Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.2, Apr. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.2.417

이중토큰을 이용한 효율적인 Wi-Fi 보안 프로토콜*

이 병 천†‡
중부대학교 정보보호학과

Efficient Wi-Fi Security Protocol Using Dual Tokens*

Byoungcheon Lee†‡
Department of Information Security, Joongbu University

요 약

WPA2-PSK[1,2]는 클라이언트와 AP가 공유비밀키에 기반하여 4-way 핸드쉐이크 프로토콜을 이용하여 보안 세션을 설정하는 방식을 사용하고 있다. 이 방식은 도청공격 등 여러 가지 보안 문제점들이 제기되고 있으며 또한 효율성 측면에서도 클라이언트와 AP간에 여러번의 상호작용을 요구하여 보안세션 설정과정이 비효율적이다. WPA2의 보안 문제점을 해결하기 위하여 최근 WPA3[3] 표준이 제안되고 있지만 근본적으로는 동일한 4-way 핸드쉐이크 방법론을 사용하는 소규모 개선이다. OAuth 2.0[4,5,6] 토큰인증 기술은 한 번 인증이 확인된 클라이언트에게 서버가 토큰을 발급하여 인증상태를 오랜 기간 효율적으로 유지시킬 수 있는 인증유지 기술로 웹에서 널리 사용되고 있다. 이 논문에서는 OAuth 2.0 기술을 개선한 이중토큰을 이용하는 난수화토큰인증[11,12] 기술을 Wi-Fi 보안 프로토콜에 결합 적용함으로써, 초기인증과 보안세션설정을 구분하여 효율성을 개선한 새로운 Wi-Fi 보안 프로토콜을 제시한다. 즉, 한 번 초기인증에 성공한 클라이언트에게는 AP가 이중토큰을 발급하고 이후에는 이것을 이용하여 한 번의 평문 메시지 교환만으로 빠르게 보안세션을 생성할 수 있도록 성능을 개선하였다.

ABSTRACT

WPA2-PSK[1,2] uses a 4-way handshake protocol based on a shared secret to establish a secure session between a client and an AP. It has various security problems such as eavesdropping attacks and the secure session establishment process is inefficient because it requires multiple interactions between client and AP. The WPA3[3] standard has recently been proposed to solve the security problem of WPA2, but it is a small improvement using the same 4-way handshake methodology. OAuth 2.0 token authentication[4,5,6] is widely used on the web, which can be used to keep an authenticated state of a client for a long time by using tokens issued to an authenticated client. In this paper, we apply the dual-token based randomized token authentication[11,12] technology to the Wi-Fi security protocol to achieve an efficient Wi-Fi security protocol by dividing initial authentication and secure session establishment. Once a client is authenticated and equipped with dual tokens issued by AP, it can establish secure session using them quickly with one message exchange over a non-secure channel.

Keywords: Wi-Fi security, WPA2, WPA3, randomized token authentication, dual tokens

Received(12. 20. 2018), Modified(03. 15. 2019), Accepted(04. 04. 2019)
* 이 논문은 2019년도 중부대학교 학술연구비 지원에 의하여 이루어진 것임

† 주저자, sultan@joongbu.ac.kr
‡ 교신저자, sultan@joongbu.ac.kr(Corresponding author)

I. 서 론

Wi-Fi는 전파를 이용하여 인터넷에 연결할 수 있도록 하는 무선랜 기술로 우리의 모바일 생활 발전에 큰 역할을 하고 있다. 무선랜에서는 전파를 통해 정보가 전송되므로 공격자에 의한 도청공격, 변조공격을 방지하는 것이 첫 번째 보안요구사항으로 요구되며, 이를 제공하기 위하여 통신패킷을 암호화하여 전달하는 보안통신 프로토콜을 사용해야 하는데 이 기술은 WEP, WPA, WPA2, WPA3[1,2,3] 등으로 발전해왔다. 초창기에 사용되었던 WEP는 RC4 스트림 암호의 공유비밀키의 길이가 40비트 또는 104비트로 짧게 설계되어 있으며 24비트의 IV값을 사용하는 방식이 취약하게 설계되어 공격자가 통신패킷을 수집하면 공유키를 쉽게 찾아낼 수 있다는 취약점이 있다[13]. WPA2[2]는 현재 널리 사용되고 있는 개선된 Wi-Fi 보안 프로토콜로 두가지 환경에서 사용되고 있다. 먼저 WPA2-PSK는 AP(access point)와 클라이언트가 공유비밀키를 사용하여 보안세션을 설정하여 사용하는 방법으로 이 방법은 가정이나 카페 등의 소규모 제한된 영역에서 하나의 AP를 이용하여 소규모 무선랜을 제공하기 위해 사용된다. WPA2-Enterprise는 다수의 AP를 운영해야 하는 큰 규모의 조직에서 인증의 효율성을 위해 사용자 인증을 RADIUS 등의 프로토콜로 운영되는 중앙집중식 인증서버에 의존하게 하는 방법이다.

WPA2에서는 AP와 클라이언트가 공유비밀키를 기반으로 보안세션을 설정하기 위해서 4-way 핸드쉐이크 프로토콜을 사용한다. 이 방법은 AP와 클라이언트가 공유비밀키를 통신상으로 직접 주고받지는 않으면서도 인증과 함께 안전한 보안세션을 생성하기 위해 고안된 방법으로 서로 랜덤하게 생성된 ANonce와 SNonce를 주고받은 후 이들 정보와 공유비밀키를 함께 이용하여 세션비밀키를 생성한다. 그런데 이 방법은 공유비밀키가 공격자에게는 알려지지 않는다는 가정하에서 설계된 것인데 실제로는 카페 등 많은 공공장소에서 무선랜의 비밀키를 공지하는 등의 방법으로 공개 무선랜 형태로 운영하고 있어서 공격자가 공유비밀키를 알고 있는 환경이 된다. 이런 환경에서는 공격자가 무선 네트워크 도청을 통해 타인의 보안세션 생성시 주고받는 ANonce, SNonce 값들을 수집하면 세션비밀키를 똑같이 계산해낼 수 있고 타인의 통신을 쉽게 도청할 수 있게 된다.

이러한 취약점을 개선하기 위해서 WPA3[3] 표준에서는 simultaneous authentication of equals(SAE)[8] 라는 방식으로 모듈러 승산 연산을 이용하는 방식으로 변형된 4-way 핸드쉐이크를 수행하는데 공격자가 공유비밀키를 알고 있고 통신을 도청하더라도 세션비밀키를 계산할 수 없도록 프로토콜이 개선되었다. 또한 opportunistic wireless encryption(OWE)[9] 방식을 이용하면 고정된 공유비밀키를 사용하는 대신 AP와 클라이언트가 Diffie-Hellman 프로토콜을 이용해 공유비밀키를 새로 생성하여 4-way 핸드쉐이크를 수행하도록 함으로써 공격자는 새로 만들어지는 공유비밀키를 알 수 없기 때문에 세션비밀키를 계산할 수 없도록 개선되었다. WPA3는 모듈러 승산 등 계산량이 많은 공개키 암호 방식을 추가 적용하여 보안성을 개선하였지만 여전히 동일한 4-way 핸드쉐이크 방식에 의존하는 소규모 개선이라고 볼 수 있다.

Wi-Fi 보안 프로토콜에서 또 하나의 근본적인 취약점은 클라이언트 입장에서는 접속하고자 하는 AP에 대한 신뢰성을 확인하기 어렵다는 것이다. 공격자가 도청 기능을 가지는 변조된 AP를 제작하여 운영한다면 이 AP에 접속하는 타인의 통신을 쉽게 도청할 수 있을 것이다. 공격자는 또한 프록시 AP를 운영하면서 남의 Wi-Fi 통신이 자신을 거쳐가도록 하여 도청할 수 있는데 AP에 대한 인증기능이 없으므로 이러한 도청공격을 방지하기 어렵다. 인증서를 이용하여 웹서비스 보안을 제공하는 HTTPS[7]의 사례에서와 같이 AP도 인증서를 이용하여 인증된 제품임을 확인할 수 있도록 개선할 필요가 있다.

Wi-Fi 보안세션 생성의 효율성을 고려해보자. 4-way 핸드쉐이크 방식은 클라이언트-서버 환경에서 서버에 로그인하기 위하여 미리 등록된 패스워드를 입력하는 전통적인 지식기반 로그인 방법을 사용하는 것과 비슷하다고 볼 수 있다. 네트워크를 도청하는 공격자에게 패스워드를 노출시키지 않기 위하여 패스워드를 직접 전송하지는 않으면서 인증을 하고, 동시에 이후의 통신보안을 위한 세션비밀키를 상호 공유하기 위하여 4번의 통신을 요구하는 복잡한 프로토콜을 사용하게 되는데 우리는 이것의 효율성을 향상시키는 것을 목표로 한다.

한편 웹서비스에서는 사용자 인증의 효율성과 사용자 편의성을 향상시키기 위하여 토큰인증[4,5,6]을 적극 활용하고 있다. 즉 초기인증시 한 번 로그인에 성공한 사용자에게는 서버가 인증된 사용자라는

의미를 가지는 토큰을 발급하고 사용자는 브라우저에 토큰을 저장하고 이것을 오랜기간 자신의 인증상태를 유지하는 목적으로 사용한다. 토큰은 서버의 서명이 포함되어 있어 공격자가 위조할 수 없다. JWT[5] 방식으로 사용하는 OAuth 2.0 bearer 토큰[4] 방식은 동일한 토큰을 오랜 기간 서버에 반복 전송하는 방법으로 사용되는데 이것은 도청공격에 취약하므로 초기인증 및 토큰의 발급, 토큰의 사용 등 인증유지의 전 과정이 HTTPS[7] 보안통신 환경에서 사용되어야 한다는 제약이 있다. OAuth 2.0 MAC 토큰[6] 방식은 클라이언트와 서버가 공유비밀키를 통해 MAC 인증을 수행함으로써 보안통신을 사용하지 않아도 된다는 장점이 있지만 서버는 인증을 위해 사용자와의 공유비밀키를 관리해야 하는 상태형 인증(stateful authentication)으로서 효율성을 중시하는 웹 환경에서의 인증유지에는 사용하기 어렵다는 단점이 있다.

이중토큰을 이용하는 난수화 토큰인증[11,12] 기술은 전통적인 토큰인증에서의 이러한 문제점을 해결하기 위한 방법으로 제안된 것으로 서버는 초기인증된 사용자에게 ID 역할을 하는 공개토큰과 패스워드 역할을 하는 비밀토큰의 두 개의 서명된 토큰을 발급하고 사용자는 이들 토큰을 브라우저에 저장하여 사용한다. 인증유지시에는 비밀토큰을 이용하여 현재시간에 대한 일회용 인증값을 계산하여 공개토큰, 현재시간과 함께 서버에 전달함으로써 인증된 상태임을 증명할 수 있으며, 서버는 사용자가 전달한 정보를 이용하여 공개토큰으로부터 비밀토큰을 계산해내고 즉시 인증을 검증할 수 있다. 이런 인증유지 과정은 평문통신 채널을 통해서도 안전하게 수행될 수 있으며 서버가 사용자별 비밀정보를 관리할 필요가 없어서 무상태 인증을 제공할 수 있다는 장점이 있다.

우리는 Wi-Fi 통신보안 프로토콜에 이러한 난수화 토큰인증 기술을 결합 적용함으로써 한 번 초기인증된 이후에는 오랜 기간 빠르게 보안세션을 설정할 수 있도록 효율성을 개선한 이중토큰 기반 Wi-Fi 보안 프로토콜을 제안한다. 아울러 AP의 신뢰성 제공을 위해 인증서를 사용하도록 설계하였다. Wi-Fi의 초기 접속시 사용자 인증에 성공하면 AP는 보안세션 설정에 필요한 공개토큰과 비밀토큰을 클라이언트에 발급한다. 클라이언트는 이후 이들 토큰을 이용하여 보안세션 설정을 빠르게 수행할 수 있으며 이러한 보안세션 설정 과정은 평문통신 환경에서도 안전하고 효율적으로 수행 가능하다는 장점이 있다.

이 논문은 다음과 같은 내용으로 구성되어 있다. 2장에서는 제안방식의 이해를 위해 필요한 기존의 관련 기술들을 설명한다. 3장에서는 이중토큰 기반의 효율적인 Wi-Fi 보안 프로토콜을 제시한다. 4장에서는 기존 프로토콜들과의 보안성 및 효율성을 비교 분석하고 5장에서 결론을 맺는다.

II. 관련 연구

2.1 WPA2 프로토콜

WPA2-PSK[2] 프로토콜의 핵심 방법론은 AP가 공유비밀키(pre-shared key, PSK)를 이용하여 클라이언트를 인증하고 동시에 이것으로부터 보안통신을 위한 세션비밀키를 생성하는 것이다. 공유비밀키를 무선네트워크로 직접 전송하는 것은 도청의 위험이 있으므로 공유비밀키를 전송하지 않고도 인증과 함께 세션비밀키를 생성하기 위한 방법으로 4-way 핸드쉐이크 프로토콜을 사용한다.

Fig. 1은 WPA2에서 사용하는 4-way 핸드쉐이크 프로토콜을 나타낸다[1]. AP와 클라이언트는 공유비밀키인 pairwise master key(PMK)를 가지고 있는데 랜덤하게 생성한 ANonce와 SNonce를 평문으로 주고받은 후 패스워드기반 키생성함수인 PRF를 통해

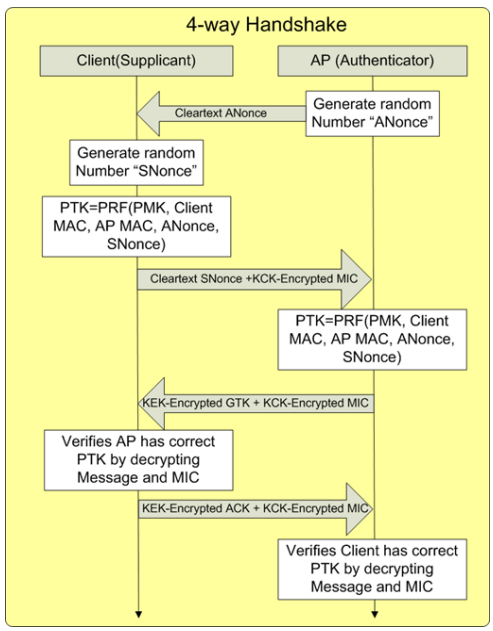

Fig. 1. 4-way handshake protocol

세션비밀키인 pairwise transient key(PTK)를 계산하게 된다. AP와 클라이언트는 PTK로부터 key confirmation key(KCK), key encryption key(KEK), temporal key(TK)등을 유도하여 사용한다. 만일 공격자가 PMK를 이미 알고 있는 환경이라면 무선랜의 세션 초기설정 패킷들을 도청하고 ANonce와 SNonce를 수집하여 이들로부터 타인의 PTK를 계산할 수 있게 되는 것이다. 실제로 카페 등 많은 공공장소에서 무선랜의 비밀키를 공지하는 등의 방법으로 공개 무선랜으로 운영하고 있어서 공격자가 공유비밀키를 알고 있는 환경이므로 이러한 공격의 위험이 크다. 이러한 도청은 수동적인 공격으로서 사용자가 알아채기 어렵기 때문에 더욱 큰 피해를 주게 된다.

Fig. 2는 WPA2-Enterprise에서의 RADIUS 인증서버를 이용하는 사용자 인증 과정을 나타내는데 사용자는 미리 등록된 ID/Password를 입력하게 되고 이것이 인증서버로 전달되어 인증을 받게 된다. 사용자가 인증되면 인증서버는 PMK를 생성하여 클라이언트와 AP에게 전달하고 클라이언트와 AP는 이를 기반으로 동일한 4-way 핸드쉐이크를 통해 세션비밀키를 생성한다. 클라이언트와 인증서버 사이의 통신을 보호하기 위하여 인증서버는 자신의 인증서를 클라이언트에게 배포하고 ID/Password 정보를 암호화 전송하도록 하며 인증 후 클라이언트에게 발급하는 PMK를 마찬가지로 암호화 전송하게 된다.

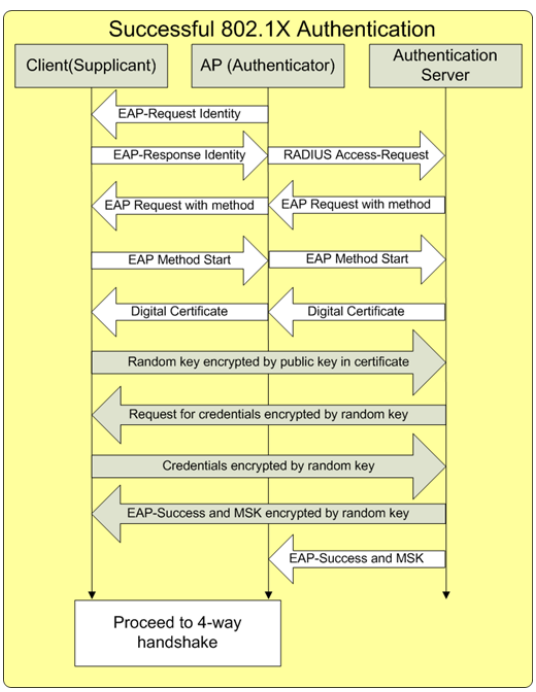

Fig. 2. Authentication in WPA2- Enterprise

그러나 현재의 WPA2-PSK 프로토콜에서는 클라이언트가 AP를 인증할 수 있는 방법이 제공되지 않는다. 공격자가 도청 기능을 가지는 변조된 AP를 제작하여 운영한다면 이 AP에 접속하는 타인의 통신을 쉽게 도청할 수 있을 것이다. 공격자는 또한 프록시 AP를 운영하면서 남의 Wi-Fi 통신이 자신을 거쳐가도록 하여 도청할 수 있는데 AP에 대한 인증기능이 없으므로 이러한 도청공격을 방지하기 어렵다. WPA2-Enterprise 환경에서는 중앙집중식 RADIUS 인증서버를 통해 사용자를 인증하며 조직이 인증한 AP만 사용하게 되므로 공격자가 활동할 여지가 없다. 더구나 인증서버는 자신의 인증서를 클라이언트에 배포하고 사용자 인증정보를 암호화 전송하게 하므로 사용자의 인증정보는 도청공격자에게 노출되지 않으며, 사용자는 제공되는 무선랜 서비스에 신뢰를 가질 수 있다. 1대의 AP만으로 운영되는 WPA2-PSK 모델에서도 AP의 제조사가 AP에 인증서를 발급하여 판매하고 Wi-Fi 접속시 클라이언트가 AP의 인증서를 통해 AP의 신뢰성을 검증할 수 있도록 개선할 필요가 있다.

효율성 측면에서 WPA2 프로토콜은 기본적으로 4-way 핸드쉐이크 프로토콜을 통해 보안세션을 설정하므로 AP와 클라이언트 사이에 여러번의 통신을 요구한다. 더구나 이 과정은 클라이언트가 AP에 접속할때마다 반복되는 작업이다. 한 번 접속했던 클라이언트에게는 재접속시 이러한 과정을 반복하지 않고도 빠르게 세션을 설정할 수 있도록 개선하면 효율성이 크게 향상될 것이다.

2.2 초기인증, 인증유지, 토큰인증

초기인증이란 클라이언트가 서버에 처음 접속시 클라이언트의 자격을 엄밀하게 검증하고 인증하는 것을 말한다. 인증유지란 초기인증을 통해 한 번 인증된 사용자에게는 엄밀한 검증과정을 거치지 않고도 효율적이면서도 안전하게 인증된 상태를 오랜 기간 유지시키는 것을 의미하며[11,12] 이를 위해 서버는 초기인증된 클라이언트에게 토큰 등의 인증유지 수단을 제공하게 된다. 초기인증시 만일 사용자가 제공하는 ID/Password 정보로부터 사용자를 인증한다면 사용자의 인증정보는 서버에게 안전하게 전달되어야

하며 서버가 발급하는 토큰 또한 안전하게 클라이언트에게 전송될 수 있어야 한다.

토큰인증은 웹서비스에서 사용자 인증유지를 위해 널리 사용하는 방법으로 OAuth 2.0 bearer 토큰[4], JWT[5] 등의 방법이 사용된다. 사용자가 초기인증을 통해 웹서버에 한 번 인증되면 웹서버는 서명된 토큰을 발급하게 되며 이것은 웹브라우저의 스토리지에 저장된다. Fig. 3 은 JWT의 구조를 나타내는데 Header와 Payload에 사용자를 확인할 수 있는 정보를 입력하고 Signature 영역에는 Header와 Payload의 정보를 서버의 비밀키와 함께 HMAC 계산한 값을 입력한다. Payload에는 "exp" 필드에 토큰의 유효기간을 명시할 수 있다. 사용자가 해당 웹서버에 재접속시에는 ID/Password를 다시 입력할 필요 없이 토큰을 첨부함으로써 토큰의 유효기간 동안 인증된 상태를 유지할 수 있다. 서버는 클라이언트가 제시한 토큰의 유효성을 검증하기 위해 비밀키를 이용하여 Signature에 포함된 HMAC 값이 일치하는지 검증한다. 이러한 토큰인증은 서버의 운용 효율성을 높일 수 있는 장점이 있지만 도청공격에 취약하므로 초기인증 및 인증유지의 전체 과정을 HTTPS[7] 보안통신환경에서 사용하여야 한다는 제약이 있다.

OAuth 2.0 MAC 토큰[6] 방식은 클라이언트와 서버가 공유비밀키를 이용하여 MAC 인증을 수행하는 방식인데 이것은 매번 달라지는 값을 생성하므로 보안통신을 사용하지 않아도 된다는 장점이 있다. 그러나 서버는 사용자별 공유비밀키를 관리해야 하는 상태형 인증(stateful authentication)으로서 효율성을 중시하는 웹 환경에서의 인증유지에는 사용하기 어렵다는 단점이 있다.

Fig. 3. JSON Web Token

2.3 이중토큰을 이용한 난수화 토큰인증

OAuth 2.0 Bearer 토큰[4] 기술은 HTTPS[7] 보안통신 환경에서만 사용되어야 한다는 제약이 있으며, OAuth 2.0 MAC 토큰[6] 기술은 보안통신을 사용하지 않아도 되지만 무상태 인증을 제공할 수 없다는 단점이 있다. 난수화 토큰인증[11,12] 기술은 이러한 문제점들을 해결하기 위해 제안된 기술로 이중토큰을 이용함으로써 평문통신 환경에서도 무상태 토큰인증을 안전하게 사용할 수 있도록 개선한 기술이다. 이것의 상세 내용을 초기인증 시 토큰발급 과정과 토큰을 이용한 인증유지 과정으로 나누어 설명한다. 서버가 토큰 발급에 사용하는 비밀키를 K라고 하자.

2.3.1 초기인증시 토큰 발급

사용자의 정보를 A라 하자. 사용자가 서버에 대해 초기인증에 성공하면 서버는 추후 인증된 사용자 정보를 쉽게 확인할 수 있도록, 그리고 유효기간 등의 필요한 추가정보를 포함하도록, 사용자 정보 I_A를 생성하고 사용자에게 공개토큰(public token)과 비밀토큰(secret token)의 두 개의 토큰을 발급한다. 먼저 I_A에 대한 서버의 HMAC 값 t_p를 생성하고 이를 Signature에 넣은 JWT 토큰을 공개토큰 T_p로 생성한다. 이후 T_p에 대한 서버의 HMAC 값 t_s를 생성하고 이를 Signature에 넣은 JWT 토큰을 비밀토큰 T_s로 생성한다. 여기서 JWT 토큰을 생성한다는 것은 JWT 양식에 따라 필요한 정보를 Header와 Payload에 넣고 이에 대한 서버의 HMAC 값을 계산하여 Signature에 넣은 토큰을 생성하는 것을 의미한다.

$$t_p = HMAC(I_A, K) \rightarrow T_p \text{ 생성}$$
$$t_s = HMAC(T_p, K) \rightarrow T_s \text{ 생성}$$

클라이언트는 서버와의 인증유지에 사용하기 위해 공개토큰과 비밀토큰을 저장한다. 웹브라우저에서는 로컬스토리지(local storage)에 저장하여 사용함으로써 세션과 상관없이 오랜 기간 서버와의 인증유지에 사용할 수 있다.

2.3.2 토큰을 이용한 인증유지

클라이언트가 공개토큰 T_p와 비밀토큰 T_s를 가지고 있다고 하자. 클라이언트는 자신의 현재시간 $time_c$와 자신의 비밀토큰 T_s를 이용하여 해시함수

계산을 통해 다음과 같이 일회용 인증정보 $auth$를 계산한다.

$$auth = H(time_c, T_s)$$

클라이언트는 서버에 $< T_p, time_c, auth >$를 전송하면서 자신이 인증된 상태임을 증명하고자 한다. 서버는 수신한 인증정보를 검증하기 위해 다음과 같은 검증과정을 수행한다.

1) 공개토큰 T_p에서 사용자 정보를 확인하고 비밀키 K를 사용하여 Signature 필드에 포함된 HMAC 값을 검증하여 자신이 발급한 유효한 토큰인지 확인한다.
2) 비밀키 K를 사용하여 공개토큰 T_p로부터 비밀토큰 T_s을 계산한다.
3) 비밀토큰을 이용하여 $auth$를 동일하게 계산하여 일치하는지 확인한다.
4) 서버는 자신의 현재시간 $time_s$를 체크하고 클라이언트가 보내온 시간 $time_c$와의 차이 $time_{diff} = time_s - time_c$를 계산하여 시간이 정해진 오차범위 내에서 일치하는지 확인한다.

이러한 검증을 통과하면 서버는 클라이언트가 이미 인증된 상태임을 인정하고 서비스를 제공하게 된다. T_p의 유효성을 검증하기 위해서는 서버의 비밀키 K를 사용해야 하므로 이것은 해당 서버만이 검증할 수 있다. 아울러 비밀토큰을 계산하기 위해서도 비밀키 K가 필요하므로 비밀토큰은 해당 서버만이 계산할 수 있으며 이런 검증과정은 토큰을 발급했던 해당 서버만이 수행할 수 있다. 클라이언트와 서버의 현재시간 차이를 검증하는 것은 이러한 일회용 인증통신을 공격자가 도청하더라도 재전송 공격에 사용할 수 없도록 하기 위한 것이다.

공개토큰 T_p는 인증된 사용자임을 나타내기 위해 인증유지시 서버에 전송하는 정보로 서버가 서명한 아이디와 같은 역할을 하며 공개되는 정보이다. 비밀토큰 T_s는 외부로 노출되지 않고 일회용 인증정보 계산에만 사용하는 정보로서 서명된 패스워드와 같은 역할을 한다. 두 개의 토큰은 상호 연관되어 있어서 비밀키 K를 알고 있는 서버는 공개토큰이 주어지면 언제든지 비밀토큰을 계산할 수 있지만 제3자는 공개토큰이 공개된다고 해도 비밀토큰을 계산해낼 수 없다는 특징이 있다. 서버는 클라이언트가 제시하는 공개토큰으로부터 언제든지 비밀토큰을 생성하여 사용할 수 있기 때문에 비밀토큰을 별도로 관리할 필요가 없다. 이렇게 토큰이 발급되면 클라이언트와 서버 사이에는 클라이언트의 신분을 즉시 검증 가능한 비밀채널이 개설되어 있는 것과 같다.

2.3.3 ID/password와 이중토큰의 특징 비교

이렇게 발급된 공개토큰과 비밀토큰은 새로운 인증수단으로 편리하게 사용될 수 있다. 공개토큰과 비밀토큰은 지식 기반의 전통적인 인증수단으로 사용해온 ID/password와 비슷한 역할을 하지만 다음과 같이 큰 차이점을 가지고 있다.

1) 일반적인 ID/password는 사용자가 선택하는 것이고 이들 사이에 특별한 연관관계가 없으나, 공개/비밀토큰은 서버가 서명하여 생성하는 것으로 특별한 연관관계가 있어서 서버는 공개토큰으로부터 비밀토큰을 언제든지 계산할 수 있다는 특징이 있다.
2) ID/password 방식에서는 인증을 위하여 패스워드가 서버에 전달되어야 하는데 패스워드는 사용자를 증명할 수 있는 기밀정보로서 노출을 방지하기 위해 보안통신을 통해 전달되어야 하고 서버는 이것의 기밀성을 보호하기 위하여 패스워드 해시값으로 변환하여 사용자 DB에 저장하게 된다. 반면 공개토큰/비밀토큰 방식에서 비밀토큰은 통신상으로 직접 전달되지 않으며 일회용 인증값을 계산하는 용도에만 사용된다. 그러므로 이중토큰을 이용하는 인증은 평문통신을 통해서도 안전하게 수행될 수 있다.
3) ID/password 방식에서는 사용자가 취약한 패스워드를 선택하여 사용하거나 여러 서버에 동일한 패스워드를 사용하는 등의 경우 취약점이 있을 수 있다. 반면 공개토큰/비밀토큰 방식에서는 서버가 토큰을 서명하여 발급하기 때문에 서버마다 랜덤하게 생성되고 다른 서버에는 사용할 수 없으므로 이러한 취약점이 없다.
4) ID/password 방식에서는 패스워드의 안전한 처리에 대해 사용자는 서버를 전적으로 신뢰해야 한다. 즉 서버가 사용자의 패스워드를 평문으로

저장하고 다른 서버에의 인증에 사용하는 등 불법적인 용도로 사용하지 않을 것이라는 것을 신뢰할 수밖에 없다. 반면 공개토큰/비밀토큰 방식에서는 이 토큰이 해당 서버에서만 사용 가능하고 다른 용도에 사용될 수 없으므로 이러한 문제가 발생하지 않는다.

5) ID/password 방식에서는 사용자 인증을 위해서 서버는 사용자 DB에 등록된 패스워드와 엄밀하게 비교해야 하므로 상태형 인증(stateful authentication)이 수행되어야 한다. 반면 공개토큰/비밀토큰 방식에서는 사용자가 제시하는 공개토큰을 이용하여 비밀토큰을 계산할 수 있고 인증을 즉시 검증할 수 있으므로 무상태형 인증(stateless authentication)이 가능하며 이것은 서버측의 효율성을 높이는데 큰 역할을 할 수 있다.

6) ID/password 방식은 사용자가 기기를 직접 동작시키면서 키보드 등으로 패스워드를 직접 입력하는 형식으로 널리 사용되는데 토큰인증 방식은 사용자 동작이 필요없는 자동화된 인증 환경에 더 안전하고 편리하게 사용할 수 있다.

III. 이중토큰 기반 Wi-Fi 보안 프로토콜

우리는 위에서 제시한 이중토큰 기반의 난수화 토큰인증 기술을 Wi-Fi 보안 프로토콜에 결합하여 활용함으로써 보안세션 설정 과정의 효율성을 높이려고 한다. AP의 인증성을 제공하기 위해 AP에는 제조사가 발급한 인증서 $Cert_{AP}$가 장착되어 있다고 가정한다. AP의 사용을 허용하기 위해 클라이언트를 인증하는 것은 WPA2-PSK에서 널리 사용되어온 바와 같이 공유비밀키를 이용하며 클라이언트와 AP는 동일한 공유비밀키 PSK를 가지고 있다고 가정한다. AP는 토큰발급에 사용하는 비밀키 K를 가지고 있다고 하자.

3.1 초기인증 및 이중토큰 발급

클라이언트는 AP가 제공하는 무선랜 서비스를 이용하고자 하며 동일한 공유비밀키 PSK를 가지고 있다. 클라이언트가 AP에 초기인증을 수행하고 이중토큰을 발급받는 과정은 다음과 같다.

1) 클라이언트 → AP : 무선랜 서비스 요청
2) AP → 클라이언트 : 인증서 $Cert_{AP}$ 배포
3) 클라이언트 → AP : 클라이언트는 AP의 인증서를 검증함으로써 신뢰할 수 있는 AP임을 확인하고 유효하지 않은 경우 프로세스를 중단한다. 클라이언트는 난수키 r을 생성하여 AP의 인증서에 포함된 공개키로 암호화하여 $E_{pk}(r,pk_{AP})$를 AP에게 전송한다.
4) AP → 클라이언트 : AP는 수신한 정보를 자신의 개인키 sk_{AP}로 복호화하여 난수키 r을 획득한다. 난수 challenge 값 c 를 생성하여 클라이언트에게 전송하고 인증을 요구한다.
5) 클라이언트 → AP : 클라이언트는 난수키 r을 이용하여 공유비밀키 PSK와 c를 암호화하여 $E(PSK\|c,r)$을 AP에게 전송한다.
6) AP → 클라이언트 : AP는 난수키 r을 이용하여 인증정보를 복호화하고 공유비밀키가 일치하는지 확인하며 일치하지 않는 경우 프로세스를 중단한다. 사용자가 인증되면 2.3절에서 설명한 바와 같이 AP는 비밀키 K를 이용하여 공개토큰 T_p와 비밀토큰 T_s를 발급하고 난수키 r로 암호화하여 $E(T_p\|T_s,r)$를 클라이언트에게 전송한다.
7) 클라이언트 : 클라이언트는 난수키로 복호화하여 공개토큰 T_p와 비밀토큰 T_s를 획득하고 시스템에 $[AP, T_p, T_s]$를 저장한다.

초기인증 및 이중토큰 발급 과정은 기밀정보를 주고받는 것으로 암호화 통신 환경에서 수행되어야 하므로 AP의 인증서를 이용하여 난수키 r을 공유하고 이를 이용하여 암호화 통신을 하도록 설계되어 있다. AP가 생성한 challenge c를 이용하는 것은 도청자에 의한 재전송 공격 및 서비스거부 공격을 방지하기 위한 것이다.

발급받은 토큰의 유효성을 검증하기 위해서는 AP의 비밀키 K가 필요하므로 클라이언트는 발급받은 공개토큰과 비밀토큰의 유효성을 직접 검증할 수는 없다. 다만 이것을 이용하는 이후의 보안세션 설정과정이 성공하면 유효한 것으로 판단하게 된다. 이러한 초기인증 과정은 해당 AP에 처음 접속하게 되는 경우, 또는 토큰의 유효기간이 지나 재발급 받아야 하는 경우에만 실행된다. 클라이언트가 여러 AP를 사용하게 되는 경우 AP마다 발급받은 토큰을 저장해

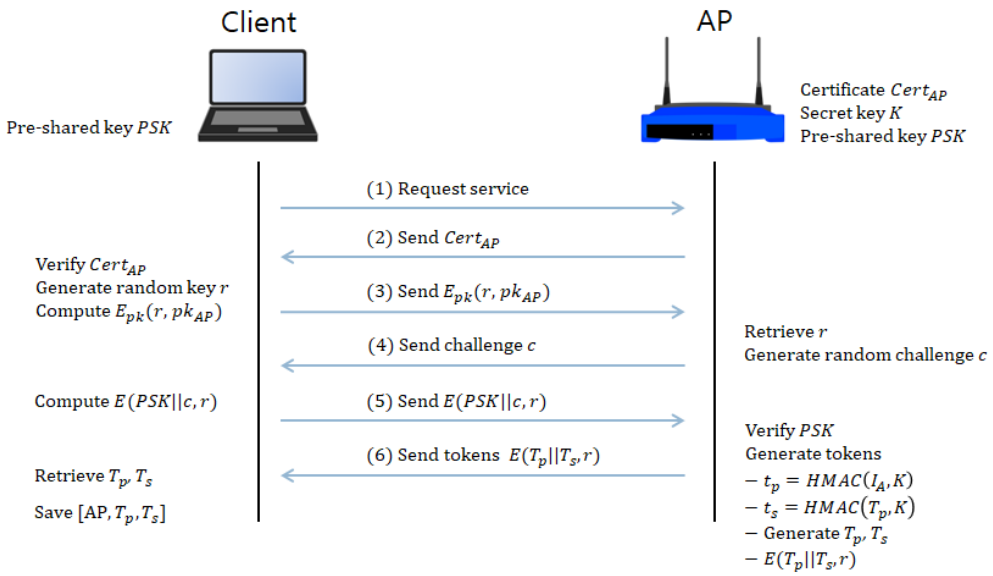

Fig. 4. Initial authentication and issuing double tokens

야 하고 해당 AP만이 해당 토큰을 사용할 수 있도록 접근제어가 필요하다.

3.2 이중토큰을 이용한 보안세션 설정

공개토큰 T_p와 비밀토큰 T_s를 발급받은 클라이언트가 AP와 보안세션을 설정하는 과정은 다음과 같다.

1) 클라이언트 → AP : 클라이언트는 자신의 현재시간 $time_c$와 자신의 비밀토큰 T_s를 이용하여 해시함수 계산을 통해 일회용 인증정보 $auth = H(time_c, T_s)$를 계산한 후 AP에게 $< T_p, time_c, auth >$를 전송한다.
2) AP → 클라이언트 : AP는 T_p의 클라이언트 정보를 확인하고 signature를 검증하여 자신이 발급한 유효한 토큰인지 검증한다. AP는 공개토큰 T_p로부터 비밀토큰 T_s를 계산하고 이로부터 동일한 방법으로 $auth = H(time_c, T_s)$를 계산하고 일치하는지 확인한다. AP는 자신의 현재시간 $time_{AP}$를 새로 생성하고 클라이언트의 현재시간 $time_c$가 정해진 오차범위 내에서 일치하는지 검증한다. 만일 어느 하나라도 유효하지 않으면 보안세션 설정 프로세스를 중단한다. AP는 세션 비밀키 $key = H(time_{AP}, auth, T_s)$와 인증코드 $mac = HMAC(time_{AP}, key)$를 계산하고 $< time_{AP}, mac >$을 클라이언트에게 전송한다. AP는 현재 접속된 클라이언트를 관리하는 메모리/DB에 $[T_P, key, time_{AP}]$를 저장한다.
3) 클라이언트 : 클라이언트는 비밀토큰 T_s를 가지고 있으므로 수신한 $time_{AP}$를 이용하여 동일한 세션비밀키 $key = H(time_{AP}, auth, T_s)$를 계산하고 mac의 유효성을 검증한다. 클라이언트는 $[AP, key, time_{AP}]$를 저장한다.

이렇게 공유된 세션비밀키 key는 전통적인 4-way 핸드쉐이크 프로토콜에서의 pairwise transient key(PTK)와 동일한 역할을 하며 이로부터 key confirmation key(KCK), key encryption key(KEK), temporal key(TK)등을 유도하여 사용하면 된다.

보안세션 설정은 클라이언트가 AP로 1회의 메시지를 주고 응답을 받으면 즉시 설정될 수 있으며 이러한 설정과정은 도청되더라도 재전송공격에 사용될 수 없으므로 평문통신 채널을 통해서도 안전하게 수행될 수 있다. 클라이언트가 AP에 재접속하는 경우

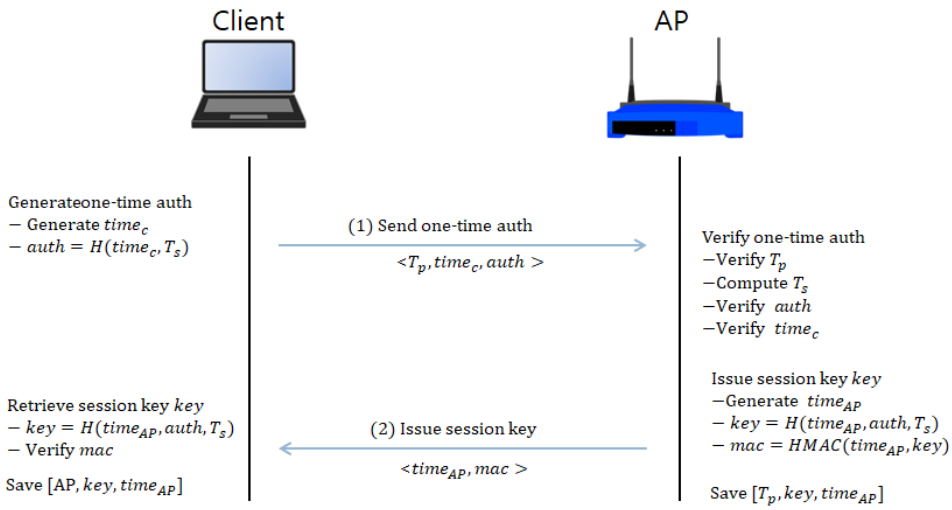

Fig. 5. Setting up secure session using double tokens

클라이언트가 토큰을 발급받은 상태이면 복잡한 초기 인증을 반복하지 않고 평문통신을 통해 1회의 메시지를 주고받음으로서 즉시 보안세션을 설정할 수 있다는 장점이 있다. 이것은 실제의 Wi-Fi 사용 환경을 고려해보면 매우 큰 장점을 가짐을 알 수 있다. AP는 수시로 접속하는 많은 사용자들에게 무선랜 서비스를 제공하게 되는데 기존의 WPA2 프로토콜에서는 매번 여러번의 통신을 통해 사용자의 인증정보를 검증하고 보안세션을 설정하는데 비해 제안방식에서는 이미 토큰을 발급받은 인증된 클라이언트와는 1회의 평문통신을 주고받으면 즉시 보안통신 세션을 설정할 수 있게 되는 것이다.

3.3 보안세션 갱신

WPA2에서는 키갱신 메커니즘을 사용하여 갱신시간(renewal time)이 지나면 세션비밀키를 새로 생성하여 공격자들의 공격을 어렵게 한다. 제안된 방식에서는 클라이언트와 AP가 동일한 세션키 key를 가지고 있으며 이를 이용하여 보안세션 갱신을 안전하게 수행할 수 있다.

먼저 갱신시간이 지나면 AP가 보안세션 갱신을 요구하게 되는데 보유한 key를 이용하여 인증값을 다음과 같이 계산하여 보낸다.

$$req = H(AP, time_{AP}, time_{curr}, T_P, key)$$

이를 수신한 클라이언트는 동일한 key를 이용하여 인증값을 검증하고 유효한 경우 앞에서 기술한 이중토큰을 이용한 보안세션 설정과정을 수행한다.

IV. 분 석

4.1 제안 방식의 특징

제안된 새로운 Wi-Fi 보안 프로토콜은 WPA2, WPA3의 경우와 비교할 때 다음과 같은 특징을 가진다.

1) 초기인증과 보안세션 설정의 분리 : 기존의 공유비밀키를 이용하는 WPA2/WPA3 인증 방식에서는 공유비밀키(PSK)를 이용하여 인증도 수행하고 이로부터 세션비밀키를 설정하는데 이들은 4-way 핸드쉐이크 과정을 통해 항상 동시에 실행된다. 반면 제안된 방식에서는 공유비밀키를 이용하는 초기인증과 이중토큰을 이용하는 보안

Table 1. Comparison of features

Features	WPA2	Proposed
Initial auth.	PSK	PSK
Keeping auth.		tokens
AP auth	No	Yes
state	stateful	stateless

세션설정을 분리하였으며 세션비밀키는 PSK와 연관성이 없다. 이러한 분리를 통해 한 번 초기 인증을 받은 클라이언트는 재접속시 이중토큰을 이용한 보안세션 설정만을 수행하면 된다.

2) 새로운 인증수단으로 이중토큰을 활용 : 초기인증이 완료되면 클라이언트는 AP가 서명하여 발급하는 공개토큰과 비밀토큰 정보를 가지게 되는데 이를 이용하면 AP와 1회의 평문통신을 주고받음으로써 보안세션 설정을 매우 빠르게 수행할 있다. 이런 인증방식은 자동화된 인증 및 보안세션 설정에 효율적인 방식이다.
3) AP의 무상태 인증 : AP는 초기인증된 클라이언트에게 자신에게 접속할 수 있는 권한을 가진 토큰을 발급해주는데 이들을 특별히 관리할 필요가 없다. 공개토큰은 클라이언트가 전송해 올 것이며 비밀토큰은 공개토큰으로부터 즉시 계산하여 사용할 수 있다. 그러므로 AP는 클라이언트 인증시 자신의 DB를 검색할 필요 없이 즉시 인증 가능한 무상태 인증이 가능하다.
4) 인증서를 이용한 AP의 신뢰성 검증 : 클라이언트는 AP의 인증서를 검증해봄으로써 신뢰할 수 있는 AP인지 확인할 수 있다. 지금까지 Wi-Fi 보안 프로토콜에 대해 제기된 중간자공격, 가짜 AP를 이용한 도청공격 등의 많은 취약성들은 AP의 신뢰성을 확인할 수 없음으로 인한 것이었다. 인증서를 활용하는 PKI 인증체계는 이미 널리 사용되고 있으며 WPA2-Enterprise 버전에서는 이미 활용되고 있는데 이를 일반 PSK 모드에서도 활용할 수 있도록 적용한 것이다.

4.2 보안성 분석

WPA2에 프로토콜에 적용되었던 기존의 공격방법들에 대해 제안방식의 안전성을 비교 검토해보자.

1) 패스워드에 대한 오프라인 공격 : WPA2 프로토콜에서는 공유비밀키(패스워드)가 알려지지 않은 상태에서 보안세션 설정 패킷들을 도청하게 되면 Aircrack 등의 도구들을 이용하여 패스워드에 대한 오프라인 공격을 시도해볼 수 있다. 만일 사용자가 패스워드 사전에 나올만한 취약한 패스워드를 사용하게 된다면 패스워드를 쉽게 찾아낼 수 있을 것이다. 제안된 방식에서는 패스워드 등 인증정보가 AP의 인증서를 이용하여 공유하게 되는 난수키 r로 암호화되어 전달되므로 이러한 사전공격을 수행할 수 없다.
2) Forward secrecy : WPA2의 경우 공유비밀키를 알게 된다면 이전에 도청된 보안세션 설정 정보로부터 남의 세션비밀키를 계산해낼 수 있어서 전체 통신 내용을 도청할 수 있게 된다는 취약점이 있다. 그러나 제안방식에서는 공유비밀키가 초기인증과정의 클라이언트 자격 확인에만 사용되고 실제 세션비밀키는 AP가 클라이언트별로 발급하는 토큰으로부터 계산되므로 이러한 공격에 대해 안전하다. 즉 공유비밀키를 공개하는 공개 무선랜으로 운영되더라도 공격자가 남의 통신을 도청할 수는 없게 된다.
3) KRACK 공격[10] : 이것은 사용자의 4-way 핸드쉐이크 프로토콜에서 공격자가 난수를 재사용하도록 유도함으로써 통신을 도청하는 공격기법이다. 제안방식에서는 4-way 핸드쉐이크 방식을 사용하지 않으므로 난수를 재사용하도록 하는 이런 공격방식이 적용되지 않는다.
4) 도청공격, 가장공격, 중간자공격 : WPA2의 경우 공개된 PSK를 알고 있는 공격자는 타인의 세션비밀키를 계산해내고 통신을 쉽게 도청할 수 있어서 많은 취약점을 가지고 있다. 타인의 통신을 도청하는 수동적 공격, 타인의 통신을 방해하고 타인으로 신분을 가장하는 공격, 변조된 가짜 AP를 운영하면서 중간자 공격을 통해 타인의 통신을 도청 또는 변조하는 공격 등 다양한 취약점이 존재한다. 제안 방식에서는 클라이언트가 AP의 인증서를 검증하여 AP의 신뢰성을 확인한 이후 이중토큰을 안전하게 발급받고 이에 기반하여 안전한 보안세션을 설정하여 암호화 통신을 수행하므로 이러한 중간자 공격이 어렵다. 공격자는 통신을 도청하더라도 비밀토큰을 획득하지 못하면 이러한 공격을 수행할 수 없다.
5) 비밀토큰에 대한 시스템 해킹 공격 : 비밀토큰은 초기인증된 클라이언트 시스템에 저장하여 사용하는 기밀정보이다. 공격자가 다른 사용자의 클라이언트 시스템을 직접 해킹하여 이들 정보를 취득하는 경우 타인의 통신을 도청할 수 있게 될 것이다. 이러한 시스템 해킹공격은 기밀정보를 시스템에 저장하여 이용하는 모든 인증시스템에 공통적으로 적용되는 취약점이다. 이러한 해킹공격에 대응하기 위해서는 기밀정보를 안전한 저장소에 저장하여 사용하여야 한다.

Table 2. Comparison of security

Attacks	WPA2	Proposed
Offline password dictionary attack	Y	N
Forward secrecy	Y	N
KRACK attack	Y	N
System hacking	Y	Y

4.3 효율성 분석

1) 초기인증과 보안세션 설정의 분리로 인한 효율성 향상 : WPA2, WPA3에서는 초기인증과 보안세션 설정이 구분되어 있지 않다. 4-way 핸드쉐이크 프로토콜을 통해 PSK를 이용하여 인증을 수행하고 아울러 이 정보를 이용하여 보안세션을 설정하게 되는데 이것은 매번 접속시마다 전체를 수행해야 한다. 반면 제안 방식에서는 1회의 초기인증을 통해 이중토큰을 발급받으면 이후에는 토큰의 유효기간 동안 n회의 보안세션 설정만을 수행할 수 있으며 보안세션 설정과정은 해시함수 기반의 인증메시지를 1회 주고받음으로써 완료할 수 있어서 매우 효율적이다.
2) 무상태 보안세션 설정으로 인한 효율성 : WPA2, WPA3 프로토콜에서는 매 접속시마다 인증서버에 사용자의 인증정보를 전송하고 검증해야 하므로 상태형 서비스가 되지만 이중토큰을 이용한 보안세션 설정과정은 클라이언트가 전송하는 정보만으로 인증과 보안세션설정이 완료되므로 무상태 서비스가 가능하다. 특히 Enterprise 환경에 적용하는 것을 고려하면 WPA2, WPA3의 경우 사용자별 인증정보를 인증서버를 통해 매번 검증해야 하는 것과 비교할 때 이중토큰을 이용하는 무상태 보안세션 설정은 매우 높은 효율성을 제공한다.
3) 초기인증의 효율성 비교 : WPA2에서는 해시함수 기반으로 4-way 핸드쉐이크 프로토콜을 수행하므로 효율성이 높다고 볼 수 있다. WPA3에서는 WPA2에서 제기된 취약점 해결을 위해 simultaneous authentication of equals (SAE), opportunistic wireless encryption (OWE) 등의 방식으로 모듈러승산을 추가한 4-way 핸드쉐이크 프로토콜을 사용하므로 WPA2에 비해 많은 계산량이 필요하다. 제안 방식에서는 초기인증시 AP의 인증서 검증, AP의 공개키를 이용한 인증정보 암호화 전송 등 공개키암호를 사용하는데 WPA3 이상의 계산량이 소요될 것이다. 그러나 이러한 비용은 인증서를 이용해 AP의 신뢰성을 제공하기 위한 필수적인 비용이다.

Table 3. Comparison of performance

Performance	WPA2	WPA3	Proposed
Initial auth.	H	L	L
Session est.			H
Stateless	N	N	Y

4.4 구현시 고려 사항

제안된 방식은 전통적인 WPA2 및 개선된 WPA3 방식과 여러 가지 점에서 차이가 있어서 효율적인 구현 환경을 위한 검토가 필요하다.

1) AP의 인증서 발급 방안 : 제조사에서는 AP별로 인증서를 설치하여 판매하여야 한다. 제조사는 인증기관으로부터 발급받은 마스터 인증서를 가지고 있다고 가정한다. 판매하는 AP에는 마스터 인증서를 이용하여 발급한 사설 인증서를 설치하여 배포하며 검증을 위해 마스터 인증서도 함께 배포한다. 클라이언트가 AP에 처음 접속할때는 AP가 배포하는 마스터 인증서와 AP의 인증서를 통해 AP의 유효성을 검증하게 되고 클라이언트가 인터넷에 연결되면 공개키기반구조(PKI) 체계를 통해 마스터 인증서의 유효성도 검증할 수 있게 된다.
2) AP의 보안정책 설정 방안 : 이중토큰 발급에 사용하는 AP의 비밀키 K는 외부로 공개될 필요가 없이 AP 내부에서만 사용되는 비밀정보이므로 AP의 보안정책에 따라 내부에서 난수로 생성될 수 있고 규칙적으로 자동 갱신되도록 할 수 있다. 비밀키가 갱신된다면 기존의 토큰들은 무효화되어 클라이언트들은 초기인증을 다시 요구받게 될 것이다. 토큰의 유효기간은 공개토큰의 Body에 "exp" 필드로 지정되는데 이것의 길이는 AP의 보안정책에 따라 지정될 수 있다. 이러한 보안정책은 AP의 관리자가 직접 설정하도록 할 수 있다.

3) 정확한 시간정보 활용 : 제안된 방식은 보안세션 설정, 보안세션 갱신 등에서 클라이언트 및 AP의 시간정보를 활용하게 되는데 이것은 공격자의 도청 및 재전송 공격을 무력화하는데 핵심적인 역할을 한다. 만일 두 시스템의 시간정보가 다르다면 Wi-Fi 프로토콜의 정상적인 운영에 문제가 발생할 수 있다. 최근의 컴퓨터 시스템 및 운영체제에서는 네트워크 시간 프로토콜 (Network Time Protocol)을 활용하여 정확한 시간을 쉽게 사용할 수 있다.

V. 결 론

이 논문에서는 WPA2 프로토콜이 4-way 핸드쉐이크 프로토콜을 사용함으로 인해 여러 가지 취약점과 비효율성을 가지고 있음을 보였으며 이 문제를 해결하기 위해 웹서비스에서 널리 사용되고 있는 토큰 인증 기술을 적용하고자 하였다. 또한 OAuth 2.0 bearer 토큰과 OAuth 2.0 MAC 토큰 기술이 가지고 있는 문제점을 해결할 수 있는 이중토큰을 이용하는 난수화 토큰인증 기술을 제시하고 이것을 Wi-Fi 보안 프로토콜에 적용하는 새로운 방법론을 제시하였다.

제안된 방식에서는 Wi-Fi 보안에 있어서 초기인증과 보안세션 설정 과정을 분리하였으며 초기인증이 완료되면 클라이언트는 AP가 발급하는 공개토큰, 비밀토큰을 보유하게 되고 이를 이용하는 보안세션 설정 과정은 평문통신을 통해 1회의 인증메시지를 주고받음으로써 매우 효율적으로 수행될 수 있다. 사용자는 동일한 AP를 통해 오랜 기간 무선랜 서비스를 제공받고자 하는데 보안세션 설정이 효율적인 이러한 방식을 사용하면 클라이언트 입장에서의 편의성과 함께 AP 입장에서의 효율성도 크게 향상될 수 있다.

WPA3 기술은 기존의 WPA2가 가지고 있는 문제점을 해결하기 위해 제안된 기술이지만 기술의 연속성을 고려하기 위해 기존의 4-way 핸드쉐이크 방법론을 유지하는 소규모 업그레이드라고 생각된다. 반면 제안 방식은 WPA2, WPA3와 비교하면 전혀 새로운 프로토콜로서 초기인증과 보안세션 설정을 분리하고 해시함수 기반의 인증메시지를 1회 주고받음으로써 안전한 보안세션을 설정할 수 있도록 하여 효율성을 획기적으로 개선하였다. 이러한 방식이 Wi-Fi 산업계에서 적극 검토되고 새로운 기술표준으로 채택된다면 Wi-Fi의 안전성과 효율성을 획기적으로 개선할 수 있을 것으로 예상된다.

References

[1] Kevin Benton, The Evolution of 802.11 Wireless Security, UNLV Informatics, Spring 2010. https://benton.pub/research/benton_wireless.pdf

[2] Wi-Fi Protected Access II (WPA2), IEEE 802.11i-2004.

[3] WPA3, https://www.wi-fi.org/

[4] Michael B. Jones and Dick Hardt, "The OAuth 2.0 authorization framework: bearer token usage," RFC 6750, Oct. 2012.

[5] Michael B. Jones, John Bradley, and Nat Sakimura, "JSON web token (JWT)," RFC 7519, May 2015.

[6] Justin Richer, William Mills, Hannes Tschofenig, and Phil Hunt, "OAuth 2.0 message authentication code (MAC) tokens," Internet-Draft, Jan. 15, 2014. https://tools.ietf.org/id/draft-ietf-oauth-v2-http-mac-05.html

[7] Eric Rescorla, "HTTP over TLS," RFC 2818, May 2000.

[8] Dan Harkins, Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks, 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008), pp. 839-844, Aug. 2008.

[9] Dan Harkins and Warren Kumari, Opportunistic Wireless Encryption, RFC 8110, Mar. 2017.

[10] Mathy Vanhoef and Frank Piessens, Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1313-1328, Nov. 2017.

[11] Byoungcheon Lee, "Strengthening of

token authentication using time-based randomization," Journal of Security Engineering, 14(2), pp. 103-114, Apr. 2017.

[12] Byoungcheon Lee, "Stateless randomized token authentication for performance improvement of OAuth 2.0 MAC token authentication", Journal of The Korea Institute of Information Security & Cryptology, 28(6), pp. 1343-1354, Dec. 2018.

[13] Scott Fluhrer, Itsik Mantin, and Adi Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", Selected Areas of Cryptography: SAC 2001, Lecture Notes in Computer Science 2259, pp. 1-24, Mar. 2001.

〈저자소개〉

이 병 천 (Byoungcheon Lee) 종신회원
1986년 2월: 서울대학교 물리학과 졸업
1988년 2월: 서울대학교 물리학과 석사
2002년 2월: KAIST 정보보호 박사
2002년 3월~현재: 중부대학교 정보보호학과 교수
〈관심분야〉 정보보호, 암호, 인증, 네트워크보안, 웹보안, IoT보안