DOI: 10.3745/KIPSTC.2009.16-C.3.299

EAP 성능 향상을 위한 흐름 제어 및 오류 복구 방식의 제안과 시뮬레이션

차 은 철[†] · 한 찬 규[††] · 최 형 기[†††]

요 약

인터넷의 사용이 증가하면서 다양한 네트워크 접근 기술이 개발 되었다. Extensible Authentication Protocol (EAP)은 네트워크 접근 시 요구되는 인증 과정을 지원하는 프로토콜이다. EAP는 다양한 인증 기법을 지원할 수 있는 확장성과 유연성 때문에 대부분의 네트워크 환경에서 사용되고 있다. 그러나 EAP의 가장 핵심적인 특성인 "lock-step" 흐름 제어는 에러 복구, 재전송 메커니즘에 영향을 주어 EAP는 물론 전체 인증과정의 성능을 크게 저하 시킨다. 본 논문은 EAP의 성능 저하 효과를 분석하고 새로운 흐름 제어 방식을 제안한다. NS-2를 사용한 시뮬레이션 결과에 따르면 제안한 흐름 제어 방식을 적용한 EAP는 최대 53% 성능 향상을 기대할 수 있다.

키워드 : EAP, 네트워크 접근, 무선랜인증, 802.11i, 성능평가

Proposal and Simulation of Flow Control and Error Recovery in EAP for Performance Improvement

Eun-Chul Cha[†] · Chan-Kyu Han[††] · Hyoung-Kee Choi[†††]

ABSTRACT

Followed by the popularity of the Internet, a number of access technologies to the Internet have been developed. EAP is an authentication framework. It is designed to provide the authentication functionality in the access network. Because of its flexibility and extensibility EAP poses a global solution for the authentication supported by many access networks. However, EAP has critical weaknesses in the protocol which may, in turn, decrease the EAP performance. Some of the weaknesses are caused by the "lock-step" flow control which only supports a single packet in flight. Considering the weaknesses, we propose a solution for the flow control. Using simulation we prove that our solutions improve the EAP performance.

Keywords : EAP, Network Access, Wireless Authentication, 802.11i, Performance Evaluation

1. 서 론

인터넷의 발달로 사용자는 언제 어디서나 네트워크 서비스에 접속 할 수 있게 되었다. 인터넷 사용의 증가는 네트워크 접속 기술에 대한 요구도 높였다. IEEE 802.11이나 802.16 와 같은 표준 네트워크에 접속하고자 하는 사용자는 서비스 제공자와의 인증 절차를 거쳐야 한다. 인증 절차는 사용자 단말기와 기지국 혹은 액세스 포인트 간의 정보 교환으로 시작된다. 기지국은 사용자의 등록정보를 통해 사용자 단말기의 접근 유효성을 확인한다.

인증 절차를 위해 Authentication and Key Agreement (AKA), Transport Layer Security (TLS), Tunneled Transport Layer Security (TTLS)와 같은 다양한 기법이 존재한다. 이러한 기법들은 오직 인증 방식에 대해 정의하고 있기 때문에, 실제 네트워크 환경에서 네트워크 프로토콜을 통해 사용되기 위해서는 추가적인 작업이 필요하다. Extensible Authentication Protocol (EAP)는 다양한 인증 기법들이 네트워크 환경에 적용될 수 있는 형식 체계를 정의한다. EAP가 제공하는 메시지 포맷과 메시지 교환 방식을 따라 인증 절차가 진행되고, 다양한 기법에 따른 가변적인 인증 내용은 EAP 내 암호화되어 전달된다.

다양한 인증 기법을 지원할 수 있는 유연성과 확장성에 불구하고, EAP는 성능과 관련된 치명적인 약점들을 가지고 있다. 약점을 유발하는 요소는 크게 세 가지이다. 첫 째, EAP의 재전송 및 에러 복구 처리과정 이다. EAP는 손실된 메시지를 재전송하기 위해 전송 왕복 시간 (Round Trip Time : RTT)을 EAP 타이머로 측정한다. 그런데 인증 절차에서

† 정 회 원 : 성균관대학교 정보통신공학부 공학석사
†† 준 회 원 : 성균관대학교 휴대폰학과 박사과정
††† 정 회 원 : 성균관대학교 정보통신공학부 조교수(교신저자)
논문접수 : 2008년 10월 9일
수 정 일 : 1차 2009년 1월 15일
심사완료 : 2009년 2월 3일

다른 네트워크 개체와 접속하거나 사용자의 입력이 필요하게 되는 특정한 경우 재전송 타이머의 측정에 오류가 발생한다. 둘 째, EAP의 "lock-step" 흐름 제어이다. "lock-step" 흐름 제어는 메시지 교환 절차에서 한 번에 하나의 패킷 전달을 허용하는 것을 말한다. 이로 인해 여러 개의 메시지를 보내야 할 때 패킷의 수 만 큼 전송 지연이 일어나게 된다. 셋 째, 크기가 큰 메시지에 대한 잘 못된 처리이다. EAP의 최대 전송 단위 (Maximum transmission unit : MTU)보다 큰 크기의 메시지는 분할된다. 분할되며 수가 늘어난 메시지는 앞서 언급한 흐름 제어의 문제점과 연결되어 전송 지연이 일어난다.

EAP의 성능 향상을 위해 본 논문은 "lock-step" 흐름 제어 대신 "sliding window "흐름 제어를 제안한다. sliding window 란 여러 개의 메시지를 파이프라인 형태로 전송 하는 방식을 말한다. sliding window 흐름제어는 전송 지연으로 인한 EAP 성능 저하를 방지 할 수 있다. 흐름 제어의 변경에 따라 에러 복구 방식에 대한 고려도 필요하게 된다. 다양한 에러 복구 방식 중에서 자동 재전송 (Automatic-Repeat-reQuest :ARQ)을 다루도록 한다. 흐름 제어 정책에 적절한 ARQ의 적용은 통신 효율에 큰 영향을 미친다. 본 논문은 기존 "Stop-and-Wait" ARQ 대신 "Selected Repeat" ARQ를 적용 하는 것을 제안한다. Sliding window 흐름 제어에 Selected Repeat ARQ 결합 했을 경우, 전송 오류로 인한 에러 발생 시 손실된 메시지만 재전송 할 수 있어 전체적인 통신 효율을 상당히 높일 수 있다. EAP 타이머의 오류 측정은 추가적인 확인 응답 문자 (Acknowledgment : ACK)로 타이머 측정 시작 시점을 정의함으로 해결 한다.

논문의 구성은 다음과 같다. 2장에서 EAP와 관련된 연구들을 살펴보고 3장에서 EAP의 주요 특성을 설명한다. 4장은 시뮬레이션 환경과 진행 과정을 설명한다. 5장에서 기존 EAP의 문제점을 정의한 뒤 제안한 방식을 적용한 EAP와 성능을 분석 한다. 그리고 6장에서 결론을 맺는다.

2. 관련 연구

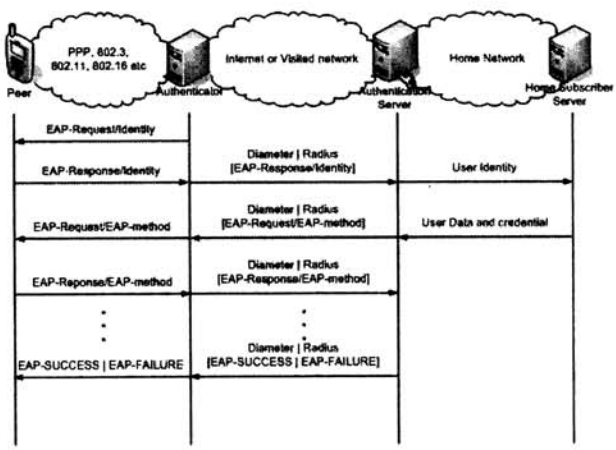
EAP에 대한 연구들은 주로 EAP를 여러 네트워크에서 활용하는 것과 관련되어 있었다. 이런 노력들의 결과로 IEEE 802 계열의 네트워크들에서 인증을 위해 사용되는 802.1X[1], 802.11i[2]와 같은 프로토콜들이 개발되었다. Avesh K. Agarwal[3]은 IEEE 802.11 WLAN에서 사용되는 보안 메커니즘들의 성능을 측정하였다. 이 논문에서는 IPSec과 RADIUS 같은 보안 프로토콜과 함께 EAP-TLS와 EAP-MD5의 성능을 측정하였다.

최근에는 EAP를 이기종 망간의 통합을 위해 사용하기 위한 연구들이 진행되고 있다. Kambourakis[4]은 WLAN과 3G가 통합된 네트워크에 인증을 위해 EAP에 기반하는 새로운 인증 메커니즘을 제안하고 있다. 이 연구에서는 먼저 기존의 WLAN과 3G 네트워크 간의 통합을 위해 사용되던 EAP-AKA의 문제점들을 소개하고 그 것을 해결할 수 있는 인증 메커니즘을 제시한다. 이 메커니즘은 EAP-TLS의 기반으로 AKA 프로토콜을 수정하여 AKA에 비해 강한 보안을 제공하도록 하였다. Yao Zhao[5] 역시 WLAN과 3G 네트워크 간의 통합을 위해 EAP-AKA 대신 EAP-TLS와 EAP-TTLS를 사용하는 방안을 제안했다. 그들은 각각의 보안 메커니즘들의 성능과 에너지 소비를 비교하였는데 EAP-TLS와 EAP-TTLS는 EAP-AKA 보다 에너지 소비는 높지만 빠른 인증이 가능한 것으로 나타났다.

위에서 소개한 것처럼 EAP의 연구는 대부분 EAP 인증의 성능 측정이나 여러 네트워크에 EAP를 적용하기 위한 주제들에 집중되었으며 EAP 자체의 성능 개선을 위한 연구는 거의 진행되지 않았다.

3. Extensible Authentication Protocol (EAP)

802.11 이나 802.16 과 같은 액세스 네트워크에 접근을 원하는 사용자들은 서비스 제공자로 부터 허가를 받아야 한다. 이런 절차는 단말과 액세스 포인트 간에 인증을 통해 이루어진다. 인증을 위해서는 AKA, TLS 그리고 TTLS와 같은 다양한 인증 메커니즘들이 사용될 수 있다. EAP는 이러한 인증 메커니즘들의 사용을 지원하기 위한 프레임워크를 제공한다. 즉, EAP에서는 인증에 필요한 메시지 형식과 메시지 교환 방식만을 정의하고 실제 인증은 EAP에 캡슐화 되어 있는 인증 메커니즘에 의해 수행된다. EAP의 인증 모델은 일반적으로 Peer, Authenticator (이하 AUTH) 그리고 Authentication Server (이하 AS)로 구성된다. AUTH는 EAP 인증을 시작하는 네트워크 개체로 802.11에서의 AP (Access Point)가 대표적인 AUTH이다. Peer는 일반적으로 인증을 받는 단말을 의미한다. AUTH는 두 가지 모드로 동작할 수 있는데 첫 번째는 AUTH 안에 인증 메커니즘을 구현하고 직접 Peer와 인증을 수행하는 것이다. 두 번째는 인증 메커니즘을 수행하는 인증서버 (AS)를 별도로 두고 AUTH 자신은 AS와 Peer 사이에서 프록시 역할을 하는 것이다 (그림 1 참조). 일반적으로 AS는 Diameter나 RADIUS와 같은 AAA 서버로 구성된다. 추가적으로 유저의 정보를



(그림 1) EAP 인증 모델과 인증 과정

저장하는 디렉토리 서버가 사용될 수 있다. 이 디렉토리 서버에는 유저의 ID와 비밀키와 같은 유저의 인증을 위한 정보가 저장되어 있으며 이 정보들은 AS의 요청에 의해 전송된다. 대표적인 예로 3G의 Home Subscriber Server (HSS)가 있다. 본 논문에서는, 별다른 언급이 없으면 (그림 1)과 같은 인증모델을 사용하는 것으로 한다.

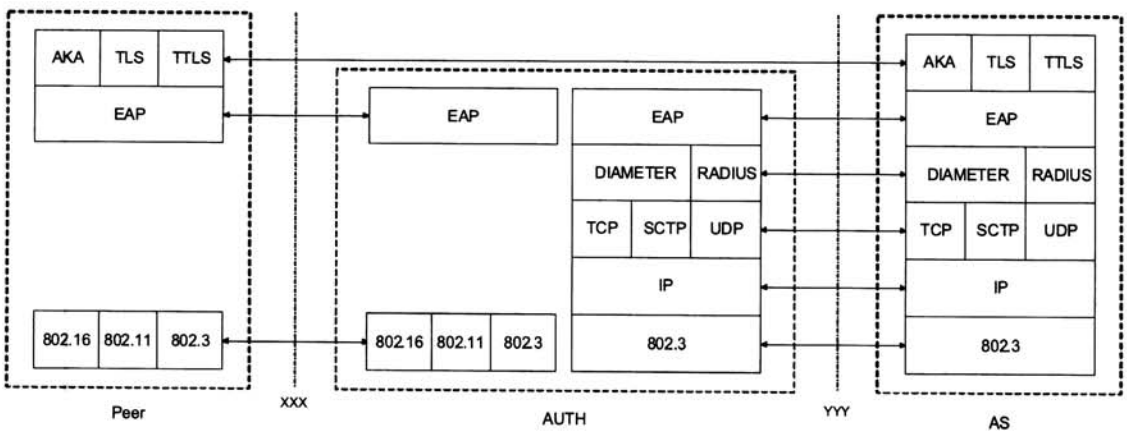
(그림 2)는 EAP를 사용하는 인증 구조의 프로토콜 스택의 예를 보여준다. 그림에서 보면 EAP는 XXX 인터페이스에서는 IEEE 802.3이나 IEEE 802.16과 같은 데이터 링크 계층 위에서 동작하며 YYY 인터페이스에서는 Diameter나 RADIUS와 같은 프로토콜 상위에서 동작한다. 현재 EAP는 IEEE 802 계열 유선망을 위한 802.1X, 무선랜을 위한 802.11i 그리고 802.16의 Privacy Key Management (PKM) 등의 프로토콜에서 활용되고 있다. EAP 상위 계층에는 여러 인증 메커니즘들이 있는데 대표적으로 EAP-AKA, EAP-TLS 그리고 EAP-TTLS 등이 있다. EAP-AKA는 차세대 개인이동통신 방식 (Universal mobile telecommunications system :UMTS)의 인증과 키 교환을 위해 사용되는 AKA 프로토콜에 기반을 두고 있다. AKA 프로토콜은 유럽식 디지털 이동통신 방식 (Global System for Mobile communication: GSM)과 호환성을 제공하며 Peer 뿐 아니라 AUTH 까지 인증하는 상호인증방식이다. AKA의 인증방식은 인증상대가 같은 비밀키를 공유하며 상대방을 서로 인증하는 상호인증방식으로서, 인증서 방식에 비해 처리속도가 빠르고 인증에 필요한 메시지의 크기도 작다. 그러나 인증에 사용되는 키 길이와 암호화 알고리즘이 고정되어 있어 유연성이 부족한 것이 문제점으로 지적이 되고 있다. EAP-TLS는 Internet Engineering Task Force (IETF) 표준인 TLS에 기반으로 하는 인증서 방식의 인증 메커니즘이다. 상호인증을 위해서는 Peer와 AUTH 모두가 인증서를 가져야 하는데 인증서의 크기는 이론상 16MB 까지 가능하다. 큰 인증서를 갖게 된다면 인증서를 전송해야 되는 EAP에게 큰 부담으로 작용한다. EAP-TTLS는 EAP-TLS 인증의 확장된 형태이다. EAP-TTLS는 TLS를 이용하여 먼저 서버 인증과 키 교환을 수행하고 Peer와 AUTH 사이에 보안 채널을 생성한다. 그 후에 CHAP[10]과 같은 id-password 방식의 인증을 이용하여 클라이언트 인증을 수행한다. 따라서 EAP-TTLS는 서버의 인증서만을 필요로 하며 클라이언트의 인증서는 필요로 하지 않는다. 그러나 TTLS는 TLS의 양방향 초기처리 즉, 핸드세이크 과정 후에 추가적인 클라이언트 인증을 수행하므로 인증을 위해 교환되는 메시지 수가 TLS보다 많다.

EAP는 하위 계층과 상위 계층의 인증 메커니즘에 독립적으로 동작한다. 다양한 종류의 액세스 네트워크와 인증 메커니즘을 지원할 수 있다는 유연성과 새로운 인증 메커니즘이 추가되거나 기존의 메커니즘이 수정되어도 EAP를 수정 없이 사용할 수 있는 확장성이 뛰어나다.

반면, EAP는 하위 계층의 전송에 대한 신뢰성을 가정하지 않고 있다. (그림 2)에서의 YYY 인터페이스에서 사용되는 AAA 중 Diameter 는 TCP와 SCTP에서 그리고 RADIUS는 RADIUS 자체에서 신뢰성을 보장하고 있다. (그림 2)의 XXX인터페이스에서 채택한 데이타링크 계층에서 EAP 메시지의 신뢰성이 보장이 되지 않을 수 있는데 이 경우에 EAP 메시지의 신뢰성 있는 전송은 EAP에서 보장해야만 한다. 이를 위해 EAP에서 자체적으로 신뢰성 보장에 필요한 흐름 제어 메커니즘, 에러복구 및 재전송 메커니즘 등을 지원해야 하는데 Request for Comments (RFC) 에서는 자세히 명시되어 있지 않다. EAP 관련 RFC[6]에 따르면 EAP에서는 Stop-and-wait 방식과 유사한 흐름 제어와 에러 복구를 수행한다. AUTH는 인증에 필요한 요청 메시지를 이전에 보내 요청 메시지의 응답을 받을 때까지 보내지 못하게 된다. 특정 시간에 하나의 요청 메시지만이 허용되는 것이다. EAP 메시지가 유실이 되면 송신 측 EAP 는 재전송 타이머로 유실을 인지하고 해당 메시지를 재전송 하게 된다. 만일, EAP 메시지가 인증서를 포함하고 있어서 크기가 EAP의 MTU를 넘게 되면 EAP 메시지는 최대 EAP의 MTU 크기로 분할된다.

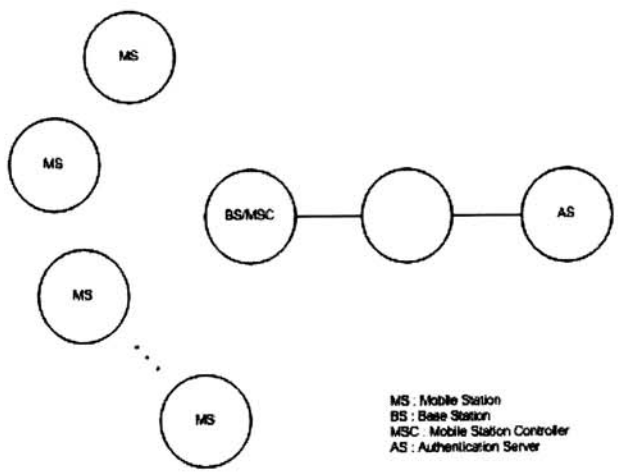
4. 시뮬레이션

본 논문에서는 EAP의 문제점들을 분석하고 논문에서 제



(그림 2) EAP 인증 프로토콜 스택의 예

시하는 해결책을 검증하기 위해 시뮬레이션을 사용하였다. 사용된 시뮬레이터는 ns-2[7]이며 시뮬레이션을 위해 선택한 네트워크(access network)는 802.16이다. 802.16은 PKM (Privacy Key Management)이라는 인증 프로토콜을 제공하며 PKM의 새로운 버전인 PKMv2에서 EAP를 사용하는 인증과정이 추가 되었다. 우리는 PKMv2의 EAP 인증을 시뮬레이션 하기 위해 미국 국립표준·기술 연구소 (National Institute of Standards and Technology : NIST)에서 개발한 Wimax ns-2 모듈을 사용하였다. 이 모듈의 현재 버전에서는 인증 과정이 구현되어 있지 않기 때문에 모듈에 802.16 표준[8]에서 정의하고 있는 EAP 인증 과정을 추가하였다. 시뮬레이션을 위해 구성된 네트워크는 (그림 3)과 같다. 시뮬레이션의 네트워크는 100개의 단말과 1개의 기지국 그리고 인증 서버로 구성된다. 단말들은 Peer의 역할을 하며 주어진 가중치 값에 의해 결정되는 불규칙한 시간 간격으로 네트워크에 들어와서 인증 요청을 발생한다. 기지국에 위치한 AUTH가 단말로부터 인증 요청을 받으면 인증을 시작하며 Peer와 AS 사이에 인증 메시지가 교환된다. 기지국과 인증 서버 사이는 1.5 Mbps의 대역폭 10ms의 지연을 가지는 두 개의 링크를 통해 연결 되어 있으며 전송 프로토콜로는 TCP 그리고 AAA 프로토콜로는 Diameter를 선택하였다. 시뮬레이션에서 Diameter는 TCP와 EAP 사이에서 패킷을 메시지를 전달하는 역할을 하도록 구현되었다. 우리는 <표 1>과 같은 시뮬레이션 파라미터를 가지고 시뮬레이션을 수행하여 802.16 네트워크에서 EAP 인증에 걸리는 시간을 측정하였다. <표 1>에서 볼 수 있듯이 802.16에서 제공되는 물리 계층 모드 중 OFDM TDD 모드를 선택하였으며 MAC 프레임의 길이는 4ms로 설정하였다. EAP의 MTU 크기는 표준에서 지정하는 최소 MTU 크기인 1020 바이트로 설정하였다. 시뮬레이션에서 암호학적 연산과 같은 프로세싱 지연은 고려하고 있지 않으며 인증 시간에 영향을 미치는 지연의 대부분은 메시지가 네트워크에서 전송되는 동안 발생한다. 그러므로 시뮬레이션 결과에 영향을 미치는 요소는 크게 두 가지로 볼 수 있다. 첫 번째는 인증 요청을 시



(그림 3) 시뮬레이션의 네트워크 구성

〈표 1〉 시뮬레이션 파라미터

파라미터	값
802.16 PHY mode	OFDM
802.16 Duplex mode	TDD
802.16MAC Frame duration	0.004
The number of mobile station	100
EAP MTU	1020 byte

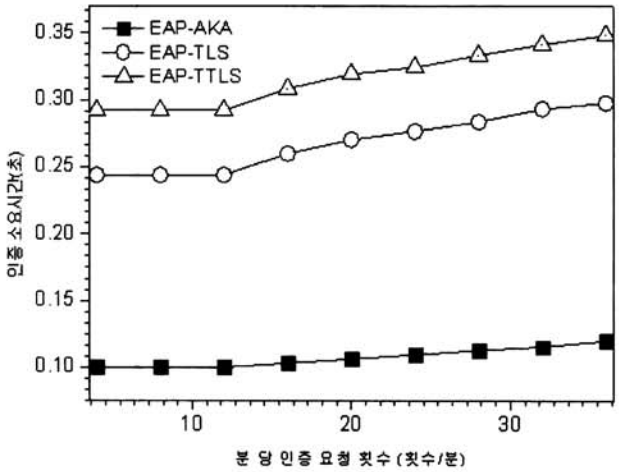
도하는 단말의 개수이다. 802.16은 대역폭 요청 및 할당에 기반한 MAC을 사용하여 상향링크 대역폭을 관리한다. 단말이 인증 메시지를 보내기 위해서는 충분한 대역폭을 기지국으로부터 할당 받아야 한다. 대역폭을 할당 받기 위해 단말은 한정된 대역폭 요청 슬롯에 대역폭 요청 메시지를 실어 보내야 한다. 동시에 여러 단말이 대역폭 요청 메시지를 보내려고 시도할 때 경쟁이 발생하며 802.16 표준에서는 여러 가지 물리 계층 모드마다 CDMA나 slotted aloha와 같은 다른 경쟁 처리 방식을 정의하고 있다. 시뮬레이션에는 선택한 OFDM PHY 모드에서는 slotted aloha 방식으로 경쟁을 처리한다. 이 경쟁은 메시지 전송에 지연을 발생시키며 전체 인증 시간에 영향을 줄 것이다. 두 번째는 인증 메시지의 개수와 크기이다. EAP-AKA의 경우 전체 인증에 교환되는 메시지는 6개로 인증을 위해 2 RTT가 필요하며 EAP-TLS의 경우 5 RTT가 필요하다. TTLS의 경우 12개의 메시지를 교환하며 6 RTT가 소요된다.

5. EAP 문제점 정의 및 해결

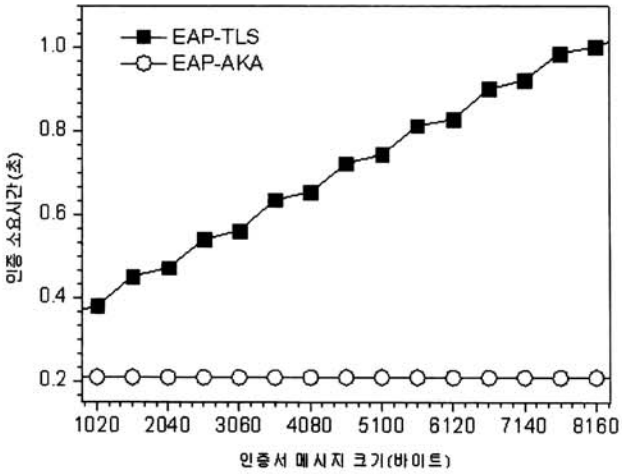
이 장에서는 시뮬레이션을 통해 EAP의 문제점을 분석하고 그 문제점을 해결하기 위한 방법을 제시한다.

5.1 전체적인 분석

먼저 시뮬레이션을 통해 EAP에서 동작하는 인증 메커니즘들의 성능을 측정하였다. 본 논문에서 선택한 인증 메커니즘은 EAP-AKA, EAP-TLS 그리고 EAP-TTLS 이다. (그림 4)는 단말들이 발생시키는 인증 요청 수에 따른 인증 시간을 보여준다. 시뮬레이션에서 인증 시간에 가장 크게 영향을 미치는 요인은 4장에서 언급한 것처럼 인증 요청을 시도하는 단말의 수와 인증 메시지의 크기와 개수이다. 시뮬레이션에서 EAP-(T)TLS에서 사용하는 인증서의 크기는 256Byte로 고정하였다. 이는 인증서의 크기가 시뮬레이션 결과에 영향을 미치지 않도록 하기 위해서이며, 256Byte 인증서의 크기는 EAP-(T)TLS의 성능을 가장 최적화 시키는 인증서 값으로 이는 (그림 5)에서 볼 수 있다. 또한 프레임 오류율 또한 1%로 고정하였다. 인증 요청을 시도하는 단말의 수는 802.16 MAC에서의 경쟁으로 인한 지연으로, 인증 메시지의 크기와 개수는 네트워크에서의 전송 지연으로 결과에 영향을 미친다. (그림 4)에서 볼 수 있듯이 경쟁으로 인한 지연은 인증 요청 수가 12 요청 횟수/분 이상에서 나타나며 인증 요청 수가 증가함에 따라 인증 시간이 증가하



(그림 4) EAP 인증 메커니즘들의 성능 분석



(그림 5) 메시지 분할 시 EAP-TLS의 성능 변화

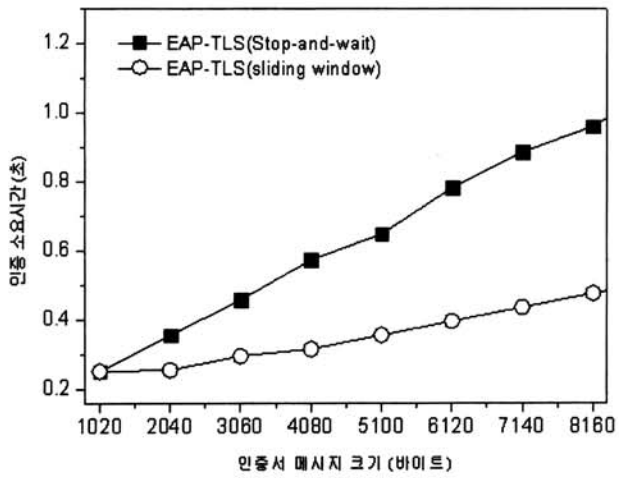
는 것을 확인할 수 있다. 반면에 인증 메커니즘 간에 존재하는 인증 시간의 차이는 교환되는 메시지의 수 때문에 발생한다. 4장에서 언급했듯이 EAP-AKA, EAP-TLS 그리고 EAP-TTLS는 각각 인증에 2 RTT, 5 RTT 그리고 6 RTT가 소요된다. 분당 인증 시도의 수가 12에 미치지 못할 경우, 즉 단말간의 경쟁에 의한 영향이 적을 경우 인증 시간은 필요한 RTT에 비례하는 것을 볼 수 있으며 1 RTT는 약 50ms라는 것을 예측할 수 있다.

5.2 흐름 제어와 메시지 분할

3장에서 언급한 것과 같이 EAP는 Stop-and-Wait 방식으로 흐름제어를 수행한다. 대부분의 인증 방식들은 통신 상대로부터 메시지를 수신 및 처리한 후에 비로소 전송할 메시지를 발생시킨다. 그러므로 확인 응답을 받기 전에 하나의 메시지만을 보낼 수 있는 Stop-and-wait 방식의 흐름제어를 사용해도 큰 문제가 없다. 그러나 인증 방식이 전송하는 메시지들이 EAP MTU를 초과하여 메시지가 분할되면서 문제가 발생하기 시작한다. EAP는 메시지가 분할 될 경우에는 데이터가 없는 EAP-Request 혹은 EAP-Response 메시지를 확인 응답으로 사용하여 모든 분할 메시지를 Stop-and-wait 방식으로 전송한다. 모든 분할 메시지를 Stop-and-wait 방식으로 보낼 경우 분할된 메시지 수만큼의 RTT가 인증에 걸리는 시간에 추가될 것이다. 이 때문에 EAP-AKA와 같은 인증 방식들은 분할을 피하기 위해 항상 EAP MTU 보다 작은 메시지를 전송하도록 권고하고 있다. 그러나 인증서 기반의 인증 방식들을 사용할 경우에 분할을 피하는 것은 어렵다. 예를 들어 TLS에서 인증서 메시지(TLS certificate message)는 이론상 16 메가바이트까지 길어질 수 있다. 표준은 최소 MTU를 1020 바이트로 규정하고 있으므로 최악의 경우 인증에 걸리는 시간은 16000 RTT 이상이 소요되게 된다.

메시지 분할 시 EAP 성능의 변화를 확인하기 위해 EAP-TLS의 메시지 중 서버 측의 인증서 메시지의 크기를 변화하면서 시뮬레이션을 수행하였다. (그림 5)는 TLS 서버 인증서 메시지의 크기에 따른 인증 시간의 변화를 보여준다. <표 1>에서 볼 수 있듯이 EAP MTU의 크기는 1020 바이트로 설정하였다. 프레임 오류율은 1%로 설정하였다. 그러므로 1020 바이트보다 큰 메시지는 분할 될 것이다. 앞서 설명한 것처럼 메시지가 분할 될 경우 분할 메시지 수만 큼의 RTT가 인증 시간에 추가된다. (그림 5)에서 볼 수 있듯이 EAP-TLS 인증 시간은 인증서 크기가 증가함에 따라 계속적으로 증가한다. 인증서의 크기가 1020 바이트 일 때 EAP-TLS의 인증 시간은 EAP-AKA에 비해 40% 높지만 인증서의 크기가 8160 바이트로 증가하면서 EAP-TLS는 인증을 마치기 위해 EAP-AKA의 약 4.8배의 시간을 소요한다.

많은 데이터를 Stop-and-Wait 방식으로 보내는 것은 명백하게 비효율적이다. EAP의 흐름 제어가 성능에 미치는 영향을 확인하기 위해 다른 흐름제어 방식을 시뮬레이션에 적용하여 인증 시간을 측정하였다. 선택한 흐름 제어 메커니즘은 sliding window를 이용한 흐름제어 방식이다. Stop-and-Wait 대신 sliding window 방식을 사용하면 송신 측은 보낸 메시지에 대한 확인 응답을 받기 전에 정해진 윈도우 크기와 같은 수의 메시지를 보낼 수 있다. 시뮬레이션에서 sliding window의 윈도우 크기는 7로 고정하였으며 서버 인증서 메시지의 크기를 증가시키며 반복적으로 수행되었다. 프레임 오류율은 1%로 설정하였다. (그림 6)에서 볼 수 있듯이 인증서 메시지 크기가 EAP MTU 크기(=1020 바이트)보다 크지 않을 때는 sliding window를 적용하기 전과 적용한 후는 크게 차이가 없다. 인증서 메시지가 1020 바이트일 경우, 즉 분할이 발생하지 않을 경우 sliding window의 적용에 상관없이 인증 시간은 252ms가 소요되었다. 그러나 인증서 메시지 크기가 1020 바이트를 초과하여 분할이 발생하기 시작되면서 성능의 차이는 뚜렷하게 나타난다. 시뮬레이션 결과 인증서의 메시지가 2040 바이트로 증가할 경우 sliding window를 적용하면 28%의 성능 증가를 보이며 인증서가 메시지가 8160 바이트일 경우 53%가 증가하는 것을 확인할 수 있었다.



(그림 6) Sliding Window을 적용 했을 때의 인증 시간 변화

EAP에 sliding window를 적용하기 위해서는 프로토콜에 몇 가지 수정이 필요하다. 우선 송신 측과 수신 측 모두에 최대 윈도 크기만큼의 버퍼가 필요하며 수신 측이 현재 버퍼 정보를 확인 응답 메시지를 통해 송신 측에 알리는 것이 필요하다. 그 대표적인 예는 TCP의 window size 필드이다. EAP의 헤더에는 header length 필드가 없으므로 필드의 추가는 option 필드의 방식으로는 불가능하며 EAP 헤더의 변경이 불가피하다. 일반적으로 윈도우 크기는 2^m - 1(m은 sequence 번호의 비트 수)와 같거나 작아야 하므로 추가되는 window size 필드의 크기는 EAP에서 sequence 번호로 사용되는 identifier 필드의 크기인 8 비트로 충분하다. 마지막으로 확인 응답 메시지는 EAP에서 이미 정의하고 있는 데이터가 없는 EAP Request 및 Response 메시지를 사용한다.

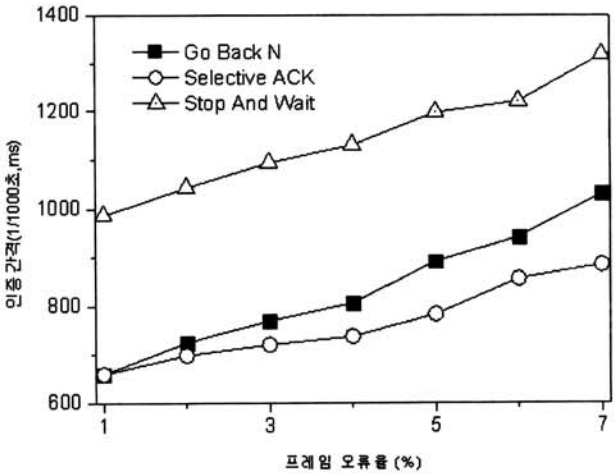
5.3 에러 복구

3장에서 언급한 것과 같이 EAP는 하위 계층이 신뢰성 있는 전송을 지원하지 않는다고 가정하기 때문에 자체적으로 오류 복구 메커니즘을 지원해야 한다. EAP는 Stop-and-wait ARQ 기반의 오류 복구를 사용하고 있다. Stop-and-Wait ARQ 오류 복구 방식은 Stop-and-Wait 흐름제어에 기반하기 때문에 앞서 설명한 sliding window를 사용하기 위해서는 오류 복구 방식 역시 수정되어야 한다. 우리는 에러 복구 방식으로써 Go-Back-N ARQ와 Selective repeat ARQ를 선택하였다. Go-Back-N ARQ와 Selective repeat ARQ는 sliding window를 사용하는 오류 복구 방식으로써 확인 응답을 받기 전에 윈도우 크기만큼 메시지를 보낼 수 있고 확인 응답을 통해 메시지가 제대로 전송되었는지 확인한다. 두 오류 복구 방식은 오류가 발생한 메시지의 처리에서 차이를 보인다. 만약 N번째 메시지에 오류가 발생할 경우 Go-Back-N ARQ의 경우 현재 (N + 3)번째 메시지를 보내고 있어도 N번째 메시지부터 다시 보내야 한다. 반면에 같은 상황에서 Selective repeat ARQ를 사용할 경우에는 N번째 메시지만을 재전송하면 된다.

오류 복구 메커니즘의 변경으로 인한 변화를 살펴보기 위해 기지국과 단말 사이의 링크에서 에러를 발생시키면서 3개의 오류 복구 메커니즘을 비교하였다. 논문에서 설정한 시뮬레이션 시나리오에서 기지국과 인증서버 사이에서는 TCP가 신뢰성 있는 전송을 보장한다. 그러나 기지국과 단말 사이에서는 오류가 발생할 수 있다. 802.16 표준은 신뢰성 있는 전송을 위한 ARQ 메커니즘을 제공하지만 모든 연결에 대해 제공하지 않고 선택적으로 ARQ 메커니즘을 제공하기 때문이다. 802.16 링크에서 오류가 발생할 경우 오류 복구는 전적으로 EAP에 의존해야 한다.

(그림 7)은 에러 발생 시 오류 복구 메커니즘에 따른 EAP 성능의 변화를 보여준다. (그림 7)을 통해 확인할 수 있듯이 Stop-And-Wait 방식의 흐름 제어와 오류 복구를 사용하는 경우 프레임 오류율 (Frame Error Rate : FER) 값에 상관없이 가장 좋지 않은 성능을 보였다. sliding window를 사용하는 두 개의 오류복구 메커니즘은 FER이 1% 이하일 때는 거의 같은 성능을 보인다. 그러나 FER이 높아짐에 따라 성능의 차이가 발생하는 것을 볼 수 있다. 두 오류복구 메커니즘을 적용했을 때의 성능차는 FER가 높아질수록 커진다. FER이 3%일 때 Selective Repeat ARQ은 Stop and Wait에 비해 6% 성능 증가를 보이지만 FER가 7%로 높아지면 두 메커니즘 사이의 성능 차는 23%로 높아진다.

Go-Back-N ARQ의 경우에는 ACK를 받지 못한 메시지의 재전송 처리 이외에는 sliding window와 구현에 있어서 크게 차이점이 없다. 그러나 Selective repeat ARQ를 적용할 경우에는 한 가지 문제가 발생한다. 그 문제는 Selective repeat ARQ의 경우 오류가 난 메시지만을 따로 전송하기 때문에 메시지의 순서가 바뀔 수 있다는 것이다. EAP는 메시지의 순서가 바뀌지 않도록 보장하는 것은 전적으로 하위 계층에 의존하고 있다. 그러므로 Selective Repeat ARQ을 EAP에 적용하기 위해서는 순서가 바뀐 메시지를 재정렬 하는 것이 가능하도록 구현해야 한다. 이것은 EAP 메시지의 sequence 번호로 사용되는 identifier 필드를 사용하면 가능할 것이다.

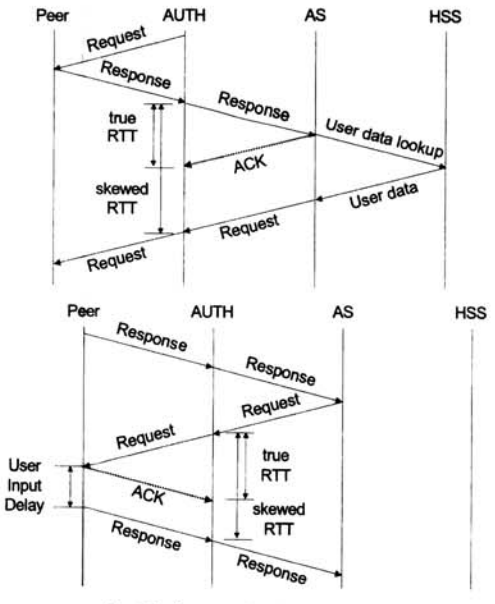


(그림 7) EAP 에러 복구 메커니즘 변경 시 성능 비교

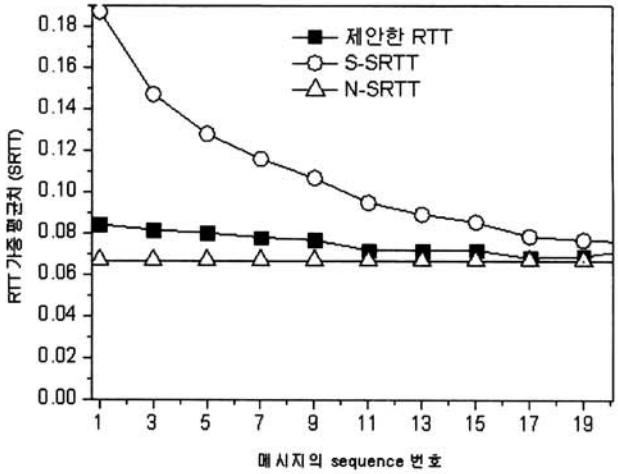
5.4 재전송

EAP의 재전송 메커니즘은 재전송 타이머 값을 동적으로 추정하도록 권고하고 있으며 동적 추정을 위한 알고리즘은 TCP의 재전송 타이머를 계산하는 알고리즘[9]을 그대로 사용한다. 재전송 타이머 값을 정확하게 추정하기 위해서는 RTT를 오차 없이 측정하는 것이 중요하다. 그러나 EAP 인증 과정에서는 정확한 RTT 측정을 방해하는 요소들을 가지고 있다. (그림 8)에서 볼 수 있듯이 첫 번째는 AS가 Peer로부터 Identity 응답 메시지를 받은 후에 사용자의 정보와 비밀 정보를 얻기 위해 다른 네트워크 개체에 접속하는 경우가 있다는 것이다. 예를 들어 3G에서는 인증을 위한 사용자의 정보를 홈 네트워크에 있는 HSS에서 얻어온다. 이 경우 사용자 정보를 얻기 위해 HSS와 메시지를 주고 받는 시간만큼 RTT 측정에 오차가 발생한다. 두 번째는 사용자의 입력이다. EAP가 지원하는 인증 방식 중에는 CHAP과 같이 인증을 위해 사용자의 입력이 필요한 경우가 있다. 이 때 측정된 RTT는 네트워크 특성보다는 사용자의 응답 시간에 의해 결정되므로 실제 RTT와는 오차가 있게 된다. 수집되는 RTT 샘플의 수가 많을 경우에는 이 오차들은 전체 성능에 크게 영향을 끼치지 않지만 일반적으로 인증을 위해 교환되는 메시지의 수는 많지 않다. 특히 앞에서 측정된 RTT 값에 더 높은 가중치를 두어 계산하는 재전송 타이머 추정 알고리즘의 특징 때문에 인증 초기에 전송되는 Identity 메시지로 인한 오차는 인증 과정 전체에 영향을 미칠 수 있다. 이 문제를 해결하기 위해 우리는 AS가 Peer로부터 Identity 메시지를 받은 직후에 추가적인 확인 응답을 보내는 방법을 고안하였다. Peer는 확인 응답을 통해 RTT를 측정하며 후에 전송되는 실제 응답 메시지를 받는 시간은 무시된다.

제안된 방법을 검증하기 위해 시뮬레이션의 설정을 변경하였다. 먼저 EAP-TLS의 메시지 크기를 수정해 총 Peer에



(그림 8) RTT 측정 오차의 예



(그림 9) SRTT의 오차와 제안된 메커니즘의 오차 보정 효과

서 20개의 메시지를 전송하게 하였다. 그리고 20개의 메시지 각각에 대해 동적 추정 알고리즘에 의해 계산된 RTT 가중 평균치 (Smoothed RTT : SRTT) 값을 기록하였다. 다음으로 AS가 Peer에게 보내는 Identity 응답 메시지의 전송을 임의로 50ms 지연되도록 수정한 후에 다시 시뮬레이션을 수행하여 20개의 SRTT 값을 기록하였다. 마지막으로 AS가 Peer로부터 Identity 요청 메시지를 받는 즉시 확인 응답을 보내고 Peer는 확인 응답을 받은 시간을 이용해 RTT 값을 추정하도록 시뮬레이션을 수정한 후 동일하게 20개의 SRTT 값을 기록하였다. (그림 9)는 Identity 응답 메시지 전송의 지연으로 인한 RTT 추정의 오차와 제안된 방법을 통해 오차를 완화할 수 있다는 것을 보여준다. RTT 샘플의 수가 증가함에 따라 오차가 발생한 skewed SRTT (S-SRTT)는 오차가 발생하지 않은 normal SRTT (N-SRTT)에 근접해 가지만 상당수의 샘플을 얻기 전에는 정확한 SRTT를 계산할 수 없다. 반면에 제안된 방법을 사용한 경우 Identity 메시지의 전송이 지연 되는 것에 상관없이 N-SRTT에 근접한 값을 계산하는 것을 볼 수 있다.

6. 결 론

인증 메커니즘에서 성능은 가장 중요하게 고려되는 문제는 아니다. 그러나 802.16이나 UMTS와 같은 네트워크에서 이동성과 핸드오버와 관련되면서 인증의 성능은 중요한 문제로 부각되고 있다. 본 논문에서는 EAP의 성능을 떨어뜨리는 요인들을 분석하고 그에 대한 해결책을 제시하였다. EAP의 성능 저하는 흐름 제어, 에러 복구, 재전송 메커니즘에서 발생한다. EAP는 stop-and-wait 방식의 흐름 제어와 에러 복구 메커니즘을 사용하기 때문에 메시지를 보낸 후 추가적인 메시지는 확인응답을 받은 후에 보낼 수 있다. 이 때문에 EAP는 "lock-step" 프로토콜이라고 불리며, EAP 메시지가 분할 될 경우에 분할된 메시지의 수 만 큼의 RTT가 추가적으로 소요되게 된다. 본 논문에서는 EAP의 성능 개선을 위해 sliding window 기반의 흐름제어 메커니즘을

적용을 제안하였다. 시뮬레이션 결과 EAP-TLS의 인증서 메시지가 8개로 분할된 경우에 sliding window를 적용하면 적용하지 않았을 때에 비해 53%의 성능 증가를 보였다.

Sliding window의 적용을 위해서는 오류 복구 메커니즘의 변경이 필요하다. 논문에서는 EAP에 적용하기 위한 오류 복구 메커니즘으로 Go-Back-N ARQ와 Selective Repeat ARQ를 선택하였다. 두 오류 복구 메커니즘을 시뮬레이션에 적용한 결과 EAP 메시지가 유실 되지 않을 경우에 두 메커니즘 사이에 차이는 없었다. 반면에 메시지 유실이 발생하기 시작하면서 Selective Repeat ARQ의 성능이 Go-Back-N ARQ에 비해 상대적으로 높아졌다. 메시지가 유실되는 확률이 높아짐에 따라 성능의 차이는 증가했으며 EAP의 메시지가 7%의 확률로 유실될 경우에 두 메커니즘 사이에는 23%의 성능 차이를 생기는 것을 확인하였다.

마지막으로 사용자 입력에 의한 지연이나 HSS와의 통신 때문에 발생하는 재전송 타이머의 동적 추정의 오차를 줄이기 위한 새로운 메커니즘을 제시하였다. 시뮬레이션 결과 이 메커니즘을 적용할 경우 재전송 타이머를 계산하기 위한 RTT 측정의 오류를 줄일 수 있는 것을 확인하였다.

참 고 문 헌

[1] IEEE std 802.1x-2001, "Port-Based Network Access Control," June, 2001.

[2] IEEE std 802.11i/D4.1, "Wireless Medium Access control (MAC) and Phsical Layer (PHY) Specification: Medium Access Control (MAC) Security Enhancement," July, 2003.

[3] Avesh K. Agarwal and Wenye Wang, "Measuring Performance Impact of Security Protocols in Wireless Local Area Networks," in Proceedings of the Second Annual International Conference on Broadband Networks, Oct., 2005.

[4] G. Kambourakis et al., "Advanced SSL/TLS-based authentication for secure WLAN-3G interworking," IEE Proceedings Communications, Vol.151, Iss.5, Oct., 2004.

[5] Yao Zhao, Chuang Lin and Hao Yin, "Security Authentication of 3G-WLAN Interworking," in Proceedings of the 20th International Conference on Advanced Information Netowrking and Applications, Apr., 2006.

[6] B. Aboba et al., "Extensible Authentication Protocol (EAP)," RFC 3748, June, 2004.

[7] "Network Simulator 2 homepage", http://www.isi.edu/nsnam/ns/index.html

[8] IEEE std 802.16-2004, "IEEE Standard for Local and Metropolintan Area Networks Part 16: Air Interface for Fixed Broadband Wireles Access Systems," Oct., 2004.

[9] V. Paxson, M. Allman, "Computing TCP's Retransmission Timer," RFC 2988.

[10] V. Kamath, A. Palekar, "Microsoft EAP CHAP Extensions", IETF draft-kamath-pppext-eap-mschapv2-01.txt, Apr., 2004.

차 은 철

e-mail : iris1212@ece.skku.ac.kr

2005년 성균관대학교 정보통신공학부(공학사)

2007년 성균관대학교 정보통신공학부(공학석사)

관심분야 : 인터넷보안, 무선랜보안, VoIP 등

한 찬 규

e-mail : hedwig@ece.skku.ac.kr

2006년 성균관대학교 컴퓨터공학과(공학사)

2008년 성균관대학교 전자전기컴퓨터공학과(공학석사)

현 재 성균관대학교 휴대폰학과 박사과정

관심분야 : 인터넷보안, 모바일 인증 등

최 형 기

e-mail : hkchoi@ece.skku.ac.kr

1992년 성균관대학교 전자공학과(공학사)

1996년 Polytechnique University 전기전자(공학석사)

2001년 Georgia Institute of Technology 전기전자(공학박사)

2001년~2004년 미국 Lancope. Inc. 연구원

2004년~2006년 성균관대학교 정보통신공학부 전임강사

2006년~현 재 성균관대학교 정보통신공학부 조교수

관심분야 : 인터넷보안, 모바일 커뮤니케이션 등