specified otherwise, this document refers to multiple PSK as a general concept, not a specific vendor implementation.

Due to its proprietary nature, multiple PSK fuels innovation between competing vendors, and as a result, there are differentiating features when comparing these solutions. These variations include the number of PSKs supported per SSID, the ability to control several other network factors such as virtual local area network (VLAN) assignment, the number of connected devices per PSK, available bandwidth, traffic shaping and many more.

Another differentiator is where the PSKs are stored; some implementations of a multiple PSK solution store the PSKs on the access point or controller. Other implementations store the list of PSKs in an external server.

The diagram below explains the journey from a passphrase to a PTK, which we looked at in the WPA2 standards-based implementation but is now viewed from a WPA2 multiple PSK implementation perspective.

Figure 3



The PSK is derived from the passphrase using the following formula:

PSK = PBKDF2(PassPhrase, ssid, ssidLength, 4096, 256)

Each time a different passphrase is used, a different PSK is derived, hence the concept of Multiple PSKs.

The PMK is obtained directly from the PSK.

The Four-way handshake derives the PTK.

This handshake uses two random numbers used only once, called the SNonce and ANonce, generated by the Supplicant (client) and Authenticator (infrastructure), respectively. These random numbers mean the PTK generated by every 4-way handshake is unique.

The formula used for deriving the PTK is:

PTK ← PRF-X(PMK, “Pairwise key expansion”,
Min(AA,SPA) || Max(AA,SPA) ||
Min(ANonce,SNonce) || Max(ANonce,SNonce))

When comparing the standards-based WPA2 implementation to the multiple PSK proprietary implementation, the key differentiator is that multiple PSKs derive a different PSK for each unique password. The PMK will also be unique, as it is obtained directly from the PSK.

In addition to the features mentioned earlier, this implementation can improve security. If a single PSK is compromised, only the device(s) using that PSK are at greater risk of having their traffic decrypted. Also, the compromised PSK can be removed or changed without impacting all devices connected to the SSID; only those using the compromised PSK will be affected.

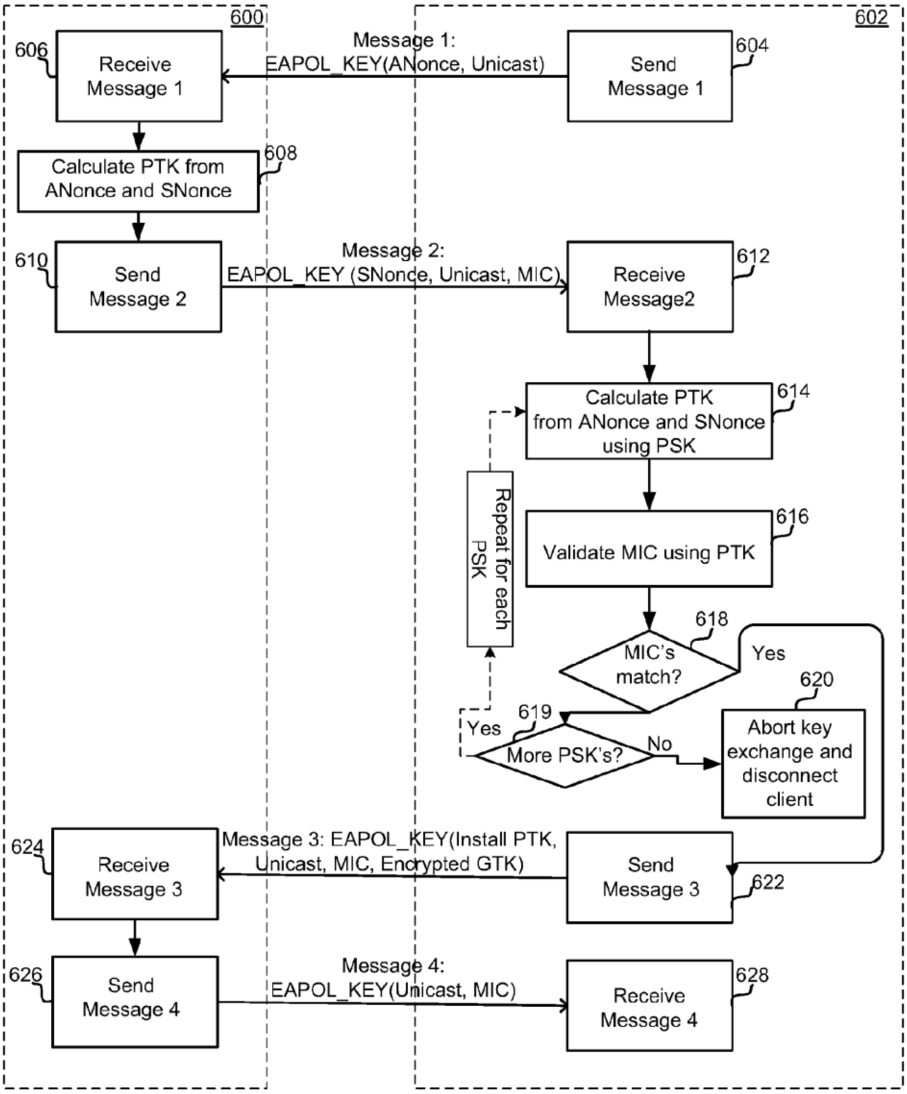
4-Way Handshake

Let us now examine the 4-way handshake, explicitly focusing on multiple PSK.

As previously mentioned, multiple PSK is not part of the IEEE 802.11 standard; this is a proprietary solution. As a result, each vendor implements their version of multiple PSK and is not required to follow a specific standard. We will examine two multiple PSK Patents; the first from Microsoft, filed in January 2009 and a second from Aerohive Networks, filed in June 2009.

Figure 4 (taken from the Microsoft Patent, “Support of multiple pre-shared keys in access point”) explains the implementation of this solution.

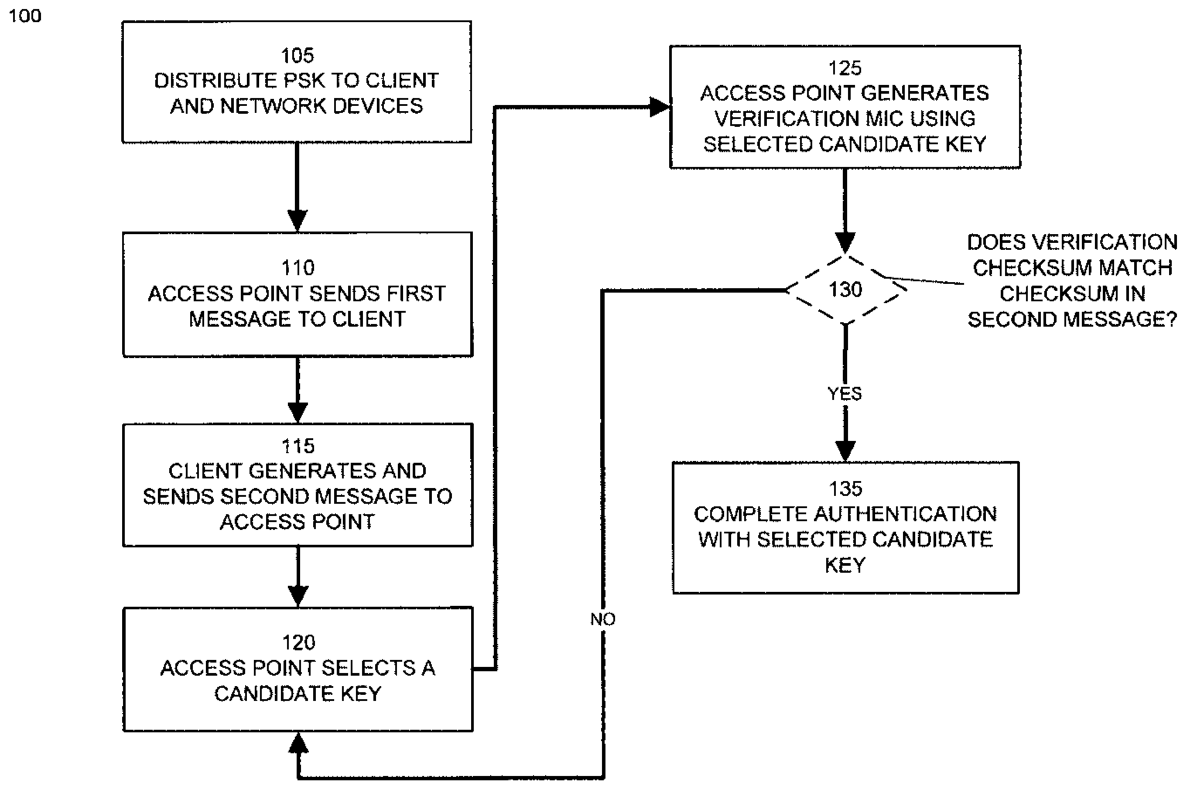
Figure 4 (from United States Patent US8898474B2, Microsoft)


600
602
606
Receive Message 1
Message 1: EAPOL_KEY(ANonce, Unicast)
Send Message 1
604
Calculate PTK from ANonce and SNonce
608
610
Send Message 2
Message 2: EAPOL_KEY (SNonce, Unicast, MIC)
Receive Message2
612
Calculate PTK from ANonce and SNonce using PSK
614
Repeat for each PSK
Validate MIC using PTK
616
618
MIC's match?
Yes
620
Abort key exchange and disconnect client
619
Yes
More PSK's?
No
624
Receive Message 3
Message 3: EAPOL_KEY(Install PTK, Unicast, MIC, Encrypted GTK)
Send Message 3
622
626
Send Message 4
Message 4: EAPOL_KEY(Unicast, MIC)
Receive Message 4
628

After the second message in the 4-way handshake, the authenticator (WLAN infrastructure) has all the information necessary to calculate the PTK. The authenticator enters a workflow to derive the PTK from each PSK stored in the authenticator, and each derived PTK attempts the MIC validation. If the first MIC fails, the next PTK is calculated from the next PSK, and the MIC validation is again attempted. This process continues until the MIC is validated. At this point, the 4-way handshake can continue, and the correct PTK can be installed. Alternatively, the 4-way handshake halts, and authentication fails if the MIC cannot be validated.

Another example, Figure 5, is the Patent filed by Aerohive Networks (owned by Extreme Networks at the time of writing). This workflow shows a similar process of trying each key until the verification checksum matches.

Figure 5 (from United States Patent US9674892B1, Extreme Networks)


100
105
DISTRIBUTE PSK TO CLIENT AND NETWORK DEVICES
110
ACCESS POINT SENDS FIRST MESSAGE TO CLIENT
115
CLIENT GENERATES AND SENDS SECOND MESSAGE TO ACCESS POINT
120
ACCESS POINT SELECTS A CANDIDATE KEY
125
ACCESS POINT GENERATES VERIFICATION MIC USING SELECTED CANDIDATE KEY
130
DOES VERIFICATION CHECKSUM MATCH CHECKSUM IN SECOND MESSAGE?
YES
NO
135
COMPLETE AUTHENTICATION WITH SELECTED CANDIDATE KEY

Both examples use each PSK to match the MIC the supplicant provided. It is this method of trying each PSK that makes multiple PSK possible.

<u>6GHz</u>

The Wi-Fi Alliance mandates WPA3 for all 6GHz connections. This requirement to use WPA3 on 6GHz connections means that any multiple PSK solution based on WPA2 will not be able to operate on the 6GHz frequency. As the adoption of 6GHz capable devices grows, the demand to move away from multiple PSK solutions based on WPA2 will also increase.

WPA3

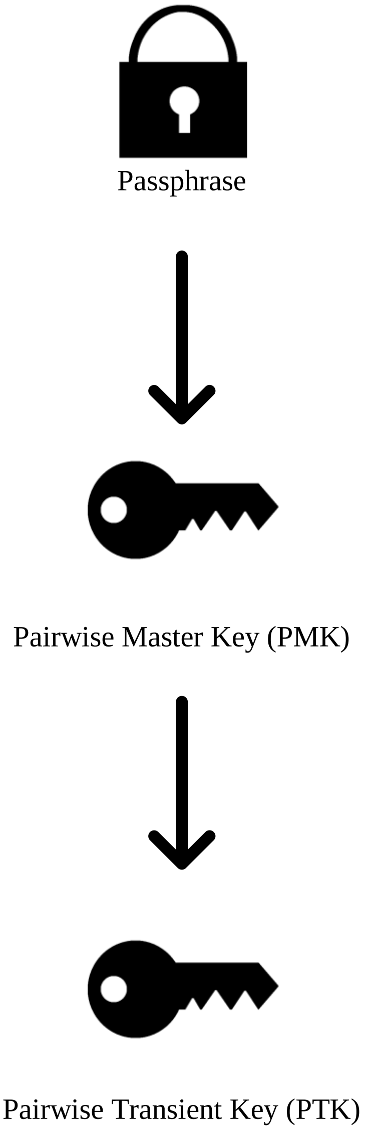
WPA3-Standards-based

The Wi-Fi alliance created WPA3 in response to the weaknesses found in WPA2. Compared to WPA2-Personal, WPA3-Personal offers a much more robust method for securing an SSID. Where WPA2-Personal relied on pre-shared keys, WPA3-Personal uses Simultaneous Authentication of Equals (SAE), often called WPA3-SAE.

From the user’s perspective, the process is identical to WPA2-Personal; they are prompted for a passphrase once they have selected the SSID. If the correct passphrase is input, the device reaches a connected state, and authentication is complete.

WPA3-Personal introduces a new handshake, the Dragonfly Handshake, used to derive the PMK. After the PMK is derived, the same 4-way handshake used with WPA2 derives the PTK. The diagram below explains the journey from a passphrase to PTK.

Figure 6



Simultaneous Authentication of Equals (SAE) uses a variant of the Dragonfly Key Exchange (RFC 7664) to generate the PMK.

The passphrase is used to derive a point on an elliptical curve; both the supplicant and authenticator each generate a unique private key (which is not shared) for use in the Diffie Hellman Key Exchange to derive the PMK.

Unlike WPA2-Personal, WPA3-Personal generates a unique PMK for each connection.

The Four-way handshake derives the PTK.

This handshake uses two random numbers used only once, called the SNonce and ANonce, generated by the Supplicant (client) and Authenticator (infrastructure), respectively. These random numbers mean the PTK generated by every 4-way handshake is unique.

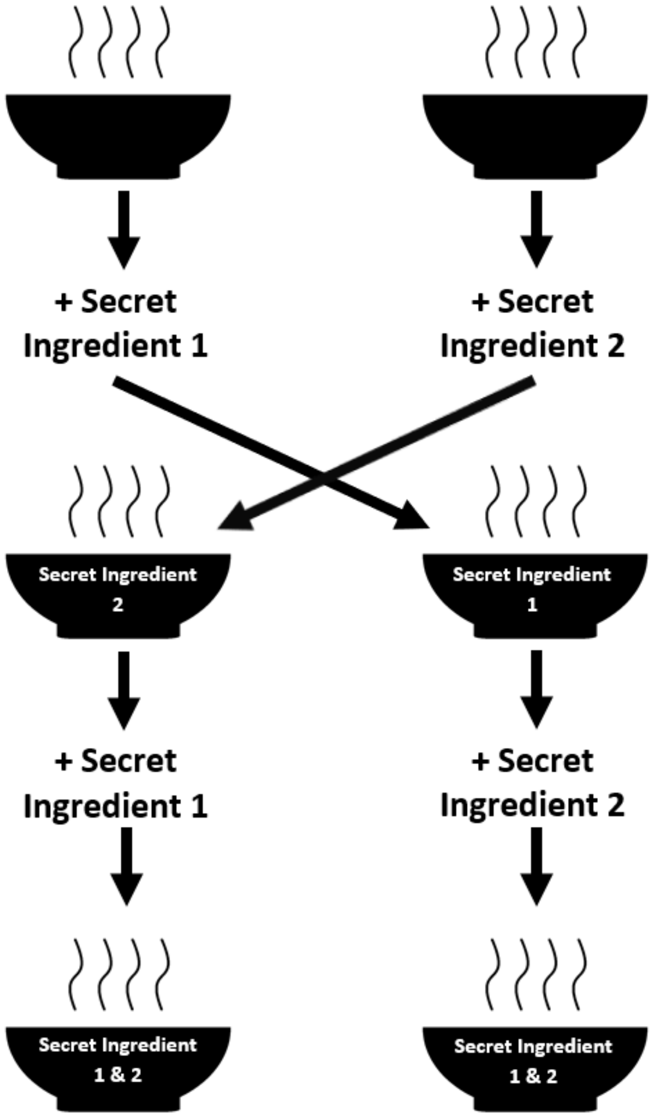
The formula used for deriving the PTK is:

PTK ← PRF-X(PMK, “Pairwise key expansion”,
Min(AA,SPA) || Max(AA,SPA) ||
Min(ANonce,SNonce) || Max(ANonce,SNonce))

Simultaneous Authentication of Equals

Simultaneous Authentication of Equals is a variant of the Dragonfly Key Exchange, which is based on the Diffie-Helman Key Exchange. Visualising the process to help understand the Diffie-Helman Key Exchange is helpful. Figure 7 details the process using a bowl of soup analogy.

Figure 7



The soup represents our passphrase indexed to a point on the elliptical curve.

A secret ingredient (private key) is added to the soup mixture (hashed).

The bowl of soup with the secret ingredient (hashed value) is exchanged.

The secret ingredient is added to the received bowl of soup (hashed with the other private key).

If the two bowls of soup (password) match in the first step, the new bowl of soup with both secret ingredients (PMK) will also match.

The original bowl of soup (password) and secret ingredient (private key) will never be exchanged over an insecure channel. The soup and secret ingredient are combined prior to being exchanged. It would be very challenging to identify the secret ingredient once it has been mixed into the soup, just as it is challenging to identify the private key and password from the hashed value.

WPA3 Multiple PSK problem

Both implementations from Microsoft and Extreme Networks examined earlier reused each password/PSK until the MIC passed on message 2 of the 4-way handshake.

Unfortunately, this is impossible with WPA3 because the password (after it is indexed to the elliptical curve and hashed with the authenticator's private key) is exchanged with the client at the beginning of the Dragonfly key exchange. Once exchanged, it is not possible to change this value.

In the soup analogy, the bowl of soup with a secret ingredient is sent to the other party; it cannot change the soup once it has been sent.

WPA3 Multiple PSK solutions

Despite the challenges that WPA3 brings to multiple PSKs, some solutions are available; at the time of writing, these solutions are available:

Cisco Networks does support WPA3-SAE with their iPSK solution by using an external RADIUS Server, which stores the Media Access Control (MAC) address of the connecting client and the corresponding password.

This solution works well for devices owned by the organisation, as building this database of MAC addresses and passphrases is possible. However, this solution is impractical for environments where the MAC address is not known in advance, such as BYOD and Guest devices.

Ruckus Networks, part of CommScope, offers another solution. The DPSK3 solution uses WPA2 to "Bound" the client to the DPSK Service, then WPA3 is used. DPSK3 requires the client device to first connect to 2.4 or 5GHz, and subsequent 6GHz connections are permitted.

This solution does not require MAC Addresses to be known in advance. However, 6GHz capable clients may experience issues connecting, as the client device controls the decision to connect to 2.4, 5 or 6GHz, not the WLAN infrastructure or the user. 6GHz-only clients are not compatible with DPSK3. In addition, using WPA2 for the initial authentication reduces the overall security of this solution.

Multiple PSK – Alternative solution

As an alternative solution to Multiple PSK, this paper suggests using WPA3-Personal or Opportunistic Wireless Encryption (OWE) for encryption, a captive portal for authentication and a control mechanism to apply rules and restrictions (such as VLAN assignment and bandwidth limits) to the client device.

By separating encryption, authentication, and the control mechanism, many benefits of a multiple PSK solution are realised without relying on WPA2. In addition, this also enables the use of the 6GHz frequency.

This paper coins Encrypt Authenticate Control (EAC) to refer to this alternative solution.

Encryption / Security

EAC is compatible with all Wi-Fi security protocols, but in the interest of security and practicality, only OWE and WPA3-Personal are recommended.

OWE provides more privacy assurances than WPA2-Personal due to the Enhanced Open authentication, which uses the Diffie-Helman Key Exchange to generate the PMK. In addition, OWE is compatible with 6GHz, unlike WPA2. OWE is, however, vulnerable to man-in-the-middle attacks.

WPA3-Personal is more resilient against man-in-the-middle attacks, but this implementation requires the user to enter a passphrase twice. Once during the WPA3-Personal authentication and once again at the captive portal.

Authentication

The captive portal handles authentication, prompting the user for a password. A database of passwords is utilised; each password should be mapped to a control policy for later use by the control mechanism.

The authentication is complete if the user supplies a valid password, and the control mechanism can proceed.

Administrators should be able to manually bypass the authentication and apply rules and restrictions for headless client devices that cannot use the captive portal.

Control mechanism

The control mechanism applies rules and restrictions to the client after successful authentication. The WLAN infrastructure should be automatically reconfigured to apply the desired rules and restrictions based on the password to control policy mapping.

The rules and restrictions should be comparable to those offered in a multiple PSK solution, such as VLAN assignment and bandwidth limitations.

EAC - Proof of concept code

This proof of concept code works with Cisco Meraki Wireless Access Points, but EAC as a concept is not limited to just one vendor.

Link to code: https://github.com/Wright-Fi/EAC-Proof-of-concept

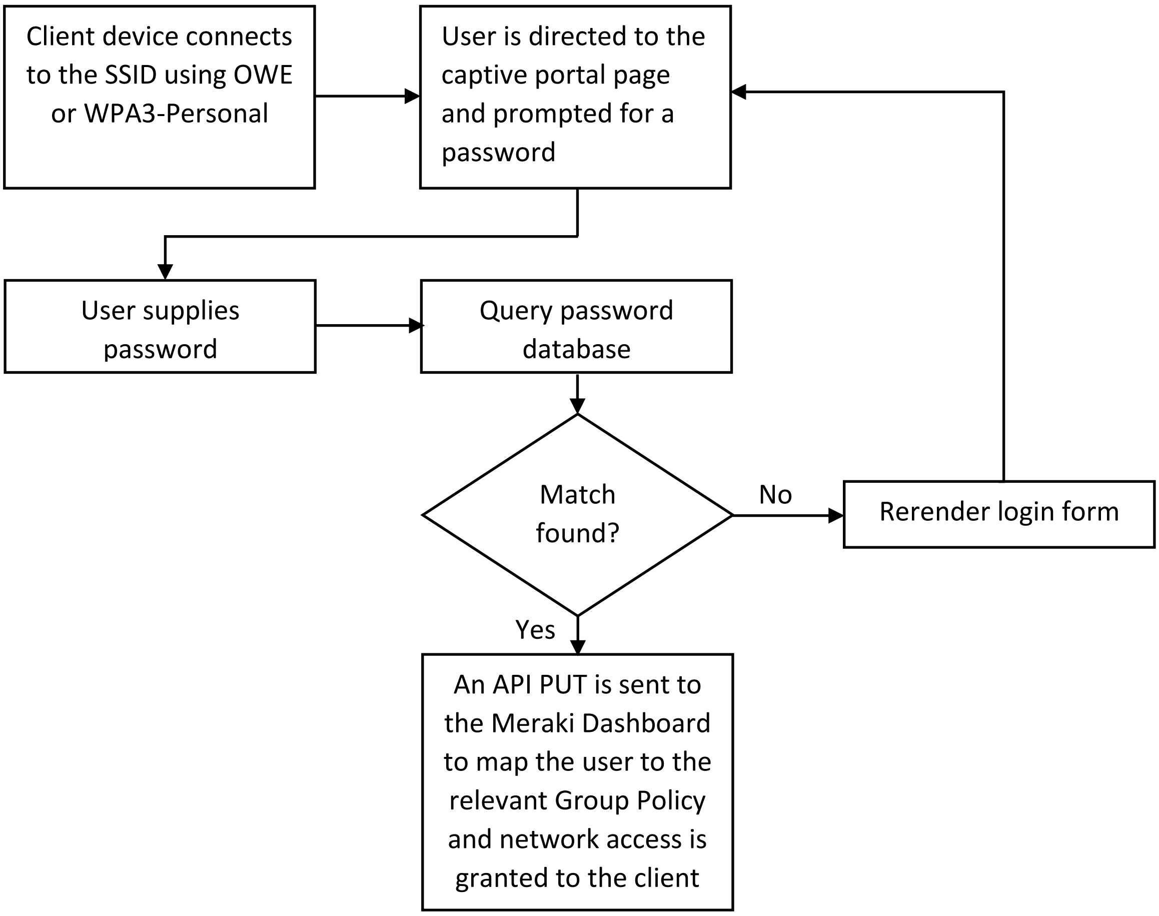
The Meraki Captive Portal API and the Meraki Dashboard API have been leveraged to build this proof of concept. The code for the captive portal is a forked version of the Meraki-splash-gp project adapted to meet the brief described above. Meraki Group Policies act as the control mechanism assigning rules and restrictions to the client device. Using Meraki Group Policies allows the same level of control as iPSK (the Meraki WPA2-Multiple Pre-Shared Key implementation). In addition, Group Policies can be manually applied to devices, allowing headless devices to be authenticated and bypass the captive portal.

Before connecting a client device, the SSID must be configured to direct users to the external Captive Portal, the Group Policies must be created, and the configs.js file must be updated with the relevant API Key, Network ID, Password and Group Policy IDs. Configuration instructions are provided in the project’s README file.

Initially, the client device connects to the SSID, using either WPA3-Personal or OWE; after installation of the PTKs, the wireless connection becomes encrypted. Next, an externally hosted captive portal is presented to the user, which prompts the user to enter a password. Once the password form is submitted, it queries the database for a matching password. If there is no matching password, the form is rerendered; if there is a matching password, an API PUT is sent to the Meraki Dashboard to map the connecting client to the Group Policy ID associated with the password used, and the authentication completes, granting the client access to the network.

Figure 8 shows the workflow used by the proof of concept code.

Figure 8


Client device connects to the SSID using OWE or WPA3-Personal
User is directed to the captive portal page and prompted for a password
User supplies password
Query password database
Match found?
No
Rerender login form
Yes
An API PUT is sent to the Meraki Dashboard to map the user to the relevant Group Policy and network access is granted to the client

Author's note: The proof-of-concept code is only suitable for testing purposes; it is not designed for a production environment, as passwords are stored and sent in cleartext.

EAC with Multiple VLAN assignments

When deploying EAC, the connecting device receives an Internet Protocol (IP) address before authenticating on the captive portal. If all passwords map to a single VLAN, there is no requirement for the IP address to change after authentication on the captive portal.

However, if multiple passwords map to different VLANs, a default onboarding VLAN is required to enable a connection to the captive portal. Once the user has connected and authenticated on the captive portal, the relevant VLAN can be updated; at this point, a new Dynamic Host Configuration Protocol (DHCP) Handshake should occur to update the IP address on the client device. EAC relies on the client device beginning a new DHCP handshake to obtain a new IP address after the VLAN change. The DHCP Handshake is outside the control of the WLAN infrastructure; this may cause problems for some client devices if they do not release their initial IP address after the VLAN change. However, this issue would only affect the initial connection, as subsequent connections would automatically connect to the VLAN assigned to the device, and no VLAN change would occur.

EAC with Man-in-the-middle attacks

An attacker could attempt a man-in-the-middle attack spoofing the captive portal to steal credentials. This attack would require either OWE to be in use, or if WPA3-Personal is in use, the WPA3 passphrase must be known by the attacker.

For this reason, a Wireless intrusion detection system (WIDS) should be deployed alongside any implementation of EAC to alert administrators of spoofed SSIDs. The deployment of a WIDS is outside the scope of this paper; the CWSP (Certified Wireless Security Professional) Study materials contain a significant amount of information for further reading on this topic.

Conclusion

The multiple PSK solutions based on WPA2 examined in this paper rely on the ability to predict the PMKs derived from known passphrases. The WLAN infrastructure can identify the passphrase used by matching the MIC.

Because of the unpredictable nature of the PMK in WPA3, implementing multiple PSK solutions based on WPA3 becomes much more challenging. A few solutions exist in the marketplace today but have limitations, restricting their deployment scope.

This paper introduces a new alternative solution to multiple PSK called EAC with a working proof-of-concept. This solution can mimic the features of a multiple PSK solution while also adding support for the latest security standards and placing no restrictions on the 6GHz frequency. In addition, there is no requirement for prior knowledge of MAC addresses. However, this solution also has its drawbacks. When multiple VLAN assignments are required, an onboarding VLAN is also required. A captive portal is used for authentication; headless devices can bypass this by manual intervention, but this adds additional administrative overhead, and this captive portal introduces a new attack vector.

As WPA3 and 6GHz gain adoption, so will the necessity for multiple PSK solutions not based on WPA2. Network administrators must carefully plan their use of Multiple PSK, weighing up the inability to use 6GHz and the security concerns of WPA2-based implementations against the constraints of WPA3-based implementations.

References

CWSP-206: Certified Wireless Security Professional: Study and Reference Guide – Lee Badman, Robert Bartz, Tom Carpenter, Brett Hill, Phil Morgan

https://standards.ieee.org/ieee/802.11i/3127/

https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-5/b_Identity_PSK_Feature_Deployment_Guide.html

https://extreme-networks.my.site.com/ExtrArticleDetail?an=000079435

https://www.ruckusnetworks.com/globalassets/digizuite/921377-dynamic-psk-pa-115991-en.pdf

https://openwifi.tip.build/device-feature-configuration-examples/device-feature-configuration-examples/multi-psk-mdu-multiple-shared-key

https://www.arubanetworks.com/techdocs/ArubaOS_86_Web_Help/Content/arubaos-solutions/mac-authentication/mpsk.htm

https://www.mist.com/documentation/multi-psk/

https://community.cambiumnetworks.com/t/epsk-multiple-pre-shared-keys/62609?page=2

https://www.cwnp.com/uploads/802-11i_key_management.pdf

https://patents.google.com/patent/US8898474B2/en

https://patents.google.com/patent/US9674892B1/en

https://datatracker.ietf.org/doc/html/rfc7664

https://en.wikipedia.org/wiki/Simultaneous_Authentication_of_Equals

https://simple.wikipedia.org/wiki/Diffie-Hellman_key_exchange

https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-13/config-guide/b_wl_17_13_cg/m_wpa3.html#Identity-iPSK

https://docs.cloud.ruckuswireless.com/ruckusone/userguide/GUID-45CA3127-0AC3-45D8-BD45-1E2CD65C84FD.html

https://youtu.be/d4C-Sf9PoBw

https://developer.cisco.com/meraki/captive-portal-api/overview/

https://developer.cisco.com/meraki/api-v1/

https://github.com/dexterlabora/meraki-splash-gp

https://documentation.meraki.com/General_Administration/Cross-Platform_Content/Creating_and_Applying_Group_Policies

https://www.wi-fi.org/system/files/Certification_Overview_v5.2.pdf