EAP 인증/키설정 프로토콜 비교분석

박동국* · 조경룡*

A Comparative Analysis of EAP Authentication/Key-Establishment Protocols

DongGook Park* · Kyung-Ryong Cho*

요 약

EAP (Extensible authentication protocol) 프로토콜은 사실, IP 기반 위에서 다양한 유무선 접속 환경에 맞는 여러 가지 인증/키설정 (authentication and key establishment) 프로토콜을 수용할 수 있게 해주는 일종의 큰 틀이라고 할 수 있다. EAP와 함께 쓰일 수 있는 다양한 인증/키설정 프로토콜이 IETF에서 표준화되고 있고, 실제 환경에서 쓰이고 있다. 본 논문은, 이들 프로토콜 중에서 대표적인 것들 여섯 개를 골라서 비교분석하고, 일부 유력한 패스워드 방식 인증/키설정 프로토콜을 둘러싼 지적재산권 분쟁의 여지에 대하여 기술적 해석을 시도하였으며, 이용환경 관점에서 어떤 취사 선택을 해야 할 것인가를 결론부분에서 제안하였다.

ABSTRACT

EAP (Extensible authentication protocol) is a sort of general framework for authentication rather than a specific authentication protocol. An important consequence of this is that EAP can accommodate a variety of authentication/key-establishment protocols for different internet access networks possibly integrated to a common IP core network This paper tries a comparative analysis of several specific authentication/key establishment protocols for EAP, and suggest a strategic viewpoint toward the question: which one to use. In addition, we tried to make things clear about an intellectual property right issue with regard to some password-based protocols.

키워드

EAP, WLAN, 인증, 키설정, 프로토콜

Ⅰ. 서 론

EAP (Extensible authentication protocol) 프로토콜은 원래 PPP 접속 프로토콜 환경에서의 인증/키설정을 위한 IETF표준이지만 [1], IEEE 802.1X 규격의 [2] 완성으로 IEEE 802.11 등의 WLAN (Wireless LAN) 환경에도 적용될 수 있게 되었다. EAP 프로토콜은 이름 그대로, 특정 인증 프로토콜이 아니라 임의의 인증 프로토콜을 덧붙일 수 있는 (extensible) 일종의 멀티콘센트라고 볼 수 있다. 여기에 기존 또는 미래의 인증 프로토콜을 EAP 규격에만 맞추면 얼마든지 EAP 용 인증 프로토콜로 이용할 수 있게 된다.

본 논문에서는, 현재 IETF RFC 형태로 확정된 EAP 용 인증 프로토콜과 현재 IETF에서 드래프트 상태에

* 순천대학교 정보통신공학부

접수일자 : 2005. 7. 4

있는 EAP 용 인증 프로토콜 몇 가지를 비교,분석하고, 이들 프로토콜 선택시 고려해야 할 점들에 관해 논하고자 한다.

Ⅱ. 인증 및 키설정 프로토콜

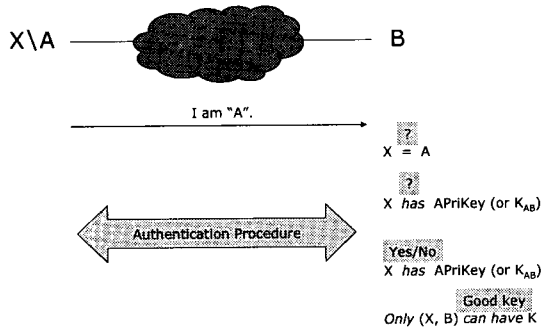


그림 1. 암호학적 인증 및 키 설정
Fig. 1. Cryptographic authentication/key-establishment

그림 1은 인증이 암호학적으로 어떻게 구현되는지 개념적으로 설명하고 있다. 우리는 B의 입장에서 인터넷을 보도록 하자. 더욱 실감나도록 하자면, 우리(B)가 KT나 하나로와 같은 ISP 사업자라고 가정하자. 저 구름 너머에서 어떤 사람이 자기 이름(ID)을 *A*라고 밝히면서 - 일종의 "identity claim" - 인터넷 접속을 요구, 즉 로긴(log-in)을 시도하여 왔다. 하지만, *B*의 입장에서 볼 때, 이 사람이 진짜 *A*인지 아닌지는 알 수가 없다. 즉, 자신을 *A*라고 주장하는 사람 *X*가 정말로 *A*인지 알 수 없다. 이것을 기호를 써서 나타내면,

$$X = A$$

가 되고, 이것을 암호학적으로 번역하면

$$X\ has\ APriKey$$

처럼 될 것이다. 여기서 *APriKey*는 *A*의 비밀개인키 (secret private key)를 나타낸다. 물론, 이것은 비대칭키 (asymmetric key) 즉 공개키 (public-key) 방식 인증 프로토콜을 가정할 때 그렇다. 만약 인증 프로토콜이 공개키가 아닌 기존 대칭키 (symmetric key) 즉 비밀키 (secret-key) 방식일 때는 다음처럼 표현할 수 있다.

$$X\ has\ K_{AB}$$

여기서 K_{AB}는 *A*와 *B*가 서로 공유하는 비밀키를 말한다.

비밀키 방식이든 공개키 방식이든, "*X*가 *A*만 알고 있는 비밀정보(credential)를 - 비밀키 방식 인증 프로토콜이라면 K_{AB}가 될 것이고, 공개키 방식 인증이라면 *APriKey* - 알고 있는가"라는 질문을 확인해야 하며, 이 검증과정이 바로 인증 프로토콜이다. 검증을 위한 인증 메시지 교환 절차를 거친 다음, *X*가 관련 키를 알고 있다고 확인되면 인증에 성공한 셈이 된다. 이렇게 인증 받은 사용자 *X*, 즉 *A*를 상대로 *B*는 후속 데이터 암호화에 쓸 키를 설정할 수 있다. 이런 세션 암호화용 키를 암호학에서는 간단히 세션 키(session key)라고 부른다. 이 키의 성질은, "오로지 *X*(*A*)와 *B*만이" 알 수 있는 값이어야 한다는 것이다. 이런 성질을 가진 세션 키를 "good key between *A* and *B*"라고 부를 수 있다.

현재 인터넷 환경에서 널리 사용되는 이용자 ID/패스워드 인증은 앞에서 설명한 비밀키 방식에 속한다. 다만, 어떤 장치 대신 사람이 비밀키를 기억하고 있다가 필요시 직접 입력할 때, 이를 패스워드라고 부른다. 장치에 저장된 비밀키를 이용하여 인증하는 경우의 좋은 예로는 바로 IEEE 802.11 WLAN의 WEP 인증 프로토콜이 있다 [3] (그림 2). 이 인증 방법을 IEEE 802.11에서는 "Shared Key" 인증이라고 부른다. 즉, WLAN 단말기와 AP가 서로 공유하는 비밀키를 이용한다. 물론, 이 인증 프로토콜에서는 이용자 ID 대신 WLAN 카드의 MAC 주소가 식별 (identification) 정보로 이용된다.

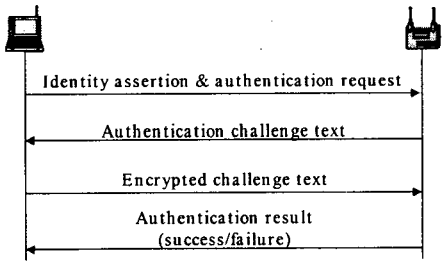


그림 2. IEEE 802.11의 Shared Key 인증
Fig. 2. Shared Key authentication in IEEE 802.11

Ⅲ. EAP 인증 프로토콜

EAP 프로토콜은 IETF에서 1998년 확정한 PPP 용

인증 프로토콜이다. EAP 프로토콜의 주 목적은, NAS(Network Access Server)의 변경 없이도 새 인증 프로토콜을 PPP (Point-to-Point Protocol) 접속 환경에 수용할 수 있게 하자는 것이다. 그 결과, NAS는 (WLAN 환경이라면 AP) 일종의 중간 경유점 (pass-through)으로만 동작하고 EAP 에 쓰이는 구체적인 인증 메커니즘을 알아야 할 필요가 없게 되었다.

EAP 패킷에는 Request/Response/Success/ Failure 네 가지가 있으며 (그림 3), 대부분의 구체적 인증 프로토콜 메시지는 Request/Response 두 가지 패킷 안에 캡슐화되어 교환된다.

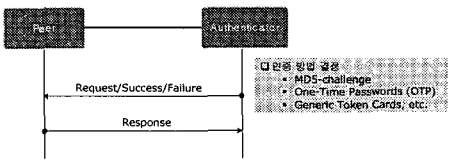


그림 3. EAP 프로토콜 패킷 타입
Fig. 3. EAP Protocol packet types

대부분의 EAP 용 후보 프로토콜은 크게 세 가지 범주로 나누어 볼 수 있다.

- ID/패스워드 기반 인증 프로토콜
- 공개키 인증서 (public-key certificate) 기반 인증 프로토콜
- 비밀 난수키 기반 인증 프로토콜

이제, 이 세 가지 범주에 속하는 대표적 AP 용 인증 프로토콜을 차례로 살펴 보고, 중요 이슈를 짚어 보기로 한다.

1. EAP-MD5

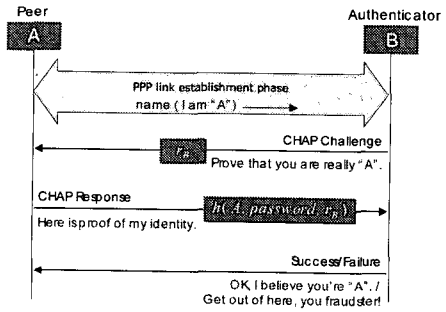


그림 4. CHAP 인증 프로토콜
Fig.4. CHAP authentication protocol

인터넷 접속 프로토콜인 PPP에서는 세 가지 인증 프로토콜을 쓸 수 있다. 첫째는 PAP(Password Authentication Protocol)이고 [4], 둘째는 CHAP (Challenge Handshake Authentication Protocol)이다 [5]. 마지막 방법은 바로 EAP다. 현재 국내 ISP 사업자들도 쓰고 있는 PAP는 사실 암호학적 기법이 전혀 쓰이지 않는다. 즉, 이용자단말에서 NAS까지 패스워드가 그냥 평문으로 전달된다. 따라서 유선을 따서 도청하면 (소위, "wiretapping") 패스워드를 알 수 있게 된다. 반면, CHAP를 쓸 경우는 시도-응답 (challenge-response) 기법을 이용함으로써, 패스워드를 평문형태로 노출하지 않으므로 이와 같은 취약점이 없다.

이제, CHAP 프로토콜의 구체적 절차를 살펴보자. 먼저, CHAP 링크 설정 단계에서 이용자의 ID 즉, A가 네트워크 B로 - 구체적인 예를 들면, 전화국내에 있는 NAS로 - 전달된다. B는 "CHAP challenge" 메시지를 이용자 단말기로 보낸다. 이에 대한 응답으로 A는 "CHAP Response"를 B로 보내고, B는 인증 결과 즉, 성공/실패를 A에게 통보한다.

이러한 메시지들의 상식적 의미는 각각 다음과 같다.

- CHAP Challenge: Prove that you're really A.
- CHAP Response: Here is proof of my identity A.
- Success (Failure): OK, I believe you're A (Get out of here, you fraudster!).

이 메시지들이 암호학적으로는 다음과 같이 구현된다.

- CHAP Challenge: r_B (B가 고른 난수(亂數)로서, 전혀 예측 불가능한 값이다.)
- CHAP Response: $h(A, password, r_B)$ (ID 값 A와 패스워드 그리고 r_B를 해쉬 함수 h( )에 입력하여 얻은 결과로서, 이 결과로부터 원래의 입력값을 역추적할 수 없다.)

패스워드를 알고 있는 네트워크 B는 이용자 단말기로부터 받은 해쉬 결과를 자신도 만들 수 있다. 만약, 해쉬 결과 생성을 위해 이용자가 입력한 패스워드가 네트워크 B가 저장해둔 A의 패스워드와 동일하다면, 이용자로부터 받은 해쉬 결과와 네트워크 B가 스스로

생성한 해쉬 결과가 서로 일치할 것이다. 이것은, 이용자가 올바른 패스워드를 알고 있다는 증거가 되므로 인증에 성공한다. 그 반면 두 해쉬 값이 서로 다르다면, 이것은 이용자가 A의 패스워드를 모르고 있다는 얘기가 되므로, 그 이용자가 진짜 A가 아닌 것으로 간주되어 인증에 실패하게 된다.

EAP-MD5라고 부르는 프로토콜은 다름 아닌 이 CHAP 프로토콜이 EAP와 결합되어 쓰일 때 부르는 이름이다. 따라서 별도의 RFC 규격이 없고, 기존 CHAP 규격을 준용한다. 현재 국내의 대표적 WLAN 서비스인 NESPOT에서도 쓰고 있는 EAP-MD5, 따라서 CHAP 프로토콜의 한 가지 단점은 - 적어도 무선 환경에서는 - 암호화 용 키 설정 기능이 없다는 것이다. 원래의 CHAP이 적용되는 구간이 이용자 단말기와 NAS 간의 "유선" 가입자 회선이며, 더구나 이 접속 환경은 바로 ISP의 공중 서비스 환경인 점을 감안하면, 유선 회선에 대한 도청 공격은 그리 심각하지 않다고 볼 수 있다. 이러한 이유로, CHAP 프로토콜은 암호화 용 키 생성 기능은 없고 순수 인증 기능만을 가지는 프로토콜로 설계되었다. WLAN과 같은 무선접속 환경이 유선보다 도청에 취약하다는 점을 생각하면, 이 키 설정 기능의 결여는 적지 않은 단점이 될 수도 있다. 그러나, 키 설정 기능을 추가하는 것이 그리 어렵지는 않다. 일례로서, CHAP 프로토콜의 Microsoft 버전인 MS-CHAP 프로토콜 환경에서 암호화용 키를 생성하는 방법이 IETF 드래프트로 제안된 적이 있다 [6].

그러나, CHAP 프로토콜이 무선 환경에 부적합하다고 볼 수 있는 가장 큰 이유는, 앞에서 언급한 키설정 기능 부재 때문이라기보다는, 이 프로토콜이 패스워드 추측 공격(password guessing attack)에 [7] 취약하다는 점이다. 이 공격은 "off-line dictionary attack"으로도 불리며, 대부분의 패스워드 기반 인증 프로토콜에 가해질 수 있다. 그림 4에서 보는 것처럼, 패스워드 정보는 해쉬되어 서버로 전달된다:

$$h(A, password, r_B)$$

여기서, A는 이용자의 ID이고, r_B는 서버 쪽에서 이용자에게 보낸 난수로서, 이 두 가지 값은 도청하는 공격자도 알 수 있다. 따라서 해쉬 함수 입력 중에서 유일한 비밀정보는 패스워드다. 공격자는 이 해쉬 결과와 A 그리고 r_B를 녹취해 두었다가, 패스워드 값을 추측해서 몇 번이고 그 값을 바꾸어 가며 해쉬 계산을 할 수 있다. 그 해쉬 결과가 녹취된 해쉬 값과 동일하면 추측된 패스워드가 실제 패스워드일 가능성이 매우 높다. 더구나, 이런 녹취를 두세 번하여, 처음 성공한 패스워드 값을 다른 "r_B , $h(A, password, r_B)$" 쌍에 적용했을 때도 녹취된 해쉬 값과 동일하다면 추측한 패스워드 값이 바로 진짜 패스워드 값이라고 볼 수 있을 것이다. 유선 인터넷 접속 환경이라면, 이런 녹취 자체가 그리 흔치 않겠지만, WLAN에서는 녹취의 우려가 다소 있다고 볼 수 있다. 이런 연유로, EAP-MD5는 WLAN 환경에 그리 바람직한 인증 프로토콜이라고 할 수는 없겠다.

2. LEAP (Lightweight EAP)

LEAP 프로토콜은 "EAP-Cisco Wireless"로도 알려진 Cisco 버전 EAP 용 인증/키설정 프로토콜이다 [8]. 이 프로토콜의 규격은 IETF 등의 드래프트로 공개된 적이 없어 구체적인 프로토콜 내용은 알 수 없으나, CHAP 프로토콜과 매우 유사한 패스워드 기반 challenge-response 방식이며, CHAP 프로토콜과는 달리 양방향 (상호) 인증과 세션 키 설정 기능을 가지고 있다. 또한, 이 LEAP는 단순히 인증/키설정 프로토콜만을 의미하는 것이 아니라, Cisco 나름의 (proprietary) 암호화 메커니즘까지 포함하는 것으로 보인다. 즉, 기존 WEP 암호화 메커니즘에서 WEP 키가 적용되는 방법을 약간 수정함으로써, 현재 악명 높은 WEP 암호화의 약점을 보완하고 있다. 다만, 이 LEAP라는 인증/키설정 프로토콜 겸 암호화 메커니즘을 이용하기 위해서는 WLAN 카드, AP 및 인증서버 모두가 Cisco LEAP 기능을 지원해야 한다는 제약이 따른다. 기본 EAP 프로토콜 외의 특정 EAP-XXX 프로토콜을 몰라도 되는 AP가 LEAP를 지원해야 한다는 뜻은 언뜻 모순처럼 보이나, 이것은 바로 LEAP가 암호화 메커니즘에도 영향을 미치기 때문인 것으로 보인다.

기업 내부의 인트라넷 구축을 위한 WLAN 솔루션은 핫스팟 (hot spot) 모델과는 달리, 호환성이 그리 중요하지 않으므로 암호화 강도가 높은 LEAP를 쓰는 것도 좋은 대안이 될 수 있다. 그러나, 동일 WLAN 카드로 기업 내부망의 AP와 핫스팟 AP에 다 쓸 수 있으려면 호환성을 위한 조정 작업이 어떤 형태로든 필요할

것으로 보인다.

3. EAP-TLS

TLS (Transport Layer Security) 프로토콜은 HTTP 세션 보안을 위해 개발된 SSL (Secure Socket Layer) 프로토콜이 IETF에서 표준화되어 다듬어진 표준 프로토콜이다 [9]. SSL 프로토콜은 90 년대 초반 Netscape에서 개발했으며, 그 3.0 버전이 TLS 프로토콜의 출발점이 된다. SSL/TLS 프로토콜은 인증/키설정 두 가지 기능을 모두 가지고 있으며, 공개키 인증서 기반의 프로토콜이다. 다음은 SSL/TLS 프로토콜의 메시지 교환을 보여 주는 그림으로서, Paulson이 관련 표준을 분석하여 도출한 것을 [10] 약간 수정한 것이다.

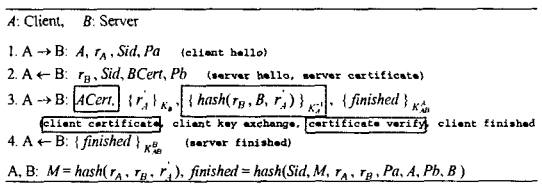


그림 5. TLS handshake protocol
Fig. 5. TLS handshake protocol

여기서, r_A와 r_B는 TLS 규격에서 각각 "client random"과 "server random"으로 불리는 난수며, r_A'은 TLS 규격에서 pre-master-secret(PMS)에 해당되는 난수로서 클라이언트가 서버 쪽으로 보내는 챌린지 데이터 역할을 한다. *ACert* 및 *BCert*는 각각 *A*와 *B*의 공개키 인증서를 말하며, Sid는 세션 식별자를 뜻한다. $\{\bullet\}_{K_B}$ 및 $\{\bullet\}_{K_A^{-1}}$는 각각 B의 공개키(K_B)를 이용한 암호화와 *A*의 개인서명키(K_A^{-1})를 이용한 메시지 서명을 표기한다. r_A, r_B, 그리고 M은 클라이언트 *A*와 서버 *B*가 세션 키 K_{AB}^A 및 K_{AB}^B를 생성하는 데 이용된다. 이 두 세션 키는 각각 *A*, *B*의 암호화 키, 즉 *B*, *A*의 복호화 키로 이용된다. 위 메시지 중에서 *Pa*(*Pb*)는 암호화/압축 방법에 관련된 *A*(*B*)의 선호(preference)를 나타낸다.

위 메시지 교환에서 점선으로 네모를 친 부분은 SSL/TLS에서 (EAP-TLS에서도) 선택적으로 이용된다. 그 의미는, 클라이언트에 의한 서버의 인증은 필수지만 그 반대 경우는 필수가 아니라는 것이다. 그 이유는, SSL 프로토콜의 주 이용분야인 인터넷 보안에서, 서버의 진위가 이용자의 진위보다 더 중요하기 때문이기도 하며, 동시에 이용자 레벨의 PKI가 구축되는 것이 서버 레벨의 PKI보다 더 어렵기 때문이다. 따라서 실제 대부분의 SSL 프로토콜 운용에서는 이용자의 공개키 인증서 대신 패스워드를 이용하여 이용자 인증을 하게 된다.

TLS 등의 공개키 방식 프로토콜의 이점은 여러 가지가 있으나, 패스워드 추측 공격으로부터 자유롭다는 점이 큰 장점이다. 패스워드 대신 인증서를 이용해서 이용자를 인증할 경우는 말할 것도 없으며, TLS와 패스워드의 혼합운용에서도 TLS 암호화 채널에 의해 패스워드가 암호화되어 전달되므로, EAP-MD5 부분에서 설명한 것과 같은 공격이 불가능하다.

이론적으로는 TLS와 같은 공개키 방식 프로토콜이 비밀키 방식 프로토콜보다 키 관리가 쉽고, 불특정 통신 주체들간의 인증/키설정이 쉽게 지원되는 매력이 있으나, 실제 구현상의 제약점 때문에 여전히 많은 인증 프로토콜이 비밀키 방식 특히 패스워드 방식을 이용하고 있다. 이용자의 공개키 인증서를 위한 PKI 구현에는 적잖은 관련 인프라 구축과 호환성 문제가 여전히 남아 있다. 이런 이유로, 기존 패스워드 프로토콜과 TLS를 결합한 EAP-TTLS (Tunneled TLS) 프로토콜 등이 제안되었다 [11].

4. EAP-SRP

SRP (Secure Remote Password) 프로토콜은 스탠포드 대학교의 Wu가 제안한 패스워드 기반 인증/키설정 프로토콜로서 [13], 2001년 IETF에 EAP 용 인증 프로토콜로 제안되었다 [14]. SRP 프로토콜은, 1992년 AT&T의 Bellovin과 Meritt가 제안한 EKE (Encrypted Key Exchange) 프로토콜로부터 [15] 시작된 소위 "strong password authentication"의 계보를 잇고 있다. EKE 프로토콜의 기본 핵심 개념은, 난수를 이용한 비밀키와는 달리 외우기 쉽고 길이가 비교적 짧다는 패스워드의 특징이자 약점을 보완하기 위하여 공개키 암호 기법을 이용한다는 것이다. 그 구현 방법의 기본 개념은, 이용자가 임시 공개키(ephemeral or short-term uncertified public key)를, 이용자 패스워드를 암호화 키로 이용해서 비밀키 방식으로 암호화하여 서버로 전달하고, 서버는 다시 이 임시 공개키를 암호화 키로 이용해서 세

션 키를 공개키 방식으로 암호화하여 그 결과를 다시 이용자 패스워드를 이용한 비밀키 방식 암호화를 거쳐서 이용자에게 전달한다는 것이다 [15]. 결국, 패스워드를 공격하려면 공개키 암호화를 공격해야 하는 문제로 귀결된다. 따라서, EAP-MD5 프로토콜 부분에서 설명한 패스워드 추측 공격이 통하지 않게 되는 것이다. 이 프로토콜은 미국 특허로 등록되었으며 [16], 이후 많은 유사 프로토콜이 제안되었다. EAP-SRP 프로토콜의 바로 그 SRP 프로토콜도 이러한 종류에 속한다. 특히, 현재 Phoenix Technologies 사의 Jablon이 제안한 B-SPEKE는 [17, 18] 매우 정교하게 다듬어진 프로토콜로 보이며, 이 프로토콜에서 제안된 핵심 기법들이 SRP에서도 이용된 것으로 볼 수 있다 (물론, SRP 제안자인 Wu는 이 사실을 인정하지 않으려 하겠지만). 이것이 현재 SRP 프로토콜을 둘러 싼 지적재산권 문제를 불러 오게 된 것으로 보인다 [19, 20, 21]. 다음 표에 유사 프로토콜들을 두 가지 주요 특징에 따라 분류하였다.

표 1. SRP 관련 프로토콜의 분류
Table 1. Protocols related to SRP

	Plaintext password stored in server	Password-verifier (not password itself) stored
forward-secrecy provided	EKE, DH-EKE, SPEKE	B-SPEKE , SRP
forward-secrecy *not* provided		A-EKE

패스워드를 그냥 평문 형태로 저장하면, 서버가 공격 당하여 패스워드 저장 데이터베이스가 노출될 경우, 공격자가 이용자의 신원을 가장한 공격이 가능해진다. 이를 막기 위한 방법으로서, 패스워드 대신 패스워드로부터 유도된 값, 즉 "password-verifier"를 저장해둘 수 있다. 보통 여기에는 해쉬 함수가 주로 쓰인다. 즉, h(password) 형태의 verifier를 쓰게 된다. 그런데, 이 password-verifier를 만드는 데, 해쉬 함수를 쓰지 않고, 공개키 암호화 기법을 쓰는 것이 B-SPEKE에서 제안되었고, 직후 SRP에서도 이용되었다. 특히 이 부분은, B-SPEKE 관련 지적재산권을 가진 Phoenix Technologies 사의 관련 특허를 [18] 면밀히 검토해야 정확한 판단을 내릴 수 있겠지만, SRP 프로토콜을 사용하려 할 경우 분쟁을 피하기 어려울 것으로 판단된다.

위 표에서 이용된 또 한 가지 분류 기준은 "forward secrecy"라는 암호학적 성질이다. 어떤 인증/키설정 프로토콜이 이 암호학적 성질을 가진다는 것은, 인증 관련 비밀키를 - 패스워드나 공개키 방식 개인키 등 - 공격자가 알아 내는 일이 있더라도, 그 키를 이용한 세션의 암호키, 즉 세션 키를 알아 낼 수는 없다는 것을 말한다 [22]. 결국, 그 세션 키로 암호화된 이용자 데이터가 노출될 우려가 없다. 이 forward secrecy는 IPSec 등에서도 중요한 암호학적 성질로서 취급되고 있다 [23]. 결론적으로, SRP 프로토콜은 두 가지 중요한 성질을 모두 만족시키는 우수한 프로토콜임에는 틀림 없다고 할 수 있겠다.

이제, SRP 프로토콜을 좀 더 자세히 분석하여 SRP 프로토콜과 Lucent 사의 지적재산권과의 마찰이 무엇 때문에 발생하는지 살펴 보도록 하자. 먼저 A와 B를 각각 이용자와 서버로 두자. 이용자 A는 자신의 패스워드를 결정한 다음, 임의의 난수 s를 고른다. 그리고 다음 계산을 수행한다.

$$x = h(s, password)$$
$$v = g^x$$

서버 B는 (v, s)를 A의 password verifier로서 저장해 둔다. 여기서, v의 계산에 쓰인 멱승계산은 $GF(p)$에서 정의되며 g는 이 필드(field)의 generator이다. 결국, (x, g^x)는 CA(Certificate Authority)에 의해 공인되지 않았다는 점을 빼면 바로 A의 개인키-공개키 쌍이다. 즉, Diffie-Hellman 프로토콜에서의 비공인 (uncertified) 개인키-공개키 쌍과 다를 바 없다. 다만 Diffie-Hellman 환경에서는 이 쌍이 해당 세션 동안만 지속되는 반면, SRP 환경에서는 반영구적으로 지속된다는 점이 다르다. 이제, 이런 준비 단계를 마치고 나면, A와 B는 다음과 같은 메시지 교환을 통하여 인증과 키설정을 할 수 있게 된다.

A: Client, B: Server
1. A → B: A, g^{r_A}
2. A ← B: s, $v + g^{r_B}$, r_B
3. A → B: $h(g^{r_A}, v + g^{r_B}, K)$
4. A ← B: $h(g^{r_A}, 3^{rd}\ \text{message}, K)$
Session key, K for A: $K = h((v + g^{r_B}) - g^x)^{r_A + x r_B}$
Session key, K for B: $K = h((g^{r_A} v^{r_B})^{r_B})$

그림 6. SRP 프로토콜
Fig. 6. SRP protocol

매우 복잡해 보이는 이 메시지들의 의미는 사실 그리 어려운 것이 아니다. g^{r_A}는 일단 A가 B에게 보내는 챌린지 값으로 볼 수 있으며, 반대로 g^{r_B}와 r_B'은 B가 A에게 보내는 챌린지라고 생각할 수 있다. 세션 키 K를 계산하기 위해서, A는 패스워드로부터 정확히 계산된 x 값을 적용해야 하며, B는 password-verifier 값인 v를 적용해야 한다. 즉, A는 패스워드를 알고 있다는 것을 B에게 증명해야 하고, B는 password-verifier를 알고 있다는 것을 A에게 증명해야 한다는 것을 알 수 있다.

이 프로토콜에서 한 가지 매우 흥미로운 점은, 두 번째 메시지에 있는

$$v + g^{r_B}$$

이다. 이것은 명백히 g^{r_B}를 패스워드로 암호화한 모양에 해당된다. 즉,

password { BPubKeyX }

의 모양이다. 실제로, A는 세션 키 K를 계산하기 위해 이 데이터로부터 g^{r_B}를 추출, 즉 복호화하고 있다는 점을 주목하기 바란다. 이러한 형태는 앞에서 소개한 EKE 프로토콜의 계보에 속하는 DH-EKE 프로토콜의 두 번째 메시지의 형태와 정확히 일치한다. 이것이 바로 SRP 프로토콜에 대하여 Lucent 사가 지적재산권을 주장하는 근거가 되는 것으로 추정된다. 참고로, DH-EKE는 AT&T의 Bellovin 등이 제안한 프로토콜이며 이 프로토콜은 특허로 등록되어 있다 [15]. 결국, SRP 프로토콜은 그 제안자인 Wu의 주장과는 달리, DH-EKE과 SPEKE 두 프로토콜에서 핵심적인 개념을 빌려 왔다고 볼 수 밖에 없는 것이다. 한 가지 흥미로운 점은, Jablon이 SPEKE 프로토콜을 개발하게 된 동기가 바로 EKE 프로토콜의 지적재산권을 피해 가기 위한 것이었다는 점이다.

5. EAP-TTLS 및 PEAP

EAP-TLS 부분에서 언급한 것처럼, TLS 프로토콜에서 이용자 인증은 선택사항이며, 실제 대부분의 HTTP 환경에서 서버 인증과 세션 키 설정은 TLS로 하고, 설정된 세션 키를 이용한 암호화 채널을 통하여 이용자 패스워드를 인증하고 있다. EAP-TTLS와 PEAP는 바로 이런 TLS 운용의 한 예라고 할 수 있다. 이 두 프로토콜은 "EAP - TLS - Any password protocol"처럼 표현될 수 있다. 즉 EAP 멀티콘센트에 TLS 콘센트를 꼽고 여기에 다시 다른 인증 프로토콜을 꽂아서 쓰는 방법으로 볼 수 있다.

EAP-TTLS는, 2002년 초, Funk Software와 Certicom이 공동으로 제안하였으며, 현재 IETF PPP Extensions Group에서 드래프트 작업 중이지만 [11], 이미 많은 상용제품이 이 프로토콜을 지원하고 있으며, 국내 NESPOT 서비스에도 일부 도입되어 있는 것으로 알려져 있다.

이 프로토콜은 앞에서 언급한 것처럼, 서버의 공개키 인증서만 필요할 뿐, WLAN 이용자의 공개키 인증서는 쓸 필요가 없다. 대신, 기존 패스워드를 이용하여 이용자를 인증할 수 있다는 것이 장점이다. 따라서 CHAP, PAP, OTP 등의 기존 패스워드 기반 인증 프로토콜을 EAP-TTLS와 결합하여 쓸 수 있다. 심지어 EAP 프로토콜 자체도 EAP-TTLS와 결합될 수 있다. 이처럼, EAP-TTLS와 결합되는 프로토콜을 EAP-TTLS 규격에서는 "tunneled authentication protocol"이라고 부른다. 즉, TLS 수행 결과로 형성된 암호화 채널, 즉 TLS 터널 안에 보호된 채로 수행되는 프로토콜이라는 뜻이다. 다음 그림 7에 EAP-TTLS 프로토콜의 기본 개념을 나타내었다.

결국, EAP-TTLS도 넓은 의미에서는 앞에서 살펴본 SRP 등의 프로토콜처럼 패스워드 기반 프로토콜과 공개키 기법의 결합으로 볼 수 있다. 다만, 그 구현에 이용된 공개키 기법이, 완성된 형태의 TLS 프로토콜이라는 것이 차이점이다. 따라서, 이 EAP-TTLS 프로토콜도 패스워드 추측 공격으로부터 안전하다는 것이 큰 장점이다.

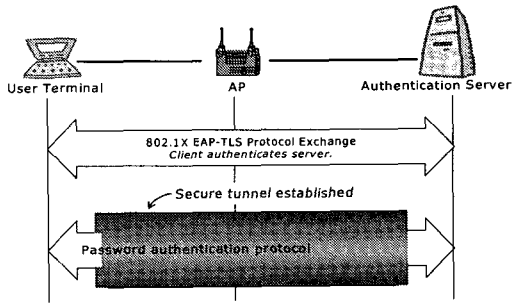


그림 7. EAP-TTLS 프로토콜 개념
Fig. 7. Basic concept of EAP-TTLS protocol

Microsoft와 Cisco가 제안한 PEAP (Protected EAP) 프로토콜은 [24], 그 구조가 EAP - TLS - EAP - "Any authentication protocol"로 되어 있어, EAP-TTLS보다 "encapsulation level"이 한 단계 더 추가된다. 즉, EAP-TTLS는 바로 구체적인 패스워드 인증/키설정 프로토콜이 바로 터널링되지만, PEAP는 또 다시 EAP라는 매개체를 통해서만 임의의 인증/키설정 프로토콜이 터널링된다.

이 두 프로토콜의 단점은, 프로토콜 구조가 복잡한 만큼 (EAP - TLS - XXX) 신호 교환 및 처리 부하가 적지 않다는 점이다. 또한, 서버의 공개키 인증서를 이용하는 TLS를 이용하는 만큼 공개키 연산에 따르는 부하도 다른 패스워드 기반 프로토콜보다는 더 무거울 것이다. EAP-SRP 프로토콜 역시 공개키 암호화 기법을 이용하긴 하지만, 여기에 쓰이는 공개키는 모두 임시 (ephemeral) 공개키이므로 관련 부하가 상대적으로 적다고 할 수 있다. 그럼에도 불구하고, 이미 잘 다듬어진 기존 인증 프로토콜을 활용할 수 있다는 점이 두 프로토콜의 장점이다.

6. EAP-SKE

EAP-SKE (Shared Key Exchange) 프로토콜은 [25] 패스워드 방식도 공개키 증명서 (public-key certificate) 방식도 아닌 비밀 난수키 방식이다. 그러나 본질적으로는 패스워드 방식과 함께 비밀 공유 키 (shred secret key) 방식이라고 볼 수 있다. 다만 그 공유 키가 이용자의 머리 속에 저장된다는 점이 차이점이다. 그러나, 현실적으로는 흔히 "공유 키" 방식이라고 하면 보통 단말기 내부에 안전하게 저장된 비밀 공유 키를 말한다. 이러한 비밀 난수 키 방식은, 엄밀하게 보자면 "이용자 인증"이라기보다는 "단말기 인증"으로 볼 수 있다. 이 방식의 차이점은, 일단 보통의 패스워드 프로토콜이 안고 있는 "offline dictionary attack" 관련 취약성이 없으며, 그러면서도 관련 계산/신호 부하가 EAP-SRP나 EAP-TTLS/PEAP 등에 비해 훨씬 더 작다는 장점이 있다. EAP 용으로 제안된 공유 키 방식 프로토콜 중 하나가 EAP-SKE 프로토콜이다. 공유 키 방식과 패스워드 방식은 일방적으로 어느 하나를 선택하려는 관점에서 볼 것이 아니라, 앞으로 WLAN 단말기가 더욱 다양해지는 환경을 감안하여 공유 키 방식 프로토콜을 패스워드 방식 프로토콜과 서로 보완적으로 운영하는 것이 바람직할 것으로 보인다. 예를 들면:

- PDA처럼 철저히 개인화된 단말기에 대해서는 이용자 인증이냐 단말기 인증이냐 하는 구분이 사실 무의미하며,
- 더욱이 패스워드 타이핑이 노트북 단말기보다 불편하다는 점, 그리고
- 단말기 계산 성능도 노트북에 비해 떨어진다는 점을 감안하면,

EAP-SKE는 매우 유력한 대안이 될 수 있다. 사실 EAP 인증 체계의 가장 큰 장점 중의 하나가 여러 가지 인증 방법을 함께 운용할 수 있다는 것이다.

Ⅳ. 결 론

지금까지 EAP 환경에서 쓸 수 있는 여러 가지 인증/키설정 프로토콜들을 살펴 보았다. 이들은 다음과 같이 몇 가지 기준으로 나누어 볼 수 있다.

키설정 기능의 유무
- 순수 인증 프로토콜: EAP-MD5
- 인증 및 키설정 프로토콜: EAP-MD5를 제외한 나머지 EAP-XXX 프로토콜

이용자 인증 관련 비밀정보
- 공개키 증명서: EAP-TLS
- 패스워드: EAP-TLS와 EAP-SKE를 제외한 나머지 EAP-XXX 프로토콜
- 비밀 난수 키: EAP-SKE

패스워드 추측 공격에 대한 면역성
- 취약함: EAP-MD5, LEAP
- 취약하지 않음: EAP-MD5를 제외한 나머지 EAP-XXX 프로토콜 (단, 이 보고서에서 다룬 프로토콜 범위 내에서)

양방향 인증 (이용자 인증 + 서버 인증)
- 이용자 인증만 하는 경우: EAP-MD5
- 서버 인증도 지원하는 경우: EAP-MD5를 제외한

나머지 EAP-XXX 프로토콜

신호 및 계산 부하 정도

- 저: EAP-MD5, LEAP, EAP-SKE
- 중: EAP-SRP
- 고: EAP-TLS, PEAP/EAP-TTLS

어떤 프로토콜을 선택할 것인가는 운용환경, 기존 인증체계, 지적재산권 문제, 구현 비용, 운용 부담 등 많은 것을 고려해야 하는 만큼, 그리 간단한 문제는 아니다.

소규모 WLAN 환경이라면 굳이 패스워드 인증을 고집할 필요 없이, 공개키 인증서 기반 이용자 인증을 지원하는 EAP-TLS를 이용하는 것이 가장 무난해 보인다. 더구나, 이 프로토콜은 윈도우즈 운영체제에 내장되어 있다. 무선단말기의 형태가 주로 PDA인 환경이라면, 패스워드는 이용자가 직접 입력해야 하는 불편과 패스워드 추측 공격의 부담을 떠안을 필요 없이 비밀 난수 키 기반의 인증 프로토콜을 선택하는 것이 더 바람직해 보인다.

대규모 서비스 망과 패스워드 기반의 기존 인증체계를 가진 ISP 사업자들의 경우는, 패스워드 기반 인증체계를 계속 이어가는 것이 가장 현실적으로 보인다. 이런 측면에서 볼 때, 성능과 보안 두 가지 측면에서 모두 무난해 보이는 EAP-SRP 프로토콜이 선행 프로토콜과의 잠재적 지적재산권 분쟁 여지가 있다는 것은 매우 아쉬운 점이라 할 수 있다. 그러나, EAP-MD5 프로토콜도, 패스워드 추측 공격에 대한 취약성을 보완하는 방법으로 꼭 프로토콜 측면이 아니라 단말기 접속 프로그램 차원에서 구해 볼 수도 있다. 자세한 설명은 생략하겠지만, 이용자 패스워드와 제 삼의 마스터 비밀키를 해쉬 함수로 결합하고, 그 결과를 패스워드 대신 이용하는 경우, 패스워드 추측 공격은 통할 수 없게 된다. 다만, 마스터 비밀키의 보안 관리가 추가적인 부담이 될 수는 있겠다.

어쨌든, EAP 인증 프로토콜 자체가 일종의 멀티콘센트로서 여러 가지 인증 프로토콜을 동시에 삽입하여 운용할 수 있기 때문에, 단일 관리 영역에서라도 필요한 보안 강도나 기타 여건에 따라 여러 인증 프로토콜을 적절히 함께 운용하는 방향이 바람직하다고 하겠다.

참고문헌

[1] L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," IETF RFC 2284, March 1998.

[2] IEEE Standard 802.1X, Standards for Local and Metropolitan Area Networks: Port-Based Access Control, 2001.

[3] IEEE, "LAN MAN standards of the IEEE Computer Society: wireless LAN medium access control (MAC) and physical layer(PHY) specification", IEEE Standard 802.11, 1997.

[4] B. Lloyd, et al., "PPP Authentication Protocols", IETF RFC 1992, October 1992.

[5] W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)", IETF RFC 1994, August 1996.

[6] G. Zorn, "Deriving Keys for use with Microsoft Point-to-Point (MPPE)", IETF draft, Octo-ber 2000

[7] B. Schneier, Applied Cryptography, 2nd ed. Wiley, 1996, pp. 171-173.

[8] http://www.cisco.com/warp/public/784/packet/exclusive/apr02.html

[9] B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol," IETF RFC 2716, October 1999.

[10] L.C. Paulson, "Inductive Analysis of the Internet Protocol TLS", ACM Transactions on Computer and System Security 2 3, 1999, pp. 332-351.

[11] P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)", IETF draft, July 2004.

[12] T. Wu, "The Secure Remote Password Protocol", in Proceedings of the 1998 Internet Society Network and Distributed System Security Symposium, San Diego, CA, Mar 1998, pp. 97-111.

[13] J. Carlson, B. Aboba and H. Haverinen, "EAP

SRP-SHA1 Authentication Protocol", IETF draft, July 2001.

[15] S.M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks", Proceedings of the 1992 IEEE Computer Society Confer-ence on Research in Security and Privacy, 1992, pp.72-84.

[16] S.M. Bellovin and M. Merritt, "Cryptographic Protocols for Secure Communications", U.S. Patent #5,241,599, 31 August 1993.

[17] D. Jablon, "Extended password methods immune to dictionary attack", In WETICE '97 Enterprise Security Workshop, Cambridge, MA, June 1997.

[18] D. Jablon, "Cryptographic methods for remote authentication", U.S. Patent #6,226,383, 1 May 2001.

[19] http://www.ietf.org/ietf/IPR/LUCENT-SRP

[20] http://www.ietf.org/ietf/IPR/WU-SRP

[21] http://www.ietf.org/ietf/IPR/PHOENIX-SRP-RFC2945.txt

[22] DongGook Park, et al., "Forward secrecy and its application to future mobile communications security", PKC 2000, Lecture Note in Computer Science (LNCS) 1751, Springer-Verlag, 2000.

[23] N. Doraswamy and D. Harkins, IPSec: the new security standard for the Internet, intranets, and virtual private networks, 2nd Ed., Prentice Hall, 2003.

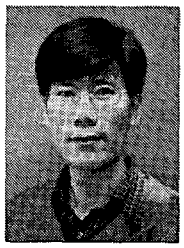
[24] H. Andersson et al., "Protected EAP protocol (PEAP)", IETF draft, 23 February 2002.

[25] L. Salgarelli, "EAP SKE authentication and key exchange protocol", IETF draft, Nov 1, 2003.

저자약력

박동국(DongGook Park)

1986년 경북대학교 전자공학과 졸업 (공학사)
1989년 KAIST 전기및전자공학과 졸업 (공학석사)
2001년 호주 QUT (Queensland University of Technology) School of Data Communications, PhD
1989년 ~ 2004년 KT 연구개발본부 선임연구원
2004년 ~ 현재 순천대학교 정보통신공학부 전임강사
※ 관심분야: 인증/키설정 프로토콜 및 그 응용



조경룡 (Kyung-Ryong Cho)

1987년 숭실대학교 전자공학과 졸업 (공학사)
1989년 숭실대학교 전자공학과 졸업 (공학석사)
1995년 숭실대학교 전자공학과 졸업 (공학박사)
1989년 ~ 1990년 한국증권전산(주) 통신시스템부 사원
1990년 ~ 1996년 SK텔레콤 중앙연구원 선임연구원
1996년 ~ 현재 순천대학교 정보통신공학부 부교수
※ 관심분야: 채널코딩, 디지털변복조, 이동통신