electronics

MDPI

Article

Stateless Re-Association in WPA3 Using Paired Token

Byoungcheon Lee

Department of Information Security, Joongbu University, 305 Dongheon-ro, Goyang-si 10279, Korea; sultan@joongbu.ac.kr

Abstract: In Wi-Fi Protected Access 3 (WPA3), a secure connection is established in two sequential stages. Firstly, in the authentication and association stage, a pairwise master key (PMK) is generated. Secondly, in the post-association stage, a pairwise transient key (PTK) is generated from PMK using the traditional 4-way handshake protocol. To reduce the heavy load of the first stage, PMK caching can be used. If the client and AP are previously authenticated and have a PMK cache, the first heavy stage can be skipped and the cached PMK can be used to directly execute the 4-way handshake. However, PMK caching is a very primitive technology to manage shared key between a client and AP and there are many limitations; AP has to manage a stateful cache for a large number of clients, cache lifetime is limited, etc. Paired token (PT)is a new secondary credential scheme that provides stateless pre-shared key (PSK) in a client-server environment. The server issues a paired token (public token and secret token) to an authenticated client where the public token has the role of signed identity and the secret token is a kind of shared secret. Once a client is equipped with PT, it can be used for many symmetric key-based cryptographic applications such as authentication, authorization, key establishment, etc. In this paper, we apply the PT approach to WPA3 and try to replace the PMK caching with the one-time authenticated key establishment using PT. At the end of a successful full handshake, AP securely issues PT to the client. Then, in subsequent re-association requests, the client and AP can compute the same one-time authenticated PMK using PT in a stateless way. Using this kind of stateless re-association technology, AP can provide a high performance Wi-Fi service to a larger number of clients.

Keywords: Wi-Fi security; WPA3; PMK caching; paired token; secondary credential; JSON web token; one-time authenticated key establishment; stateless re-association

Citation: Lee, B. Stateless Re-Association in WPA3 Using Paired Token. *Electronics* **2021**, *10*, 215. https://doi.org/10.3390/electronics10020215

Received: 13 December 2020
Accepted: 14 January 2021
Published: 19 January 2021

1. Introduction

Although WPA2 has been used for a long time to protect Wi-Fi communications, there have been many criticisms regarding the limitations of Wi-Fi Protected Access 2 (WPA2) [1]. The password in personal mode can be cracked offline. If the password is known to attackers, they can sniff or spoof other users. Since the management frame is not protected, attackers can try to disconnect other's connections easily. An open connection has no communication security.

WPA3 [2] released in 2018 provides several security improvements over WPA2. Open connection also provides communications security using the opportunistic wireless encryption (OWE) [3]. Th e password in personal mode is protected from an offline crack with the simultaneous authentication of equals (SAE) [4]. Using the device provisioning protocol (DPP) it provides easy connectivity to devices which do not have a display. It provides improved security using the 192-bit security suite.

In WPA3, a secure handshake is executed in two sequential stages. The first stage is the authentication and association stage which results to share a pairwise master key (PMK) between the client and AP. PMK is generated from OWE in open connection and from SAE in personal mode. In enterprise mode, a RADIUS server checks the authenticity of a client using various extensible authentication protocols (EAP) and then generates and distributes PMK to both client and AP through a secure communication channel.

The second stage is the post-association stage using the traditional 4-way handshake. It confirms the mutual authenticity of client and AP and generates the pairwise transient key (PTK) from PMK. The first stage is heavy in computation and communication, but the second stage of the 4-way handshake is reasonably efficient. In WPA3-Personal, SAE requires not only DH key exchange but also computation of password element (PE) from password which uses expensive hunting-and-pecking technique [5]. In WPA3-Enterprise extensible authentication protocol (EAP) with the remote RADIUS server takes some time. In WPA3-Open, an unauthenticated DH key exchange is required. If it should be repeated in every connection request, the consumption time will be very long both for client and for AP. Thus, reducing the latency of the first authentication and association stage is a very practical requirement for better performance. To reduce the latency of the full handshake, quick re-association technologies have been introduced.

PMK caching has been used as a fast roaming technology in an enterprise environment. If the client and AP have been previously authenticated and have a PMK cache, they can skip the first heavy stage and reuse the cached PMK to directly execute the 4-way handshake. If PMK caching is enabled, the client and AP keep the previous PMK and PMKID in the cache. In subsequent connection requests, the client can request re-association by presenting a valid PMKID, and then AP tries to find the corresponding PMK in the cache. If it is successful, heavy authentication of the first stage is skipped and the 4-way handshake is executed using the cached PMK. In WPA2-Personal PMK caching has no advantage in performance, since PMK is computed from a shared password with a simple hash computation. However, in WPA3-Personal, the PMK is computed using the expensive SAE so that PMK caching can enhance the performance a lot. However, PMK caching is a very primitive technology to manage shared secret between client and AP and there are many limitations; AP has to manage dynamic cache for large number of clients, connection process requires stateful service in AP, available cache lifetime will be limited, the number of clients will be limited, etc.

Paired token (PT) is a new secondary credential scheme that provides a stateless pre-shared key more efficiently in a client-server environment [6–9]. A server can manage a pre-shared key in a stateless way that it does not need to keep any client-specific information. Assume that there is an independent authentication system between client and server using some primary credential. The server authenticates the client using the primary credential and then issues a PT (public token and secret token) to the authenticated client as a secondary credential. Public token has the role of signed identity that represents the authenticated state of the client and its validity can be verified only by the server who has issued it. Secret token is a kind of shared secret between the client and server with a special property so that the server can compute a secret token any time from a given public token; thus, the server does not need to save client tokens issued by itself. This feature provides the stateless PSK property in the server side. PT can be applied to many symmetric key-based cryptographic applications such as authentication, authorization, secure communications, etc.

In this paper we apply the PT approach to WPA3 and try to replace the PMK caching with the one-time authenticated key establishment using PT. At the end of a successful full handshake, the AP securely issues a PT to the authenticated client and client saves it. It is used as a secondary credential of client during the lifetime of PT. In subsequent connection requests, the client can request re-association using PT. In this stage, the client and AP can compute the same one-time authenticated PMK from PT and use it to compute PTK in the 4-way handshake protocol. The proposed re-association protocol using PT has the following advantages.

1. The same PT can be used multiple times for re-association during the lifetime of PT.
2. Re-association request message by client provides one-time authentication of client and every re-association requests produce distinct one-time authenticated PMKs.
3. AP can compute a one-time authenticated PMK in a stateless way without using any client-specific saved information.

4. Once a client is equipped with PT, the re-association process is the same in heterogeneous authentication scenarios such as WPA3-Open, WPA3-Personal, and WPA3-Enterprise. Thus, a single AP can be configured to provide secure connection service for 3 different authentication scenarios.
5. The overall re-association process provides high performance in the AP side due to the stateless property.

This paper is organized as follows. Section 2 reviews WPA2, WPA3 and paired token. Section 3 presents the proposed stateless re-association scheme in WPA3. Section 4 provides security and performance analysis. Finally, Section 5 concludes the paper.

2. Related Works

The first Wi-Fi security protocol was the Wired Equivalent Privacy (WEP) in IEEE 802.11 standard released in 1997, but it has been proven to be easily broken. Wi-Fi Protected Access (WPA) was announced in 2003 by the Wi-Fi alliance to overcome flaws in WEP. In 2004 WPA2 was released as the IEEE 802.11i standard and it has been used for long time to protect wireless communications. Recently, in 2018 WPA3 was released by Wi-Fi Alliance with several security improvements over WPA2 [1,10]. Here, we will review WPA2 and WPA3.

2.1. WPA2

There are two modes of authentication in WPA2. WPA2-Personal, or referred to as WPA2-PSK (pre-shared key) mode, is designed for home or small office networks with single access point (AP). The client and AP authenticate each other using PSK, and prove the possession of PSK, without exchanging it. WPA2-Enterprise, referred to as WPA2-802.1X mode, is designed for enterprise networks with multiple APs. It requires a central Remote Authentication Dial-In User Service (RADIUS) server, and various kinds of extensible authentication protocols (EAP) can be used for authentication.

In WPA2-Personal mode, the client and AP share a static PSK. From PSK, a pairwise master key (PMK) is computed using the PBKDF2 key derivation function as follows,

$$PMK = PBKDF2(HmacSha1, PSK, SSID), \tag{1}$$

and then a 4-way handshake follows.

In WPA2-Enterprise mode, the client is authenticated by a central RADIUS server using various extensible authentication protocol (EAP) and then RADIUS server generates PMK and distributes it securely to client and AP. After that, a 4-way handshake follows between the client and AP.

The core component of WPA2 security is the 4-way handshake protocol. Using this protocol client and AP prove the possession of same PMK each other without exposing it over the communication channel, and then establish a fresh session key called pairwise transient key (PTK). The client and AP exchanges AP nonce (AN) and STA nonce (SN), and then PTK is computed from the attributes PMK, AN, SN, AM (AP MAC address), and SM (STA MAC address) as follows,

$$PTK = PRF(PMK, AN, SN, AM, SM). \tag{2}$$

The handshake also yields the group temporal key (GTK) which is used to decrypt multicast and broadcast traffic.

There have been many criticisms of the security of WPA2-PSK. The shared password (PSK) can be cracked offline; thus, using a strong password is highly recommended. Although a strong password is used, there are so many misuse cases in the real world in which a password is shared to public. For example, a Wi-Fi password is announce to the public in a cafe, restaurant, etc. If the password is known to attackers, they can sniff or spoof the communications of other users very easily. Management frames are not protected

such that attackers can disconnect other's connections with a de-authentication attack. In public Wi-Fi services using WPA2-Open, there is no communications security.

2.2. WPA3

WPA3, released in 2018 [2], provides several security improvements over WPA2. WPA2-Open connection uses plaintext communications. To provide communications security and user privacy even in open connection WPA3-Open provides individualized encryption using opportunistic wireless encryption (OWE, RFC 8110) [3]. In OWE protocol, the client and AP execute unauthenticated Diffie-Hellman key exchange to create one-time PMK and then a 4-way handshake follows to derive PTK from the PMK. Thus, WPA3-Open provides communications security.

In WPA2-Personal, PMK is computed from PSK using PBKDF2 function and it is static. Therefore, an offline dictionary attack on PSK was possible. To provide better protection of PSK WPA3-Personal uses simultaneous authentication of equals (SAE) [4,5] protocol in PSK-based authentication. In this protocol, the password element (PE) is computed from the password (PSK) using the hunting-and-pecking technique, and then Diffie-Hellman key exchange is executed using PE as a base element. The resulting PMK changes dynamically depending on Diffie-Hellman key exchange; thus, this handshake is resistant against offline dictionary attacks. The PMK is then used in a 4-way handshake to generate PTK.

Using a new device provisioning protocol (DPP), WPA3 provides easy connectivity of devices that do not have a display. It provides a simple and secure way to add these devices to an existing Wi-Fi network using QR codes. It provides concrete mutual authentication using public key cryptography and easy configuration of security in those devices.

WPA3 has improved security using 192-bit security suites. Using protected management frame client and AP exchange management frames in encrypted form can prevent an attacker's misbehavior.

2.3. PMK Caching for Fast Roaming

Full handshakes in WPA3 are heavy in computation and communications. OWE in WPA3-Open requires unauthenticated DH key exchange. In WPA3-Personal, SAE requires not only DH key exchange but also computation of password element (PE) from a password which uses expensive hunting-and-pecking technique [5]. In WPA3-Enterprise, an extensible authentication protocol (EAP) with remote RADIUS server takes some time. If it should be repeated in every connection request, it will be very time consuming both for the client and AP. To reduce the latency of full handshake, quick re-association technologies have been introduced.

PMK caching is a quick re-association technology that client and AP reuse the previously shared PMK in next connection requests. It has mainly been used in WPA2-Enterprise networks as a fast roaming technology, since full authentication with the central RADIUS server using EAP is heavy in performance, and sometimes takes several seconds. If the client and AP are previously authenticated and have PMK cache, a heavy full handshake can be skipped and the cached PMK can be reused to directly execute the 4-way handshake. If PMK caching is enabled, the client and AP keep the previous PMK and PMK_{ID} in the cache. PMK is computed as (1) and PMK_{ID} is a HMAC value computed from PMK as follows.

$$PMK_{ID} = H(PMK, PMK_{Name}||AM||SM) \quad (3)$$

PMK_{ID} is used as an index to identify PMK. If client requests re-association connection using PMK_{ID} and AP finds corresponding PMK in cache, the full authentication is skipped and the cached PMK is reused. Thus, the client and AP can immediately execute the 4-way handshake process, ensuring a minimal latency.

Opportunistic key caching (OKC) is an extended version of PMK caching in roaming scenario in a multiple AP enterprise environment. Once a client completes a full handshake with an AP, the PMK cache is synchronized automatically among all the APs in the same

network. Now if the client roams to any other AP in the same network, that AP would also have the PMK cache and the expensive EAP can be skipped, making the roam a lot faster.

If PMK caching is applied to WPA2-Personal, it has no performance gain since PMK can be computed easily from PSK with hash computation (1). Moreover, it is vulnerable to a brute-force offline attack called the PMKID attack [11]. Since PMKID is a static information computed from PSK using two Equations (1) and (3) and it is transported over the air, the attacker can launch an offline dictionary attack to match dictionary password and eavesdrop on PMKID. Thus, PMK caching is not recommended in WPA2-Personal mode.

On the other hand, if PMK caching is applied to WPA3-Personal, it will provide lots of performance gain. SAE in WPA3-Personal requires not only Diffie-Hellman key exchange but also computation of PE from the password. If PMK caching is used, the client and AP can skip this kind of heavy full handshake.

However, PMK caching is a very primitive technology to manage shared secret between client and AP and there are many limitations. AP has to manage a dynamically changing cache for large number of clients; thus, the number of clients will be limited. To provide re-association service AP has to find the PMK in cache corresponding to the presented PMK_{ID}, which requires a stateful service in AP. Because of the characteristics of cache memory, available cache lifetime will be limited.

If PMK caching is used extensively for long period of time, the same PMK is used multiple times in the 4-way handshake protocol to produce different PTKs. It is not recommended in the point of security.

2.4. Stateless Authenticated Key Establishment Using Paired Token

Paired token (PT) is a new secondary credential scheme that provides stateless pre-shared key (PSK) more efficiently in a client-server environment [6,7,9]. Assume that there is an independent authentication system between the client and server using some primary credentials. The server authenticates the client using a primary credential and then issues a paired token (public token and secret token) to the authenticated client as a secondary credential. The public token has the role of signed identity of the client that represents the authenticated state of the client. A secret token is a kind of shared secret between the client and server with a special property such that the server can compute secret token any time from a given public token; thus, the server does not need to save issued client tokens. Here, we describe the scheme in the following two stages.

2.4.1. Initial Authentication and Issuing Paired Token

Let us consider a simplified authentication model between a client and server. The client is registered to the server and has some primary credential for initial authentication. Assume that the server has a master secret key K which is used for issuing tokens. It is used only inside the server and never exposed outside.

In initial authentication, the client logs into the server using primary credential, for example, using ID and password. If initial authentication is successful, the server computes two tokens as follows.

1. Public token $T_p = G_{JWT}(K, Info)$: a normal JSON web token (JWT) on user's authorization information $Info$.
2. Secret token $T_s = G_{JWT}(K, T_p)$: a recursive JWT on the above public token T_p.

Here, $G_{JWT}(K, Info)$ is an abstract notation of issuing process of a JWT [9,12–14]. It represents that the server prepares user-specific authorization information $Info$ and puts it in the Payload, prepares proper Header, and generates a Signature, a HMAC value of the Header and Payload using the server's secret K,

$$Signature = HMAC(K, Header||Payload).$$

Then, $Token = [Header.Payload.Signature]$ is a valid JWT issued to the user by the server. $Info$ is a JSON object prepared by the server that server can decide which informa-

tion is included in $Info$ according to its policy. To issue JWT with limited lifetime, $Info$ can have information on the issuing time and expiration time. If T_p is used after its lifetime has passed, it will be invalidated. T_s is computed from T_p and it will be computed frequently in the server in later authentication stages. Therefore, no time information is included in the computation of T_s to make these repeated computations easy with no lifetime check. $< T_p, T_s >$ is a paired token that T_s is valid only if T_p is valid.

The server sends $< T_p, T_s >$ to a client through a secure communication channel. In the issuing stage of the paired token, the secure communication channel is required to send PT to the client securely. Note that the initial authentication requires a secure communication channel to send password securely and issuing paired token can use the same secure communication channel. As a secure communication channel, we can use https, or another custom secure channel. Client stores paired token securely in the application or key storage. In a web security environment, a paired token can be stored in browser storage such as local storage.

Public token T_p represents a signed identity of the user and can be sent to the server to provide identification of client. Note that its validity can be verified only by the server who has issued it, since the master secret key K is needed in verification. A secret token T_s is a kind of shared secret between client and server, and it will never be sent to the server directly. The server does not need to save $< T_p, T_s >$ in DB, since T_p will be presented by the client and T_s can be computed anytime from T_p. Therefore, T_s is an inherently shared secret with the server in a stateless way. Maybe the server can decide to store T_p for logging purposes, but it will not be used in the later authentication stage.

2.4.2. One-Time Authenticated Key Exchange Using Paired Token

If a client is equipped with PT as shown above, a single message quick one-time authenticated key exchange is possible using PT. Now, the client equipped with $< T_p, T_s >$ wants to establish a fresh session key with the server.

The client gets the current time t, computes a time-based one-time authentication value $auth$, and computes a one-time authenticated key k as follows.

$$auth = HMAC(T_s, t||T_p), \tag{4}$$

$$k = HMAC(T_s, t||T_p||\text{“}key\text{”}). \tag{5}$$

Here “key” is a pre-agreed label for key generation. The client sends $< T_p, t, auth >$ to the server.

Upon receiving $< T_p, t, auth >$, the server first verifies the validity of $auth$ as follows.

1. Verifies the validity of T_p and identifies who is requesting authentication.
2. Gets its own current time and checks that client's request time t is within an allowed limit (checking liveness of request to defend against replay attack).
3. Computes the secret token $T_s = G_{JWT}(K, T_p)$ from T_p and then verifies the validity

$$auth \stackrel{?}{=} HMAC(T_s, t||T_p). \tag{6}$$

If it is valid, the server computes the same one-time authenticated key k in Equation (5) using T_s. Here $auth$ is a time-based one-time authentication of the client and proves the possession of T_s. It is an application of time-based one-time password (TOTP) scheme [15] to a paired token scenario to prove the possession of T_s without exposing it. Thus, the same PT can be used multiple times for a one-time authenticated key exchange.

PT is a fully hash-based secondary credential scheme that its use in authentication protocol is very efficient. It is specially designed credential that can be used in 1-to-1 communication in client-server environment. It cannot be used in other communication channels with other servers.

3. Stateless Re-Association in WPA3 Using Paired Token

Since PT is a secondary credential scheme that provides stateless PSK in a client-server environment, it is a perfect solution to manage PMK in Wi-Fi connection. In this section we show how PT can be incorporated with WPA3 to enhance the re-association connection. We will replace the PMK caching-based re-association with PT-based re-association. If the PT-based re-association function is enabled, AP will issue PT to the authenticated client and client will use it for re-association in subsequent connections. In the following, we describe the PT-based re-association protocol.

3.1. Full Handshake and Issuing Paired Token

We consider 3 authentication scenarios; WPA3-Open, WPA3-Personal, and WPA3-Enterprise. In all 3 cases, the client and AP will share the same PMK and PTK after a successful full handshake. After that AP prepares client's authorization information $Info$ and computes the following paired tokens.

1. Public token $T_p = G_{JWT}(K, Info)$
2. Secret token $T_s = G_{JWT}(K, T_p)$

Then, AP encrypts $< T_p, T_s >$ using PMK (or PTK) and sends it to the client. Now, the client decrypts it to recover $< T_p, T_s >$ and save it in the client system. If AP wants to distinguish between 3 different authentication methods, AP can prepare $Info$ differently according to AP's policy. For example, in the case of WPA3-Enterprise client is explicitly authenticated by the RADIUS server that $Info$ can be prepared in a privacy preserving way. In the case of WPA3-Open AP can decide to include more client-specific information in $Info$ such that AP can distinguish the client in subsequent connections. If the privacy of a user is a prime issue, $Info$ can be prepared in anonymous manner. The lifetime of PT can also be decided according to AP's policy.

3.2. Quick Re-Association Using Paired Token

Now the client is equipped with $< T_p, T_s >$ and PT-based re-association function is enabled. If client wants to connect to the same AP again, the client gets current time t, computes a one-time authentication value $auth$ and computes a one-time authenticated PMK as follows.

$$auth = HMAC(T_s, t||T_p), \tag{7}$$

$$PMK = HMAC(T_s, t||T_p||"key"). \tag{8}$$

The client requests a re-association connection to the AP by sending $< T_p, t, auth >$. Upon receiving $< T_p, t, auth >$ AP checks the authenticity of client as follows.

1. Verifies the validity of T_p and identifies who is requesting re-association connection.
2. Gets its own current time and checks that client's request time t is within allowed limit.
3. Computes the secret token $T_s = G_{JWT}(K, T_p)$ from T_p and then verifies the validity

$$auth \stackrel{?}{=} HMAC(T_s, t||T_p). \tag{9}$$

If all verifications are valid, AP computes the same one-time authenticated PMK (8) using T_s. Note that AP computes PMK in a stateless way without using any client-specific stored information.

Now the client and AP have the same one-time authenticated PMK. The client and AP execute the 4-way handshake protocol to compute PTK from PMK. Note that $auth$ is a time-based one-time authentication and PMK is changing depending on t. So the same PT can be used multiple times for re-association for a longer period of time.

3.3. Forward Secure Re-Association Using Paired Token

The above quick re-association protocol is efficient, but does not provide forward security. If an attacker gets knowledge of T_s, then he can decrypt every previous encrypted

traffic using the same PT. Since PT is a secondary credential that is intended to be used multiple times during its lifetime, providing forward security is important.

To provide forward security, DH key exchange can be incorporated into the protocol. If the client wants to connect to the same AP again, the client prepares current time t and ephemeral DH key share g^x and computes

$$auth1 = HMAC(T_s, t||T_p||g^x). \tag{10}$$

Client sends $< T_p, t, g^x, auth1 >$ to AP.

Upon receiving $< T_p, t, g^x, auth1 >$, AP verifies the validity of $auth1$ in the following steps.

1. Verifies the validity of T_p and identifies who is requesting re-association connection.
2. Gets its own current time and checks that the time difference from client's request time t is within certain limit.
3. Computes the secret token $T_s = G_{JWT}(K, T_p)$ from T_p and then verifies the validity

$$auth1 \stackrel{?}{=} HMAC(T_s, t||T_p||g^x). \tag{11}$$

If the above verification is successful, AP prepare its ephemeral DH key share g^y and computes

$$auth2 = HMAC(T_s, t||T_p||g^{xy}), \tag{12}$$

$$PMK = HMAC(T_s, t||T_p||g^{xy}||\text{“}key\text{”}). \tag{13}$$

AP sends $< T_p, t, g^x, g^y, auth2 >$ to the client.

Then, the client can compute g^{xy} and verify the validity of $auth2$ in the same way. If it is valid, client computes the same PMK (13). Now the client and AP share the same PMK and can execute a 4-way handshake to derive PTK.

3.4. Fast Roaming in Enterprise Environment

Let us consider the fast roaming scenario in enterprise environment with multiple APs. If PMK caching is used for fast roaming, multiple APs have to share the dynamically changing PMK cache in real time, which is a quite heavy task. If PT-based re-association is used for fast roaming, it is enough for multiple APs to share the static master secret key K. If all APs share K, any AP can provide a fast roaming service by itself without any help of neighbor APs. In an enterprise environment, the RADIUS server and multiple APs are connected with a secret communication channel such that sharing K secretly is a quite practical assumption.

4. Analysis

4.1. Comparison of Features

We compare the features of PT-based re-association with the PMK caching-based re-association in Table 1.

In the case of PMK caching, the client and AP share PMK as a long-term secret and also use it as a session secret in a 4-way handshake. Thus, using a PMK multiple times for a long period of time should be very carefully done, although the real session key PTK changes depending on the exchanged randomness in the 4-way handshake. On the other hand, in PT-based re-association, the client and AP share T_s as a long-term secret and one-time authenticated PMK is used as a session secret in 4-way handshake. Since the session secret is changing in every request, the same T_s can be used safely for a long period of time.

In PMK caching, the client requests re-association to AP by presenting a valid PMKID. AP authenticates the client if it finds the corresponding PMK in the cache. If an eavesdropping attacker replays the request of the client, AP will accept the request and continue the re-association protocol. On the other hand, in PT-based re-association, the client requests

re-association to AP by presenting $< T_p, t, auth >$. Then, AP will verify the validity of $auth$ to determine the authenticity of request. If an eavesdropping attacker replays the request of the client at another time, the request will be invalidated and AP will not continue the re-association protocol. Thus, AP can defend against DOS attack.

Table 1. Comparison of features; PMK caching-based vs. PT-based re-associations.

	PMK Caching-Based	**PT-Based**
long-term secret	PMK	secret token
session secret	PMK	one-time PMK
request info.	PMKID	$< T_p, t, auth >$
authentication	possession	verify $auth$
identify client	cannot	$Info$ in T_p
service type	stateful	stateless
no. of clients	limited	unlimited
lifetime	cache limit	lifetime of PT
enterprise roaming	share cache	share K

Let us consider how AP identifies the client who is requesting re-association connection. In PMK caching, the client will send PMKID to AP. Although a corresponding PMK is found in the cache, AP cannot identify the client from PMKID and PMK, since this is randomized information. In PT-based re-association, the client sends T_p in the first move of request that AP can easily identify client from $Info$ and can even verify the authenticity of the client.

In the PMK cache, the AP's re-association service is stateful, since AP has to try to find the corresponding PMK in cache. It should be repeated in every request, regardless whether it is a legitimate request or forged attack. The stateful service requires lots of time and energy consumption depending on the number of clients. Thus, AP cannot provide a re-association service to a large number of clients. On the other hand, in PT-based re-association, the amount of time and computation required for each request is fixed and independent from the number of clients; thus, the number of clients is not limited.

In terms of possible lifetime of the re-association service, PMK caching depends on the lifetime of cache memory. If AP is rebooted and a PMK cache is deleted, a re-association connection is not possible. In PT-based re-association, the lifetime of the re-association service depends on the lifetime of PT, but note that it can be managed by AP according to its policy. Rebooting of AP does not affect the availability of the re-association service.

Let us consider how to provide a roaming service in an enterprise environment with multiple APs in the same network. In PMK caching, multiple APs have to share the dynamically-changing PMK cache synchronously in real time. In PT-based re-association, it is enough for multiple APs to share the master secret key K to be able to provide an efficient re-association service.

PT-based re-association has another advantage that the re-association process is consistent in heterogeneous authentication environments such as WPA3-Open, WPA3-Personal, and WPA3-Enterprise. Once client is equipped with PT regardless of how it has been issued, and the re-association process is the same in these authentication scenarios. Thus, a single AP can be configured to provide a secure re-association service for 3 different authentication scenarios.

4.2. Security Analysis

Unforgeability. Public token and secret token are JWTs signed by AP that they cannot be forged by other entities. Attackers can try to collect public tokens and re-association protocol messages, and then try to compute secret token or even AP's master secret key K. Attackers can also try to forge another re-association request message without having a secret token. The security of this kind of attack will depend on the security

of the underlying hash function. Note that JWT contains a HMAC value signed by AP. A successful forgery of JWT will be reduced to a successful forgery of the underlying HMAC without having a master secret key.

Resistance to replay attack. Any kind of eavesdropping and replaying attack will be difficult since time-based one-time authentication $auth$ is sent to AP in the first move of request, and AP will check its validity. If $auth$ is not valid, AP will not continue the re-association protocol. A simple replay attack will not work at another time. Attackers should be able to compute fresh protocol messages working at current time.

Resistance to DOS attack. Attackers can try to attack the availability of service by sending incorrect requests to AP. However, AP can detect this kind of attack very early in the first move of request. The client's request message contains time-based one-time authentication $auth$, and the verification process is very efficient with just a few hash computations. AP can detect and stop invalid re-association requests from attackers very early. The attackers will be requested to start from the full handshake again.

Resistance to MITM attack. Man-in-the-middle (MITM) attack is an issue related with the full authentication. The client has to be able to verify the authenticity of AP and AP has to issue PT only to the authenticated client. Once the client is equipped with PT correctly issued by AP, the client and AP have a special 1-to-1 secure communication channel. Although multiple clients share the same PSK in WPA3-Personal, they will have different PTs issued by the same AP. Note that PT is not related with PSK in any way. Any attacker in the middle cannot intrude into the secure communication channel established using PT between client and AP. In WPA3-Open, the client is not authenticated in any way, and there are possibilities of MITM attack. If there is a way that the client can verify the authenticity of AP, then a MITM attack can be prevented.

Secrecy of messages. In PT-based re-association protocol, the client sends $< T_p, t, auth >$ to AP in plaintext to start the re-association. If $auth$ is verified to be valid by AP, one-time authenticated PMK is computed and a 4-way handshake is executed to generate PTK. Afterward, all messages are encrypted with PTK. All communications except $< T_p, t, auth >$ are kept secret.

Forward security. Since PT is a secondary credential that is intended to be used multiple times during its lifetime, providing forward security is important. We have shown the forward secure version of re-association protocol, although it requires more communications and computation.

Privacy and untraceability. In the PT-based re-association public token is sent to AP in plain communication channel as an identification of the client; therefore, network attacker can identify the client from the communication traffic. If privacy is a prime issue, the AP can issue anonymous opaque PT with no client-specific information in public token. If the AP still wants to identify the client, AP can keep a record of the issued public token. This will depend on policy.

If fixed anonymous PT is used for long period of time, the network attacker can try to trace the activity of the same client. To provide untraceability, AP can issue renewed anonymous PT securely inside the already established secure channel. Issuing a renewed PT to an already authenticated client is not heavy in performance. A network attacker cannot correlate the renewed PT with the original PT, but AP can trace the identity of the client if renewal history is kept in AP.

System security. PT-based re-association uses time-dependent one-time authentication and one-time authenticated key establishment using PT. Therefore, any network attacker who does not have the knowledge of the secret token cannot generate fresh protocol messages and cannot continue the attack. Since the same PT is used multiple times during its lifetime, attackers will be more interested in system attacks that can get PT itself.

Since the secret token is a secondary credential that has to be stored and used in the client system, its security will highly depend on the system security, key storage security, and application security. If an attacker can get the secret token itself by hacking the client operating system or using some malicious software, then he will be able to sniff or spoof

the attacked legitimate users. Therefore, the client system has to be kept secure using the best practice in the point of system security. This is a common system security argument in which a credential is stored and used in the system itself.

4.3. Performance Analysis

We compare the performance of PT-based re-association with PMK caching-based re-association as shown in Table 2.

Table 2. Comparison of performance; PMK caching-based vs. PT-based re-associations.

	PMK Caching-Based	PT-Based
service type	stateful	stateless
no. of clients	limited	unlimited
enterprise roaming	share cache	share K
overall performance	moderate	high

PMK caching is a cache-based stateful service. To provide a re-association service, the AP has to keep PMKs and PMKIDs of a connected client in the cache. When a client requests re-association by presenting a PMKID, AP has to find the corresponding PMK from the cache such that it is a stateful service. It is hard for AP to provide a PMK caching-based re-association service to a large number of clients. To provide a fast roaming service in enterprise environment, multiple APs have to synchronize the dynamically changing PMK cache.

PT-based re-association is a stateless service that can be serviced to an unlimited number of clients. If a client sends a re-association request message $< T_p, t, auth >$, the client and AP share the same one-time authenticated PMK and can start the 4-way handshake immediately. All the computations required for the verification of one-time authentication and computing one-time authenticated PMK are 4 HMAC computations. This is a huge performance gain compared with the stateful service of PMK caching. In the point of fast roaming, multiple APs who share the master secret key K can provide a fast roaming service by themselves without any interaction with other APs.

If efficient PT-based re-association is used more extensively, we can expect a huge performance gain with reduced usage of a full handshake. The same AP device can provide a more efficient Wi-Fi connection service to a larger number of client devices.

4.4. Discussion on Weaknesses and Implementation Issues

In this section, we provide some discussion about the possible weaknesses and implementation issues of the proposed re-association scheme.

In the proposed PT-based re-association scheme, PT is a kind of secondary credential that is saved and used in the client system. A client will visit many different Wi-Fi networks, and each AP will issue a PT to a client for an efficient re-association service in future connections. Since PT is a credential that is used only for a specific 1-to-1 communication channel with the AP who had issued it, the overall number of PT saved in the client will be large. The whole life cycle of PT, issuing, storing, removing, and usage, will be managed automatically by the client system without requiring user awareness; thus, it provides easy security to users.

Client software should be implemented carefully such that these PTs are saved in secure place and used securely. Proper access control should be guaranteed such that PT can be accessed only when the client is connecting to the AP that had issued it. There is a possibility of attack using a malicious software that tries to steal PTs saved in the client system. Thus, its security will depend on the system security of the client. The client system should be managed using the best practices in terms of system security. This is a common security argument for computer systems that use saved credentials.

To provide fast roaming service in enterprise environment, we have considered the model that the RADIUS server shares the master secret key K with multiple APs, al-

though the RADIUS server and APs have a different role. The TOTP-like authentication model requires time synchronization between the client and AP. Modern operating systems and network time protocol (NTP) service can provide synchronized time service easily.

4.5. Further Research Directions

In the proposed PT-based re-association protocol, two authenticated key establishment protocols are used in a sequential manner. Firstly, a TOTP-like one-time authenticated key establishment protocol is used to produce one-time authenticated PMK from a shared secret T_s. Then, secondly, the traditional 4-way handshake protocol is used to produce PTK from the shared secret PMK. These two protocols have similar roles in the point of authentication and key establishment. It looks like that similar function is repeated twice.

Since the 4-way handshake is commonly used in three authentication scenarios such as WPA3-Open, WPA3-Personal, and WPA3-Enterprise, we decided to keep using it as a basic building block and focused on the PMK management. TOTP [15] is a well-known and widely used one-time authentication method in a client-server environment. Using a single shared secret client can prove its authenticity multiple times to the server without exposing the shared secret. It is a one-way authentication from client to server.

In the TOTP-like one-time authenticated key establishment protocol, *auth* is a non-interactive proof of knowledge of T_s in which time is used as a pre-agreed challenge. If *auth* is valid, PMK is computed in a pre-agreed manner. This is a very efficient one-way proof system from client to AP without requiring mutual interaction, but it is a deterministic proof.

On the other hand, the 4-way handshake protocol is a simple interactive proof of knowledge of PMK without exposing it. The client and AP exchange AP nonce (AN) and STA nonce (SN) as a randomized challenge, and then compute PTK (2) using these information. They prove the knowledge of PMK each other by exchanging Message Integrity Code (MIC). It is a randomized proof, but it requires two rounds of interactions.

Currently, the system is designed such that these two key establishment protocols are executed sequentially. We think there is a possibility of designing a more efficient re-association protocol by combining these two protocols.

5. Conclusions

In this paper, we proposed a new stateless re-association protocol using paired token which can be used in WPA3. It provides better performance than the traditional PMK caching-based re-association in the point of stateless service, a larger number of clients, and easy operation of fast roaming, etc. If the PT-based re-association is used more extensively in WPA3, we can expect huge performance gain with reduced usage of expensive full handshake and efficiency of the re-association process. The same AP device can provide a more efficient Wi-Fi connection service to larger number of client devices.

This paper is the first proposal of PT-based re-association in wireless security protocol. We need to invest more effort to analyze the security and examine the practicality of the proposed scheme in more detail. In order for this to be used in the real world, we also need to investigate every technological detail required in the wireless security protocol.

Funding: This research received no external funding.

Acknowledgments: This work was conducted while the author visited the University of Alabama at Birmingham as a sabbatical research fellow with the support of Joongbu University in 2020. Special thanks to Yulian Zheng for the favor of invitation and many helpful discussions on this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kohlios, C.P.; Hayajneh, T. A Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3. *Electronics* **2018**, *7*, 284. [CrossRef]

2. Wi-Fi Alliance. WPA3 Specification Version 3.0. 2018. Available online: https://www.wi-fi.org/download.php?file=/sites/default/files/private/WPA3_Specification_v3.0.pdf (accessed on 16 January 2021).
3. Harkins, D.; Kumari, W. RFC 8110-Opportunistic Wireless Encryption. Available online: https://tools.ietf.org/html/rfc8110 (accessed on 16 January 2021).
4. Harkins, D. Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks. In Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications (Sensorcomm 2008), Cap Esterel, France, 25–31 August 2008; pp. 839–844.
5. Harkins, D. Dragonfly Key Exchange. RFC 7664. 2015. Available online: https://tools.ietf.org/html/rfc7664 (accessed on 16 January 2021).
6. Lee, B. Strengthening of token authentication using time-based randomization. *J. Secur. Eng.* **2017**, *14*, 103–114. [CrossRef]
7. Lee, B. Stateless Randomized Token Authentication for Performance Improvement of OAuth 2.0 MAC Token Authentication. *J. Korea Inst. Inf. Secur. Cryptol.* **2018**, *28*, 1343–1454.
8. Lee, B. Efficient Wi-Fi Security Protocol Using Dual Tokens. *J. Korea Inst. Inf. Secur. Cryptol.* **2019**, *29*, 417–429.
9. Lee, B. Paired Token: A New Secondary Credential Providing Stateless Pre-Shared Key. *Int. J. Inf. Secur.* **2020**, submitted.
10. Benton, K. The Evolution of 802.11 Wireless Security. UNLV Informatics-Spring. 2010. Available online: https://benton.pub/research/benton_wireless.pdf (accessed on 16 January 2021).
11. Steube, J. New Attack on WPA/WPA2 Using PMKID. Available online: https://hashcat.net/forum/thread-7717.html (accessed on 10 December 2020).
12. Hardt, D. The OAuth 2.0 Authorization Framework. Available online: https://tools.ietf.org/html/rfc6749 (accessed on 16 January 2021).
13. Jones, M.B.; Hardt, D. The OAuth 2.0 Authorization Framework: Bearer Token Usage. Available online: https://tools.ietf.org/html/rfc6750 (accessed on 16 January 2021).
14. Jones, M.B.; Bradley, J.; Sakimura, N. JSON Web Token (JWT). RFC 7519. 2015. Available online: https://tools.ietf.org/html/rfc7519 (accessed on 16 January 2021).
15. M'Raihi, D.; Machani, S.; Pei, M.; Rydell, J. TOTP: Time-Based One-Time Password Algorithm. RFC 6238. 2011. Available online: https://tools.ietf.org/html/rfc6238 (accessed on 16 January 2021).