情報保護學會論文誌
第18卷 第5號, 2008. 10

IEEE 802.11 환경에서 빠른 핸드오프를 제공하는 그룹키 기반의 인증 프로토콜*

이 창 용[1†], 김 상 진[2‡], 오 희 국[1†], 박 춘 식[3]
[1]한양대학교, [2]한국기술교육대학교, [3]한국전자통신연구원 부설연구소

A Group Key based Authentication Protocol Providing Fast Handoff in IEEE 802.11*

Changyong Lee[1†], Sangjin Kim[2‡], Heekuck Oh[1†], Choonsik Park[3]
[1]Hanyang University, [2]Korea University of Technology and Education, [3]The Attached Institute of ETRI

요 약

IEEE 802.11 기반 무선랜에서 실시간으로 대용량의 데이터 전송이 필요한 멀티미디어 서비스를 원활히 제공하기 위해서는 빠른 핸드오프가 필수적이다. 핸드오프 지연을 줄이기 위해서는 인증 지연을 최소화하는 것이 필요하며, 이를 위해 Mishra 등은 예측을 통해 표준 기법의 인증 지연을 개선하였으며, 박찬일 등은 Blom의 키 사전분배 기법을 이용하여 인증 지연을 개선하고자 하였다. 이 논문에서는 Bresson 등의 그룹키 프로토콜을 활용한 개선된 형태의 인증 프로토콜을 제안한다. 제안하는 프로토콜의 경우 네트워크에 참여한 경험이 있는 이동 노드는 해시 연산만을 사용하여 액세스 포인트와 빠르게 상호인증을 수행할 수 있어 박찬일 등의 기법보다 효율적이며, Mishra 등의 예측 기법까지 적용하면 매우 빠른 상호인증을 제공할 수 있다.

ABSTRACT

Reducing handoff latency is essential in providing seamless multimedia service in Wireless LAN based on the IEEE 802.11 standard. Reducing authentication delay is critical in reducing handoff latency. To this end, several authentication protocols for fast handoff have been proposed. Mishra et al. used proactive key distribution to improve the authentication delay incurred in the current standard and Park et al. proposed a new authentication protocol based on Blom's key pre-distribution scheme. In this paper, we propose an enhanced authentication protocol based on Bresson et al.'s group key protocol. If a mobile node has previously access the network, our proposed protocol only requires simple hash operations in providing mutual authentication between a mobile node and access points. Our protocol is more efficient than Park et al.'s and Mishra et al.'s technique can be used in our protocol to further enhance our protocol.

Keywords : IEEE 802.11, handoff, authentication.

Ⅰ. 서 론

통신기술의 발달과 함께 사용자는 기존의 유선 통신에서 벗어나 언제 어디서든 서비스를 제공받을 수 있는 무선 통신을 요구하게 되었고 제공되는 서비스의 종류도 기본적인 텍스트 기반의 서비스에서 대용량의 실시간 데이터 전송을 필요로 하는 멀티미디어 서비스로 변화하고 있다. 이에 따라 현재 IEEE 802.11 기반의 무선랜 시스템이 빠른 속도와 낮은 가격, 사용자의 이동성 지원이란 장점을 기반으로 초고속 무선 인터넷의 기반 구조로 자리 잡게 되었다.

이러한 무선랜의 보편화와 함께 사용자의 안전한 통신을 위한 무선랜 보안 표준 프로토콜인 IEEE 802.11i가 제정되었다[1]. 이 표준은 공개키 기반의 상호 인증 프로토콜인 EAP-TLS(Extensible Authentication Protocol-Transport Layer Security)[2]를 인증 표준으로 채택하고 있다. 하지만 실시간 데이터 전송을 요구하는 멀티미디어 서비스에서 빠른 핸드오프는 필수 요구조건인데 반해, 이 프로토콜의 수행에는 평균 1초가량의 수행시간이 소요되어 빠른 핸드오프를 지원하지 못하는 문제점을 가지고 있다.

이에 따라 핸드오프 과정에서 인증 시간을 단축하기 위한 많은 연구가 진행되었고 관련 프로토콜이 제안된 바 있다. Mishra 등은 새로운 인증 프로토콜을 제시하는 대신에 기존 인증 프로토콜의 인증 지연을 개선할 수 있는 기법을 제안하였다[3]. 그들은 이웃 그래프(neighbor graph)를 이용하여 이동 노드(MN, Mobile Node)가 이동할 수 있는 액세스 포인트(AP, Access Point)를 예측하고 사전에 AP로 키를 분배하여 실제 핸드오프 과정에서 필요한 인증 지연을 개선하였다. 이들의 기법은 모든 인증 프로토콜에 활용 가능한 기법이다. 박찬일 등은 Blom의 키 분배 기법[4]을 이용한 사전인증 기법을 제안하였다[5]. Blom의 기법은 N명의 사용자가 있을 때 적은 정보를 각 사용자에게 사전에 배포하여 각 사용자 쌍마다 독특한 대칭키를 공유할 수 있도록 해주는 기법이다. 박찬일 등은 하나의 이동 노드와 모든 AP를 N명의 사용자로 모델링하여 이동 노드와 AP에게 적은 정보를 주어 독특한 대칭키를 공유할 수 있도록 하고 있으나 실제 핸드오프 과정에서 교환되어야 하는 정보는 비교적 많으며, 핸드오프가 이루어질 때마다 행렬 곱셈 연산이 필요하다. 본 논문에서는 그룹키 기반의 키분배 방식을 통해 빠른 핸드오프를 지원하는 인증 프로토콜을 제안한다. 제안하는 프로토콜은 저전력 기기를 고려한 Bresson 등의 그룹키 프로토콜[6]을 사용하여 적은 연산으로 이동 노드와 모든 AP간에 그룹키를 확립하고 확립된 키를 이용하여 이동 노드와 AP가 빠르게 상호인증을 수행한다. 제안된 프로토콜의 경우 네트워크에 한번 접근한 경험이 있는 이동 노드는 공개키 연산 없이 해시 연산만 사용하여 빠르게 AP와 상호 인증을 수행할 수 있으며, Mishra 등의 예측 기법을 추가로 적용하면 핸드오프할 때 공개키 연산없이 빠르게 상호 인증이 가능하다.

이 논문의 구성은 다음과 같다. 2장에서는 제안하는 프로토콜의 연구 배경과 기존에 제안된 빠른 핸드오프를 지원하는 IEEE 802.11 상의 인증 프로토콜들을 소개한다. 3장에서는 제안하는 프로토콜에 대해 자세히 기술하고, 4장에서는 제안하는 프로토콜의 안전성과 효율성을 분석한다. 끝으로 5장에서는 결론과 향후 과제를 제시한다.

Ⅱ. 연구 배경

2.1 표기법

이 논문에서는 [표 1]에 기술된 표기법을 사용하여 관련연구와 제안된 프로토콜을 설명한다.

2.2 IEEE 802.11

무선 통신의 보급과 함께 IEEE 802.11 기반의 무선랜의 사용이 보편화되었다. IEEE 802.11은 가격이 저렴하고 전송 속도가 빠르며 사용자에 대한 이동성을 지원하여 널리 사용되고 있다. IEEE는 자체적으로 IEEE 802.11i 표준를 통해 무선랜 환경에 대한 인증과 보안을 제공하고 있다[1]. 본 논문에서는 IEEE 802.11i 환

접수일 : 2008년 2월 28일; 수정일 : 2008년 5월 26일;
채택일 : 2008년 7월 3일
* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터(홈네트워크연구센터) 육성 · 지원사업의 연구결과로 수행되었음
* 이 논문은 2008년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2006-000-10957-0)."
† 주저자, chylee@kisa.or.kr
‡ 교신저자, sangjin@kut.ac.kr

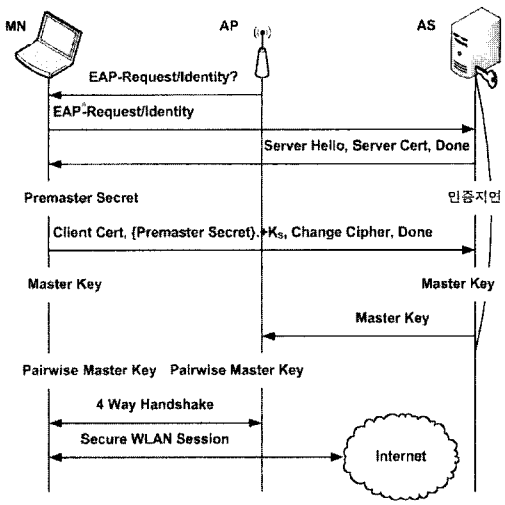
〔표 1〕 표기법

표기	의미
M	이동노드
S	인증 서버
A	액서스 포인트
q	충분히 큰 소수
Z_q^*	q를 법으로 하는 곱셈군
x_i	사용자 i의 개인키
$y_i = g^{x_i}$	사용자 i의 공개키
K	그룹키
K_i	그룹키를 생성하기 위한 시드(seed)
N	난스값
$H(\alpha)$	메시지 α에 대한 해쉬값
$Sig(\alpha)$	메시지 α에 대한 서명값
$\{\alpha\}_K$	대칭키 K를 이용한 α에 대한 암호화

경에서의 통신을 가정하였다. 이 환경에서는 이동 노드, 인증 서버(AS, Authentication Server), 액세스 포인트가 개체로 참여한다. MN이 통신 서비스를 받고자 할 때 AP와 MN은 AS를 이용하여 상호인증을 수행한다. AP와 AS간 연결은 비제어 포트(uncontrolled port)로 회선의 개폐스위치 없이 항상 연결되어 있다. 이 연결은 AP가 임의대로 통제할 수 없는 연결이며 AS와 AP 간 대칭키 공유 등으로 안전한 채널로 간주된다. AP와 네트워크 간 연결은 제어 포트(controlled port)로 AP에 의해 개폐가 가능한 연결이다. AP는 AS에 의해 허용된 사용자에 대해서만 네트워크로 접근을 허용한다.

IEEE 802.11 기반의 무선랜 서비스의 보안 요소에는 사용자 인증(authentication), 접근 제어(access control), 권한 검증(authorization), 데이터 기밀성(confidentiality), 데이터 무결성(integrity) 등이 있으며 이러한 보안 요소가 전체적으로 만족되었을 때 안전한 무선 통신이 이루어질 수 있다[7]. 본 논문에서는 사용자 인증 방식에 대해 다루고자 한다.

AP와 MN의 상호인증은 정당하지 않은 사용자의 네트워크로의 접근을 차단하고, 사용자의 정보가 거짓 AP로 전달되는 것을 막는 역할을 한다. IEEE 802.11의 보안 표준인 IEEE 802.11i에서는 AP와 MN의 상호인증을 위해 EAP-TLS 프로토콜을 표준으로 채택하고 있다. EAP-TLS 프로토콜은 [그림 1]에 기술되어 있으며, 크게 인증 단계와 세션키 확립 단계로 구성되어 있다. 인증 단계에서 AS와 MN은 자신의 공개키 인증서를 서



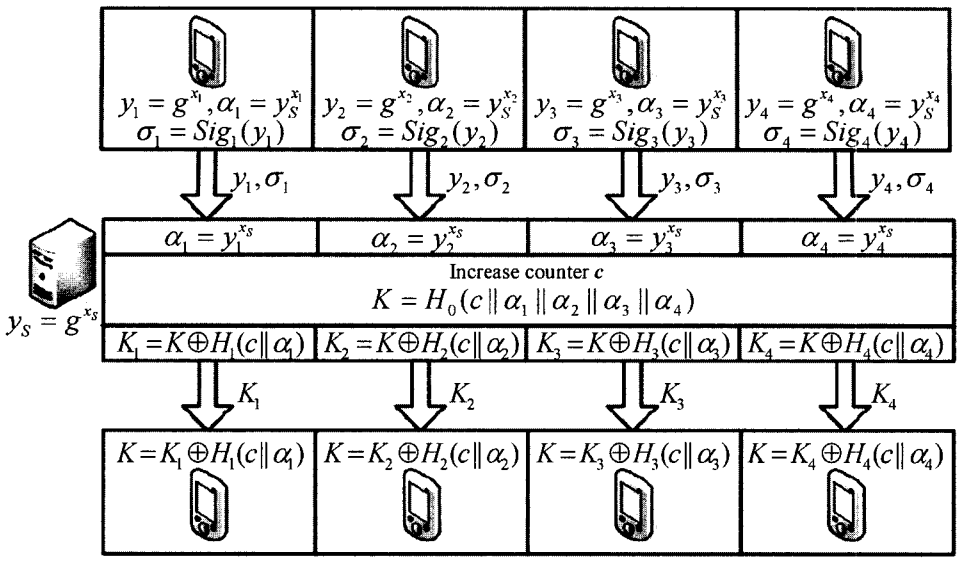
〔그림 1〕 EAP-TLS 프로토콜

로 교환하여 서로를 인증하며, 이 과정에서 PMK (Pairwise Master Key)를 성립한다. 세션키 확립 단계에서는 PMK를 이용한 4 단계 핸드쉐이크(4-way handshake)가 수행되며, 이를 통해 PTK(Pairwise Transient Key)를 확립한다. 모든 과정이 성공적으로 완료되면 MN은 AP와 확립된 PTK를 이용하여 안전한 채널로 통신이 가능하다. 하지만 이 프로토콜의 경우 하나의 MN과 AS가 상호 인증을 수행 하는데 적게는 750ms부터 많게는 1200ms가 소요된다[5]. 이는 상당히 짧은 시간처럼 보일 수 있으나 빠른 핸드오프를 지원하기에 적합하지 않으며 무선랜을 통해 VoIP(Voice over Internet Protocol), IPTV(Internet Protocol TV)와 같은 음성을 포함한 실시간 멀티미디어 서비스의 품질을 보장하기 어렵다.

2.3 Bresson 등의 그룹키 프로토콜

본 절에서는 제안하는 프로토콜에서 사용되는 그룹키 기법에 대해 알아본다. Bresson 등은 저전력 환경을 고려하여 지수연산 없이 적은 연산량으로 그룹키를 생성하는 그룹키 동의 프로토콜을 제안하였다[6]. 이 프로토콜은 중앙 서버를 필요로 하는 중앙 집중형 방식으로 중앙 서버가 그룹키를 생성하고 각 사용자는 중앙 서버로부터 시드를 받아 그룹키를 계산하는 방식으로 동작한다. 전체적인 프로토콜 동작은 [그림 2]에 도식화 하였다.

중앙 서버의 그룹키 생성부분에서의 프로토콜 동작



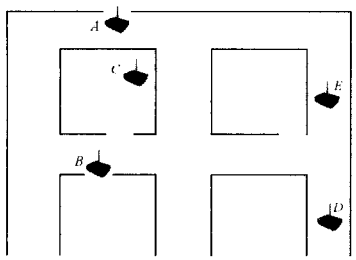
〔그림 2〕 Bresson 등의 그룹키 프로토콜

은 다음과 같다. 각 사용자 i는 개인키 $x_i \in Z_q^*$를 선택하고 공개키 $y_i = g^{x_i}$를 계산한다. 이때 중앙 서버는 개인키 $x_S \in Z_q^*$를 선택하고 공개키 $y_S = g^{x_S}$를 계산하여 각 사용자에게 배포한다. 각 사용자는 중앙 서버의 공개키를 자신의 개인키로 지수승한 $\alpha_i = y_S^{x_i}$를 계산하고 자신의 공개키 y_i를 인증하기 위한 서명값 $\sigma_i = Sig(y_i)$와 함께 중앙 서버에 전달한다. 중앙 서버는 각 사용자로부터 받은 공개키를 자신의 개인키 x_S로 지수승한 $\alpha_i = y_i^{x_S}$값을 계산한다. 이때 사용자가 계산한 $y_S^{x_i}$ 값과 중앙 서버가 계산한 $y_i^{x_S}$ 값은 같은 값을 가지므로 중앙 서버와 사용자는 대칭키를 공유하게 된다. 중앙서버는 모든 사용자로부터 받는 α_i값과 카운터 c를 비트결합하고 해쉬하여 그룹키 $K = H(c \| \alpha_1 \| \alpha_2 \| \ldots \| \alpha_n)$을 계산한다. 카운터 c는 그룹키가 업데이트 될 때마다 1씩 증가되는 카운터이다. 일단 그룹키가 생성되면 중앙 서버는 각 사용자에게 $K_i = K \oplus H(c \| \alpha_i)$ 값을 전달한다. K_i값은 그룹키 생성을 위한 시드가 되는 값으로 각 사용자는 자신이 가지고 있는 α_i를 이용하여 그룹키 $K = K_i \oplus H(c \| \alpha_i)$를 계산해낼 수 있다. 각 사용자만이 α_i를 계산할 수 있으므로 각 사용자는 그룹키를 얻을 수 있지만 그룹에 속하지 않는 사용자들은 그룹키를 계산할 수 없다.

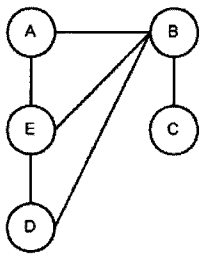
이 기법은 초기 단계에서 공개키 연산을 통해 α를 계산하지만 이 연산은 사전에 한번만 계산해두면 되고 실질적인 동작 단계에서는 매우 적은 연산량만을 요구하므로 여러 참여자들이 효율적이고 빠르게 그룹키를 성립할 수 있다.

2.4 관련 연구

Mishra 등은 이동 노드가 이동할 수 있는 다음 AP를 예측하여 미리 키를 배포함으로써 실제 핸드오프 과정에서 요구되는 인증 지연을 개선하는 기법을 제안하였다[3]. 이들 기법은 다른 인증 프로토콜에 적용이 가능한 범용적인 기법이다. Mishra 등은 이동 노드의 이동을 예측하기 위해 이웃 그래프와 PMK 트리를 이용하고 있다. 이웃 그래프란 MN이 이동할 때 물리적으로 접근 가능한 AP들을 쌍으로 묶어 만든 일종의 데이터 구조이다. [그림 3.(a)]와 같은 건물의 구조를 가정하면 [그림 3.(b)]와 같은 이웃 그래프를 만들 수 있다. 이웃 그래프를 이용하면 앞으로 MN이 접근할 AP를 예측할 수 있으므로 해당 MN과 핸드오프를 수행할 AP들에게 PMK를 사전에 배포할 수 있다. AS는 핸드오프 후보 AP를 고려하여 해당 MN이 앞으로 사용할 PMK를 순서대로 엮어서 트리 형태로 만들 수 있는데 이를 PMK 트리라 한다. MN의 최초 인증시 MN과 AS는 마스터 키(Master Key, MK)와 식 (1)를 이용하여 PMK_0를 생성하며 MN은 AP와 PMK_0를 이용하여 상호 인증한다.



(a) 무선 랜의 물리적 배치



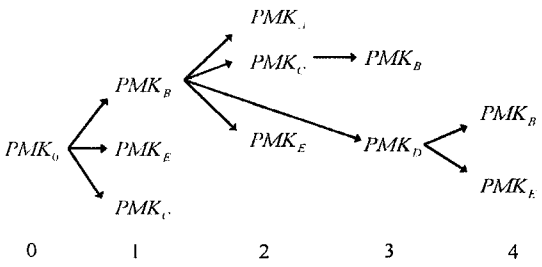
(b) 대응되는 이웃 그래프

〔그림 3〕 무선 랜의 물리적 배치와 대응되는 이웃 그래프

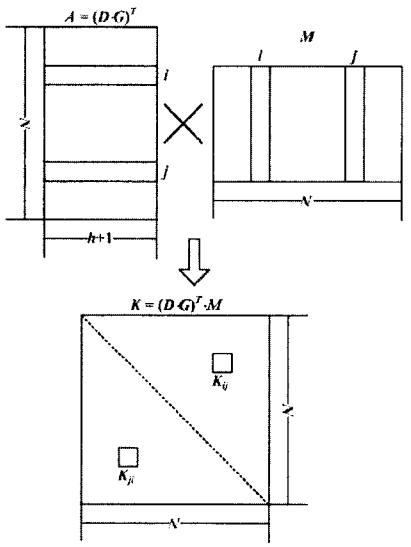
$$PMK_0 = TLS-PRF(MK, \quad client\ Hello.random \| server\ Hello.random) \tag{1}$$
$$PMK_n = TLS-PRF(MK, \quad PMK_{n-1} \| AP_{MAC} \| MN_{MAC})$$

그 이후 MN은 핸드오프 과정에서는 이전에 사용된 PMK_{n-1}, 새 AP의 MAC 주소, MN의 MAC 주소, MK를 조합하여 생성한 PMK_n을 사용하여 새 AP와 상호인증을 수행한다.

[그림 4]에 PMK 트리를 도식화 하였다. 예를 들어 MN이 AP A에 최초로 접근할 경우 AS와 MN은 EAP-TLS로 상호인증을 수행하고 AS는 ACCESS-ACCEPT 메시지에 PMK_0을 추가하여 A에 전달한다. MN과 A는 4 단계 핸드쉐이크를 통해 PMK_0를 확인하여 비밀 통신을 한다. 이때 AS는 네이버 그래프를 이용해 이웃 AP를 선정하고 이 AP들에게 PMK_1를 전달한다. MN이 PMK_1를 수신한 AP로 이동했을 경우 전체 인증과정의 수행 없이 4 단계 핸드쉐이크만을 통해 PMK_1을 가지고 있는지 여부를 확인하면 상호인증을 할 수 있다. Mishra 등의 논문에는 구체적인 언급은 없지만 핸드오프 과정에서 AS와 상호 작용이 필요 없는 경우에는 새 AP는 최소한 AS에게 MN의 접근 사실을 알려야 다음 이웃노드들에게 PMK의 분배가 가능해진다. 이 기법은 건물 내부 등 다음 AP를 충분히 예측할 수 있으며, 이웃 노드의 수가 제한적일 경우에 효과적이다.

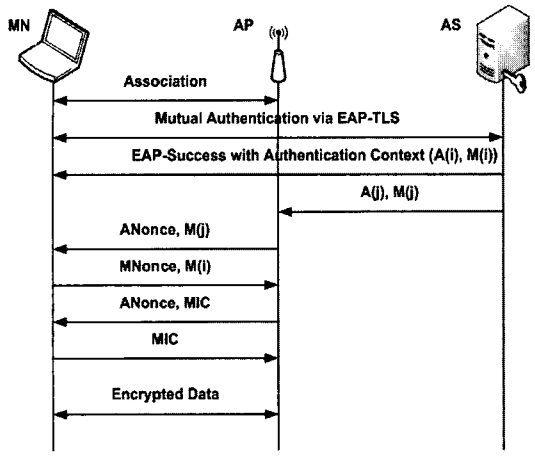


〔그림 4〕 PMK tree의 예



〔그림 5〕 Blom의 키 사전 분배 기법

박찬일 등은 무선 메쉬 네트워크에 적용 가능한 사전 인증 기법을 제안하였다[5]. 무선 메쉬 네트워크란 AP와 AS간의 연결이 유선이 아닌 무선으로 가정하는 환경으로 AP를 동적으로 배치하여 네트워크 서비스의 범위 및 환경을 변화시킬 수 있는 특성을 가지고 있다. 박찬일 등은 Blom의 키 사전 분배 기법을 응용하여 핸드오프 과정의 인증 지연을 개선하고자 하였다. Blom의 기법은 N명의 사용자가 있을 때 적은 정보를 각 사용자에게 사전에 배포하여 각 사용자 쌍마다 독특한 대칭키를 공유할 수 있도록 해주는 기법이다[4]. 이를 위해 Blom은 [그림 5]와 같은 $N \times N$ 크기의 대칭 행렬을 이용한다. 여기서 h는 시스템 파라미터로 h명 이상의 사용자가 가지고 있는 정보를 공격자가 모두 획득하면 이 공격자는 모든 사용자 쌍 간의 대칭키를 만들 수 있게 된다. 즉, h가 증가하면 시스템의 안전성이 높아지나 각 사용자가 유지해야 하는 정보는 많아진다. 각 사용자 i는 행렬 A의 i번째 행 $A(i)$를 비밀스럽게 유지해야 하



[그림 6] 박찬일 등의 사전인증 프로토콜

며, 행렬 M은 모든 사용자에게 공개되는 정보이다. 따라서 사용자 i가 다른 사용자 j와 사용하게 될 대칭키는 자신의 행 $A(i)$와 공개된 행렬 M의 $M(j)$ 열을 곱하여 생성하게 된다.

박찬일 등은 하나의 이동 노드와 모든 AP를 N명의 사용자로 모델링하고 있다. 따라서 MN이 네트워크에 최초 접속을 하게 되면 인증 서버는 새 행렬을 구성한 다음에 EAP-TLS를 수행하여 MN을 인증한 뒤 MN에게는 $A(i)$, $M(i)$를 전달하고 AP j에게는 $A(j)$, $M(j)$를 분배한다. 그 다음 핸드쉐이크를 통해 서로 $M(i)$와 $M(j)$를 교환하여 대칭키를 생성하고 상호 인증을 수행할 수 있다. [그림 6]은 박찬일 등의 프로토콜의 동작과정을 보여준다.

박찬일 등도 Mishra 등의 기법을 이용하여 이웃 노드들에게 사전에 행렬 값들을 배포하여 핸드오프 과정에서 인증 지연을 개선하고자 하였다. MN은 핸드오프 발생시 해당 AP가 행렬 값들을 가지고 있으면 전체 인증을 다시 하지 않고 4 단계 핸드쉐이크만 수행하여 해당 AP와 빠르게 상호 인증을 수행할 수 있다. 하지만 이 과정에서 새 AP k와 $M(i)$와 $M(k)$를 상호 교환해야 한다. 구체적으로는 기존의 IEEE 802.11i에 정의된 4단계 핸드쉐이크를 사용하지 않고 다음과 같이 핸드쉐이크 과정을 수행한다. 첫 단계에서 AP는 자신이 생성한 난스 값인 $ANonce$와 $M(j)$를 MN에게 전달한다. 이를 받은 MN은 자신이 생성한 난스 값인 $MNonce$와 $M(i)$를 AP에게 전달한다. $M(i)$와 $M(j)$를 교환한 MN과 AP는 각각 K_{ij}와 K_{ji}를 생성할 수 있다. MN과 AP는 K_{ij} 또는 K_{ji} 값과 교환된 난스, MN과 AP의 MAC 주소를 식 (2)에 적용하여 PTK를 계산한다. 세 번째 단계로 AP는 이전 전송했던 $ANonce$와 식 (2)를 통해 계산한 PTK를 키로 계산한 MIC값을 MN에게 전달한다.

$$PTK = PRF(K_{ij}(\text{or } K_{ji}), \quad ANonce \| MNonce \| AP_{jMAC} \| MN_{MAC}) \tag{2}$$

MN은 자신이 생성한 PTK 값으로 MIC를 검증한다. 검증이 성공한다면 MN은 AP를 인증 할 수 있다. 마지막으로 MN은 자신이 생성한 PTK를 키로 이전 전송된 $MNonce$의 MIC 값을 생성하여 AP에 전달하고 AP는 이를 검증하여 상호인증을 마친다.

이 기법은 새로운 MN이 네트워크에 접근할 때마다 새로운 행렬을 만들어야 하며, 시스템 설정된 h에 따라 교환해야 하는 정보가 비교적 많을 수 있으며, 각 AP에서 유지해야 하는 정보는 이동 노드의 수에 비례하여 비교적 많다.

Ⅲ. 제안하는 프로토콜

본 장에서는 제안하는 프로토콜의 자세한 동작을 설명한다. 제안하는 프로토콜에서는 Bresson 등의 그룹키 기법을 응용하여 AP와 MN이 빠르게 상호인증을 수행한다. 제안하는 시스템은 IEEE 802.11 환경하에 동작하며 MN, AS, AP가 참여한다. 제안하는 시스템에 참여하는 AS는 S로 표기하며 MN은 M으로 표기한다. 네트워크에는 총 n개의 AP가 참여하는 것을 가정하며 각 AP는 $A_i (1 \le i \le n)$로 표기한다. 제안하는 프로토콜은 시스템 설정 단계, 초기 인증 단계, 재인증 단계로 구분된다. 시스템 설정 단계에서는 S가 네트워크 내에 존재하는 모든 A_i들과 통신하여 그룹키 생성을 위한 정보를 미리 교환한다. 향후에 어떠한 MN이 인증에 참여한다 하더라도 S는 시스템 설정 단계에서 수행된 일을 다시 반복하지 않는다. 초기 인증 단계와 재인증 단계는 실질적으로 인증이 이루어지는 단계로 네트워크 내의 A_i들에 M을 그룹 일원으로 포함시켜 그룹키를 생성하고 이 키를 사용하여 MN과 AP가 서로를 인증한다. 즉, 각 이동 노드마다 새로운 그룹이 만들어지며, 이 그룹마다 다른 그룹키가 사용되므로 특정 이동 노드가 사용하는 그룹키를 다른 이동 노드는 알 수 없다. 또 그룹키는 시스템 설정 단계에 참여한 모든 A_i와 인증 노드만 생성할 수 있으므로 같은 그룹키를 가지고 있는 개체는 서로가

정당한 개체임을 인증할 수 있다. 시스템 설정 단계에서 모든 A_i들은 이미 S와 그룹키 생성을 위한 값들을 교환한 상태이고, 대부분의 공개키 연산은 사전에 수행 가능하기 때문에 초기 인증 단계 및 재인증 단계에서는 비교적 간단한 연산만으로 그룹키를 생성하고 M과 A_i는 서로 상호인증을 수행할 수 있다.

3.1 시스템 가정

본 논문에서 다음과 같은 네트워크 환경을 가정한다.

- 네트워크는 하나의 AS와 이 서버가 관리하는 AP들로 구성된다. 즉, 서로 다른 인증 서버가 관리하는 영역 간 이동에 대해서는 고려하지 않는다.
- 네트워크에 참여하는 이동 노드는 AS가 확인할 수 있는 인증서를 가지고 있으며, 인증서가 없는 이동 노드는 네트워크에 참여할 수 없다.

본 논문에서 제안하는 프로토콜은 다음과 같은 가정을 한다.

- S는 모든 A_i의 공개키에 대한 인증서를 가지고 있다.
- A_i는 라우터 광고 메시지(advertisement)에 N_i를 같이 실어 전송하며 네트워크에 접근하는 M는 인증 과정을 수행하기 전에 N_i 값을 알 수 있다.

3.2 시스템 설정 단계

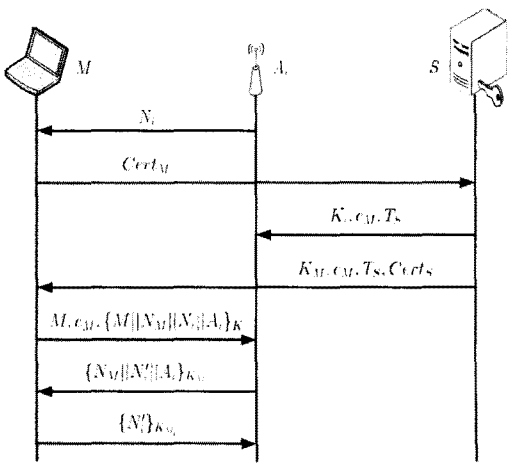
이 단계에서 S와 모든 AP는 그룹키 형성에 필요한 서로의 공개키 인증서를 교환하게 되는데, 실제로 메시지 교환을 통해 이루어지기 보다는 AP를 설치하는 과정에서는 각 AP에 등록되는 내용으로 이해하는 것이 옳다. 구체적으로 S는 매우 큰 소수 p를 선택하여 유한 순환군 Z_p^*을 만들고, 그것의 위수가 소수 q인 부분군과 그 군의 생성자 g를 생성하고, 이 군에 관한 정보 p, q, g를 공개한다. 각 소수의 크기는 해당 군에서 이산대수 문제가 계산적으로 어려울 정도로 충분히 커야 한다. 그 다음 S는 자신의 개인키 $x_S \in Z_q^*$를 선택하고 공개키 $y_S = g^{x_S}$를 계산한다. x_S 값은 S 내부에 안전하게 보관하고 공개키인 y_S는 공개한다. 각 A_i들도 동일한 방법으로 자신의 개인키 $x_i \in Z_q^*$를 선택하고 공개키 $y_i = g^{x_i}$를 계산한다. S는 모든 A_i의 공개키에 대한 인증서를 가지고 있다고 가정하고 있으므로 $\alpha_i = y_i^{x_S}$를 계산할 수 있으며, S는 이 값을 계산하여 저장하며 향후 그룹키 생성의 원소로 사용한다. A_i도 S의 공개키 y_S 값을 이용하여 $\alpha_i = y_S^{x_i}$ 값을 미리 계산해 둔다. 각 AP와 S가 초기에 유지하는 정보는 다음과 같다.

- AP : 개인키 $x_i \in Z_q^*$, 공개키 $y_i = g^{x_i}$, 인증서버의 인증서 $Cert_S$, $\alpha_i = y_S^{x_i}$
- S : 개인키 $x_S \in Z_q^*$, 공개키 $y_S = g^{x_S}$, 각 AP마다 AP의 인증서 $Cert_{A_i}$, $\alpha_i = y_i^{x_S}$

3.3 초기 인증 단계

만약 기존에 사용했던 그룹키가 없는 새로운 이동 노드 M이 네트워크로 접근을 시도하고자 하면 초기 인증 단계를 수행하여 A_i와 상호인증을 수행한다. 프로토콜의 수행 절차는 [그림 7]과 같다.

- 단계 **1.** M은 A_i를 통하여 그룹키 생성에 사용될 자신의 공개키 y_M이 포함된 인증서를 S에게 전송한다.
- 단계 **2.** S는 인증서를 검사하고 식 (3)을 이용하여 그룹키 K를 생성한다.



[그림 7] 제안하는 프로토콜의 초기 인증 단계

$$K = H(M \| c_M \| \alpha_M \| \alpha_1 \| \alpha_2 \| \cdots \| \alpha_n). \quad (3)$$

c_M은 M에게 할당된 카운터 값이며 $\alpha_M = y_M^{x_S}$, $\alpha_i = y_S^{r_i}$이다. c_M은 M에게 새로운 그룹키가 할당될 때 마다 1씩 증가한다. α_i들은 시스템 설정 단계에서 미리 계산되어져 있는 값으로 초기 인증 단계 프로토콜 수행 중에 새로 계산되어질 필요는 없다. 그리고 이전에 M이 S와 이 프로토콜을 수행한 적이 있다면 S가 이미 α_M를 가지고 있으므로 이 또한 새로 계산할 필요는 없다. 즉, S가 이미 α_M를 가지고 있는 경우에는 그룹키 생성 과정에서는 공개키 연산 없이 단 한 번의 해시 연산만 요구된다. AP의 개수가 많아질 경우에는 그룹키를 생성하기 위한 입력의 개수가 많아지는 문제점이 있지만 이 문제점은 그룹키를 다음과 같이 계산하여 극복 가능하다.

$$K = H(M \| c_M \| \alpha_M \| H(\alpha_1 \| \alpha_2 \| \cdots \| \alpha_n))$$

즉, 현재 네트워크에 운영되는 AP의 개수가 고정되어 있다면 $H(\alpha_1 \| \alpha_2 \| \cdots \| \alpha_n)$ 값은 이동 노드와 독립적으로 변하지 않는 부분이므로 사전에 한 번 계산한 후에 계속 사용할 수 있다.

- 단계 **3.** S는 $K_i = K \oplus H(M \| c_M \| T_S \| \alpha_i)$를 계산하여 c_M, T_S와 함께 A_i에게 전송한다. T_S는 그룹키 K의 만료 시간을 명시한 타임스탬프이다. K_i는 A_i가 그룹키 K를 생성하기 위한 시드 값으로 사용된다. K_i를 전송받은 A_i는 식 (4)를 이용하여 그룹키 K를 생성한다.

$$K = K_i \oplus H(M \| c_M \| T_S \| \alpha_i) \quad (4)$$

α_i는 미리 계산되어 있으므로 이 과정에서 한 번의 해시 연산으로 그룹키 생성이 가능하다.

- 단계 **4.** S는 $K_M = K \oplus H(M \| c_M \| T_S \| \alpha_M)$를 계산하여 이 값과 c_M, T_S를 M에게 전송한다. 이 값은 M이 그룹키 K를 생성하기 위한 시드 값으로 사용된다. K_M를 전송받은 M은 식 (4)와 유사한 방법으로 그룹키 K를 생성한다. α_M은 미리 계산되어 있다면 이 과정에서도 한 번의 해시 연산으로 그룹키 생성이 가능하다. 즉, 이 네트워크에 전혀 참여한 적이 없는 이동 노드는 한 번의 공개키 연산이 필요하지만, 이전에 참여하여 인증 서버의 인증서를 가지고 있는 이동 노드의 경우에는 공개키 연산 없이 그룹키 생성이 가능하다.

- 단계 **5.** 그룹키 K가 계산되면 M은 핸드쉐이크를 시작하기 위해 M, c_M, $\{M \| N_M \| N_i \| A_i\}_K$을 A_i에게 전송한다. N_M은 M이 생성한 난스 값이다.

- 단계 **6.** A_i는 M, c_M을 이용하여 자신이 가지고 있는 그룹키들 중 M에 해당하는 그룹키 K를 검색한다. 이 K를 이용하여 $\{M \| N_M \| N_i \| A_i\}_K$을 복호화하고 ID와 난스 값을 검사한다. ID와 난스가 올바른 것으로 판명되면 A_i는 M이 올바른 그룹키를 가지고 있으며 정당한 사용자임을 알 수 있다. 확인이 끝난 뒤에 A_i는 식 (5)에 따라 이번 세션에 데이터 암호화에 사용할 키 K_{Mi}를 생성한다. A_i는 이 키를 사용하여 $\{N_M \| N_i' \| A_i\}_{K_{Mi}}$를 계산하고 M에게 전달한다. N_i'는 A_i가 새로 생성한 난스이다.

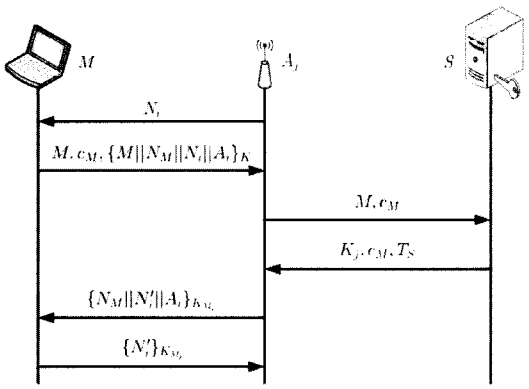
$$K_{Mi} = H(K \| N_M \| N_i) \quad (4)$$

- 단계 **7.** 메시지를 받은 M은 식 (5)를 이용하여 A_i와 동일한 값을 가지는 K_{Mi}를 생성하고 이것을 키로 $\{N_M \| N_i' \| A_i\}_{K_{Mi}}$를 복호화 한다. 만약 복호화에 성공하여 올바른 N_M과 A_i를 확인할 수 있다면 M은 A_i가 올바른 그룹키를 가지고 있으며 정당한 사용자임을 알 수 있다.

- 단계 **8.** 마지막으로 M은 A_i에게 대칭키 성립이 완료되었음을 알리기 위해 $\{N_i'\}_{K_{Mi}}$을 전송한다. A_i가 이 메시지를 성공적으로 복호화 한다면 A_i는 올바르게 대칭키가 성립되었음을 알 수 있다.

3.4 재인증 단계

M이 A_j로 핸드오프를 필요로 할 때 M과 A_j는 핸드쉐이크 과정만으로 빠르게 상호인증을 수행할 수 있다. [그림 8]은 제안하는 프로토콜의 재인증 단계를 보여주며 프로토콜의 수행 절차는 다음과 같다.



〔그림 8〕 제안하는 프로토콜의 재인증 단계

- 단계 **1.** M은 핸드쉐이크를 시작하기 위해 M, c_M, $\{M\|N_M\|N_j\|A_j\}_K$를 A_j에게 전송한다.

- 단계 **2.** 만약 A_j가 기간 만료 전의 그룹키 K를 가지고 있으면 초기 인증 단계의 6번째 과정부터 마지막까지 동일하게 수행하여 상호인증을 수행한다. 만약 A_j가 유효한 K를 가지고 있지 않으면 A_j는 K 생성을 위해 M, c_M을 AS에 전송한다.

- 단계 **3.** S는 M, c_M을 이용하여 자신이 가지고 있는 그룹키들 중 M에 해당하는 그룹키 K를 검색한다. 그리고 $K_j = K \oplus H(M\|c_M\|T_S\|\alpha_j)$를 생성하여 c_M, T_S와 함께 A_j에 전달한다. 나머지 부분은 초기 인증 단계의 6번째 과정부터 마지막까지 동일하게 수행하여 상호인증을 수행한다.

이동 노드가 새 AP로 핸드오프가 필요할 때 그 AP가 그룹키를 가지고 있지 않는 상태이면 AP는 인증 서버와 핸드오프 과정에서 통신이 필요하다. 기본적으로 AP와 이동 노드 간에는 상호 신뢰할 수 있는 구조가 아니므로 이와 같은 통신은 불가피하게 필요하다. 하지만 이것을 개선하기 위해 다음과 같은 방법을 사용할 수 있다.

- **방법 1.** 초기 인증 과정에서 인증 서버는 확립된 그룹키를 모든 AP에게 배포한다.

- **방법 2.** Mishra 등의 예측 기법을 활용하여 예측되는 다음 이웃 AP에게 사전에 그룹키를 전달한다.

방법 1은 최초에 많은 비용이 소요되지만 그룹키가 모든 AP에게 배포된 이후에는 인증 서버와 전혀 상호작용 없이 이동 노드와 AP 간에 상호 인증이 가능하다.

Ⅳ. 분 석

이 장에서는 제안하는 프로토콜의 안전성과 효율성을 분석한다.

4.1 안전성 분석

제안하는 프로토콜은 그룹키를 이용하여 MN과 AP가 상호인증하며 재생공격에 안전하다. 본 절에서는 제안하는 프로토콜에 적용되는 그룹키 기법의 안전성과 재생공격에 대한 안전성에 대해 살펴본다.

제안하는 프로토콜에서 오직 S만 그룹키 K를 생성할 수 있다면 A_i는 K를 가지고 있는 이동 노드는 S로부터 키를 부여 받은 인증된 이동 노드임을 믿을 수 있다. Bresson 등의 기법에서는 식 (3)을 사용하여 그룹키를 계산한다. DLP(Discrete Logarithm Problem)와 CDHP(Computational Diffie-Hellman Problem)에 의해서 오직 S와 A_i만이 α_i를 계산해낼 수 있으며, 모든 α_i를 계산할 수 있는 유일한 개체는 S 뿐이다. 따라서 오직 S만 식 (3)을 이용하여 K를 생성할 수 있다. 그런데 A_i는 K를 직접 전달 받지 않고 대신 K_i를 전달받는다. K_i로부터 K를 계산하기 위해서는 α_i 값이 필요하며, α_i는 S와 A_i만 생성할 수 있으므로 K_i로부터 K를 생성할 수 있는 것은 S와 A_i 뿐이다. 하지만 A_i는 자신이 생성한 K가 S가 생성한 K와 같은지 여부를 검증할 수 없다. 따라서 A_i는 M과의 핸드쉐이크 첫 번째 단계에서 M이 보낸 암호화된 메시지 $\{M\|N_M\|N_i\|A_i\}_K$를 K를 이용하여 복호화 함으로써 자신이 가진 K가 S가 생성한 K와 같은 키라고 추측할 수 있을 뿐이다. 제안하는 프로토콜에서 사용되는 그룹키의 안전성은 이 추측이 얼마나 정확한 것인가에 따라 결정된다. 즉, 공격자가 MN 또는 AP에게 S가 전달하는 값 대신에 다른 값을 보내 자신도 계산할 수 있는 키를 사용하도록 유도할 수 없어야 한다. 우선 공격자가 임의의 값을 K_i 대신 A_i에게 보낸다면 A_i는 식 (4)을 이용하여 K를 계산할 것이다. 하지만 공격자는 α_i를 알지 못하므로 A_i가 어떤 값을 계산하였는지 알 수 없다. 비슷한 이유에서 임의의 값을 K_M 대신에 MN에 전달하여도 공격자는 MN이 계산하는 값을 계산할 수 없다. 공격자가 A_i에게

는 K_i 대신 임의의 값 X_A를 보내고, MN에게는 K_M 대신에 X_M를 보내 A_i와 MN이 자신도 계산할 수 있는 키를 공유하도록 공격하는 시나리오를 생각하여 보자. A_i와 MN이 동일한 키 값을 계산하도록 유도하기 위해서는 키 계산 방식에 의해 다음이 성립해야 한다.

$$X_A \oplus X_M = H(M\|c_M\|T_S\|\alpha_i) \oplus H(M\|c_M\|T_S\|\alpha_M)$$

공격자는 X_A를 임의로 선택한 다음에 $X_M = K_i \oplus K_M \oplus X_A$을 계산하여 위 식이 만족하도록 만들 수 있다. 이 경우에 A_i와 MN이 사용하게 되는 $K = X_M \oplus H(M\|c_M\|T_S\|\alpha_M)$가 되지만 이 값은 공격자가 계산할 수 있는 값은 아니다. 따라서 α_i값들이 노출되지 않으면 공격자가 본 시스템을 공격할 수 있는 방법이 없으며, 식 (4)를 통해 확보한 대칭키가 서버가 생성한 키임을 믿을 수 있으며, 같은 그룹키를 소지한 개체끼리는 서로가 S를 통해 인증을 받은 정당한 개체임을 믿을 수 있다.

제안하는 프로토콜은 재생공격에 대해서도 안전하다. 공격자가 M과 A_i의 통신을 도청하고 있다가 후에 이 메시지들을 사용하여 거짓 인증을 시도하려고 해도 $\{M\|N_M\|N_j\|A_j\}_K$ 메시지 내부의 난스가 현재 유효하지 않기 때문에 사용할 수 없다. 즉, 항상 새로운 난스를 사용한다면 이전 메시지를 재생하여 공격에 성공할 수 없다.

4.2 효율성 분석

제안하는 프로토콜은 이동 노드가 처음으로 네트워크에 참여하는 경우에만 공개키 연산이 필요하며, 그 이후에는 공개키 연산을 사용하지 않고 해시 연산과 대칭키 연산만을 사용하여 인증을 수행한다. 최초 인증 과정에서 필요한 공개키 연산의 수는 다음과 같다.

- 이동 노드 : 인증 서버의 인증서를 확인하기 위한 공개키 연산과 α_M을 계산하기 위한 공개키 연산
- 인증 서버 : 이동 노드의 인증서를 확인하기 위한 공개키 연산과 α_M을 계산하기 위한 공개키 연산

서로의 인증서를 확인하기 위한 부분은 표준인 EAP-TLS와 박찬일 등의 프로토콜도 모두 필요하다. 두 프로토콜 모두 TLS 과정에서 마스터 키를 생성하기 위한 비밀 정보 교환을 위해 추가 공개키 연산이 필요하다. 따라서 최초 인증 과정에서 요구되는 공개키 연산 비용은 큰 차이가 없다. 하지만 박찬일 등의 프로토콜은 사용되는 행렬의 크기에 따라 교환되는 정보의 양이 비교적 많을 수 있다는 문제점을 가지고 있다.

핸드오프가 발생할 경우에 표준은 최초 인증과 동일한 과정을 다시 반복하기 때문에 빠른 핸드오프를 제공할 수 없다. 따라서 이것을 극복하기 위한 가장 기본적인 방법은 Mishra 등의 기법을 활용하여 다음 AP를 미리 예측하여 인증 서버와 상호작용 없이 핸드오프가 이루어지도록 하는 것이다. 하지만 다음 AP를 예측하기 위해서는 AP들과 인증 서버 간에 제한적인 통신은 반드시 필요하며, 건물 내부 등 다음 AP를 충분히 예측할 수 있고 이웃 노드의 수가 제한적일 경우에 효과적이다. 만약 이와 같은 예측 기법을 사용하지 않을 경우에는 인증서 대신에 빠르게 이동 노드를 인증하기 위한 방법이 필요하다. 제안된 프로토콜은 최초 인증에서 확립한 그룹키를 지속적으로 이용하여 이동 노드를 인증하고 있으므로 예측 기법을 사용하지 않아도 빠른 핸드오프를 제공할 수 있다. 특히 제안된 프로토콜은 재인증 과정에서 인증 서버는 단순 검색과 해시 연산만 필요하다. 반면에 박찬일 등의 프로토콜은 예측 기법을 사용하지 않을 경우에는 행렬 정보를 전달해 주어야 하는데, 그 정보의 크기가 비교적 클 수 있는 문제점을 가지고 있으며, 대칭키를 계산하기 위해 행렬 곱셈 연산이 필요하다. 따라서 제안된 프로토콜은 박찬일 등의 프로토콜보다 효율적이다.

V. 결 론

이 논문에서는 IEEE 802.11 환경에서 빠른 핸드오프를 위한 상호인증 프로토콜을 제안하였다. 제안하는 프로토콜은 Bresson 등의 그룹키 프로토콜을 이용하고 있으며, 네트워크 내의 AP들과 MN은 인증 서버가 생성한 그룹키를 이용하여 서로를 상호 인증한다. 네트워크에 참여한 경험이 있는 MN의 경우에 AP와 MN은 단순 해시 연산만을 사용하여 서로를 상호 인증할 수 있으며, 예측 기법을 사용하지 않아 핸드오프가 발생할 때 인증 서버의 참여가 필요하더라도 작은 크기의 메시지를 단순 검색과 해시 연산만을 사용하여 생성하여 AP에 전달하여 상호 인증을 할 수 있다.

참고문헌

[1] IEEE, "IEEE Standard for information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11 : Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications Amendment 6 : Medium Access Control(MAC) Security Enhancements," IEEE 802.11i, 2004.

[2] B. Pboba, D. Simon, "PPP EAP TLS authentication procotol," RFC 1510, IETF, 1999.

[3] A. Mishra, M. Shin, N.L. Petroni, T.C. Clancy, W.A. Arbaugh, "Pro-active Key Distribution using Neighbor Graphs," *IEEE Wireless Communications*, Vol. 11, No. 1, pp. 26-36, 2004.

[4] R. Blom, "An optimal class of symmetric key generation systems," Advances in Cryptology, Proc. of Eurocrypt 84, pp. 335-338, 1984.

[5] C. Park, C. Hur, C. Kim, Y. Shin, H. Yoon, "Pre-authentication for Fast handoff in Wireless Mesh Networks with Mobile APs," *Proc. of the 7th Int. Workshop on Information Security Applications*, pp. 670-694, 2006.

[6] E. Bresson, O. Chevassut, A. Essiari, D. Pointcheval, "Mutual Authentication and Group Key Agreement for Low-power Mobile Devices," *Computer Communications*, Vol. 27, No. 17, pp. 1730-1737, 2004.

[7] 강유성, 오경희, 정병호, "무선랜 보안기술의 진화동향 및 전망," *전자통신동향분석*, 제 18권, 제 4호, pp. 36-46, 2003.

〈著者紹介〉

이 창 용 (Changyong Lee) 정회원

2004년 2월 : 강원대학교 전기전자정보통신공학부(학사)
2008년 2월 : 한양대학교 컴퓨터공학과(석사)
2008년 2월~현재 : 한국정보보호진흥원 응용기술팀 연구원
<관심분야> 네트워크 보안

김 상 진 (Sangjin Kim) 종신회원

1995년 2월 : 한양대학교 전자계산학과(학사)
1997년 2월 : 한양대학교 전자계산학과(석사)
2002년 8월 : 한양대학교 전자계산학과(박사)
2003년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수
<관심분야> 암호기술 응용
URL : http : //infosec.kut.ac.kr/sangjin/

오 희 국 (Heekuck Oh) 종신회원

1983년 : 한양대학교 전자공학과(학사)
1989년 : 아이오와주립대학 전자계산학과(석사)
1992년 : 아이오와주립대학 전자계산학과(박사)
1993년~1994년 : 한국전자통신연구원 선임연구원
1995년~현재 : 한양대학교 컴퓨터공학과 교수
<관심분야> 암호프로토콜, 네트워크 보안
URL : http : //infosec.hanyang.ac.kr/~hkoh/

박 춘 식 (Choonsik Park) 종신회원

1981년 : 광운대학교 졸업
1983년 : 한양대학교 전자통신전공(석사)
1995년 : 일본동경공업대학교 정보보호전공(박사)
1982년~1999년 : 한국전자통신연구원 부장
2000년~현재 : 한국전자통신연구원부설연구소 책임연구원
<관심분야> 암호이론, 정보이론, 네트워크 보안