IEEE 802.1X EAP-TLS에 따른 인증서 기반의 무선 랜 접속 인증 시스템 분석과 구현

박정현, 김원규, 김석우, 서창호*
한세대학교 정보통신학과, 공주대학교*

The Analysis and Implementation of Wireless LAN Connection Authentication system Based on IEEE 802.1X EAP-TLS

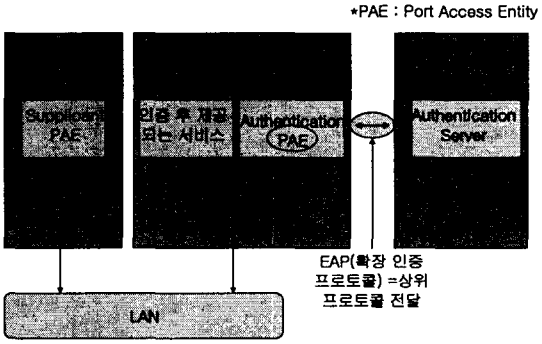
Jung-Hyun Park, Won-Kyu Kim, Seok-Woo Kim, Chang-Ho Seo*
Dep. of Information & Communication Hansei Univ. , Kong Ju Univ.*

요 약

최근 들어 무선 랜에 대한 사람들의 인식이 높아지고, KT의 NESPOT과 같은 공중망 사업자들에 의한 핫스팟 서비스가 제공되면서 무선 랜의 수요 또한 급증하고 있다. 무선 랜의 사용자가 증가하면서 무선 랜 보안의 중요성 역시 증가하고 있으며, 실제로 무선 랜에서의 안전한 네트워킹을 위하여 여러 단체들이 다양한 방향에서 연구를 진행 중에 있다. 그 한 예가 802.1X인데, 이것은 인증 서버를 따로 두어 AP를 통해 네트워크에 접속하려는 사용자들을 인증하여 주는 것이다. 이 논문에서는 802.1X 인증 방법 중 X.509 기반의 인증서를 사용하여 서버와 클라이언트간의 상호 인증을 가능하게 하여 주는 EAP-TLS 환경을 분석하고, LINUX 환경에서 공개 소스로 구축하여, 실제로 무선 랜을 사용하는 환경에 적용하는 과정을 기술한다.

I. 서론

본 논문에서는 RADIUS 서버와 Access Point, Client로 이루어진 IEEE 802.11 표준의 무선 네트워크에서 보안성 강화를 위하여 인증서를 통한 RADIUS 서버와 Client의 상호 인증을 통해 세션별 보안을 유지하는 802.1X EAP-TLS를 LINUX 환경에서 Open Source를 이용하여 구현하는 과정을 다룬다. EAP-TLS는 802.1X 구조에서 동작하는 인증 방법의 한 종류이다. 802.1X란 보안을 위해 사용자 인증이 요구 될 때 RADIUS 등과 같은 인증 서버를 이용해 사용자 인증을 받고, 인증 받은 사용자만이 네트워크에 연결되어 인터넷 등의 사용이 가능하게 해 주는 port 기반 인증의 표준이다. 이것은 유/무선에 관계없이 사용 될 수 있으나 본 논문에서는 상대적으로 보안이 취약한 무선 네트워크에서의 사용자 인증에 FreeRADIUS를 통한 802.1X 인증을 사용하게 될 것이다.



[그림 1] 802.1X 포트 기반 인증

802.1X 인증의 한 방법인 EAP-TLS는 위에서도 언급했듯이 서버와 클라이언트가 인증서를 통

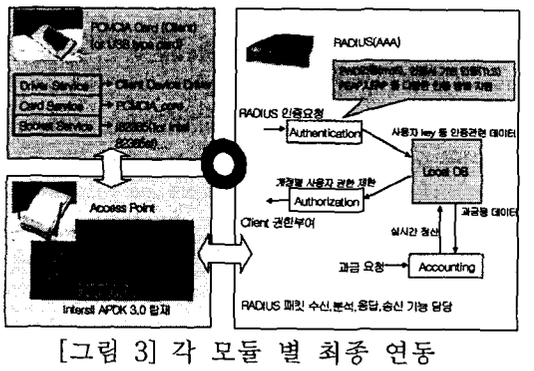
한 상호 인증을 통하여 각각의 세션에 대한 인증을 수행하기 때문에 802.1X의 EAP 방법 중 가장 보안성이 뛰어나다고 할 수 있지만, 각각의 client가 모두 인증서를 보유해야 하므로 현실적으로 적용이 어려운 단점이 있다. 이에 대한 대안으로 EAP-TTLS나 PEAP와 같이 서버 측 인증서만으로 동작하는 EAP 기술이 개발, 적용되고 있는 상황이나 보안성 자체만으로 비교한다면 EAP-TLS가 가장 높은 수준의 보안을 제공하며, 모두 EAP-TLS에 기반 한 기술이기 때문에 EAP-TLS의 구현은 충분히 의미 있는 일이라 할 수 있다.

Method	Description	Authentication Security	WEP Key
EAP-MD5	Challenge-based password	One way authentication	X
EAP-SRP	SRP-SHA1	Mutual authentication	Generate
EAP-TLS	Certificate-based two way authentication	Mutual authentication	Generate
EAP-TTLS	Server authentication via certificates; client via other method	Mutual authentication	Generate
PEAP	Server authentication via certificates; client via EAP	Mutual authentication	Generate

[그림 2] EAP 기술의 종류

Ⅱ. EAP-TLS 구현을 위한 환경

802.1X 기반의 EAP-TLS 구현을 위해서는 AAA server(RADIUS), Access Point, Client 3 부분의 setup과 연동이 필요하다. 그림 3은 각 모듈의 최종 연동을 도식화 한 것이다.



[그림 3] 각 모듈 별 최종 연동

본 논문을 위해 구현한 Test-bed의 시스템 별 S/W H/W는 다음과 같다.

1) RADIUS server(유선)

H/W : Pentium 733 Mhz, 390RAM, desktop

OS : Red Hat Linux 8.0(kernel-2.4.18-14)

S/W : FreeRADIUS-0.9.0-pre3.tar.gz

Openssl 0.9.7b.tar.gz

Openssl-SNAP-20030625.tar.gz

Openssl 0.9.7-beta.tar.gz

2) Access Point (유/무선)

H/W : howap5100 (802.11b/1x/WPA 기능지원)

F/W : Intersil APDK 3.0 사용

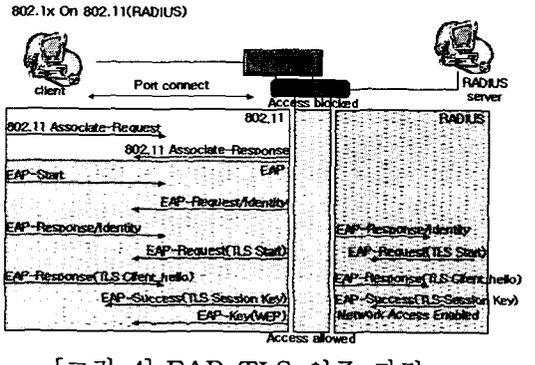
3) Client (무선)

H/W : Linksys instant wireless(USB)

OS : Windows XP professional

S/W : Windows XP service pack 1 설치(권고)

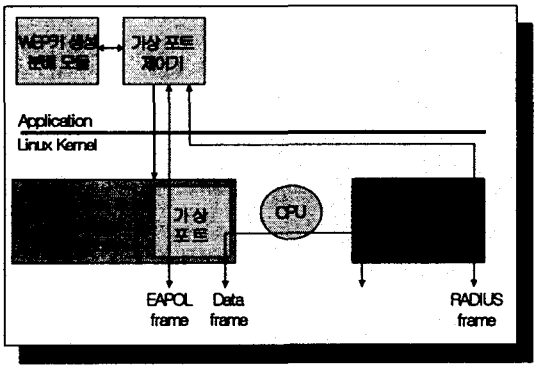
이 3가지의 요소들은 그림 4와 같은 순서로 EAP-TLS 인증을 진행하게 된다.



[그림 4] EAP-TLS 인증 과정

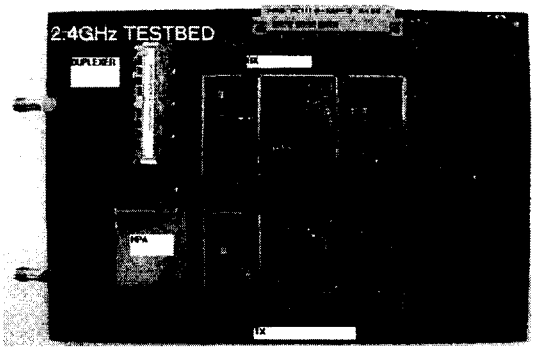
여기서 가장 주의해야 할 부분은 RADIUS server를 세팅함에 있어서 서로 다른 3개 버전의 Openssl이 필요하다는 것이다. 안정(Stable) 버전의 Openssl은 FreeRADIUS를 설치하는 데 있어서 대부분의 모듈과 연동되어져 사용된다. SNAP 버전의 Openssl(SANP-20030625)을 사용하는 이유는 EAP-TLS 모듈이 정상적으로 설치되어 동작하도록 하기 위해서이다. 안정 버전의 헤더 파일에는 SSL_set_msg_callback 등과 같이 EAP-TLS에서 사용되는 구조체들이 선언되어 있지 않으므로 설치 시 TLS 모듈 생성 부분의 makefile의 경

로를 안정 버전이 아닌 SNAP 버전의 설치 디렉토리로 변경하여 주어야 한다. 또한 Windows XP를 사용하는 Client와 연동하여 EAP-TLS를 사용하기 위해서는 RADIUS에서 인증서를 생성 할 때 확장키(enhanced OID's)를 사용하여 Client 인증을 위한 인증서를 생성해야 하는데 안정 버전의 Openssl에서는 확장키 사용을 지원하지 않으므로 beta 버전을 사용하여 인증서를 생성하여야 한다. TLS 모듈 생성과 확장키 사용에만 다른 버전을 사용하는 이유는 그 이외의 부분에 있어서 여러모로 검증 된 안정 버전을 사용하는 것이 비교적 안전하기 때문이다. 단지 TLS의 테스트 용도로만 사용할 목적으로 구현한다면 SNAP과 beta 버전의 Openssl만 설치해도 괜찮겠지만 TLS가 RADIUS 서버의 많은 기능 중 하나로 구현된다면 Openssl 안정 버전의 설치를 권장한다. 또한 AP의 측면에서 보면 반드시 802.1X와 EAP-TLS가 지원이 되는 것이라야 하는데, 구현을 위하여 사용된 AP는 국내 기업인 (주)하우텔이 만든 howap 5100이다. AP의 모듈 구성도는 그림 5와 같다.



[그림 5] howap 5100 모듈 구성도

Intersil의 APDK 3.0을 펌웨어로 탑재하고 있으며, 802.11b 규격에 1X와 WPA가 기본 지원되는 비교적 저가의 AP로서 EAP-TLS 연동 시에는 128bit의 WEP키를 AP가 자동 공급하도록 세팅한 상태에서 TLS 연동을 진행하였다. 그림6은 AP의 내부 모습을 나타낸 것으로 Intersil의 PRISM2 계열의 칩을 RF에 사용하였다.



[그림 6] howap 5100 내부 회로도

Ⅲ. EAP-TLS의 구현

1. S/W 설치

OPENSSL (stable, SNAP, beta)

위에서 언급했던 3개의 버전의 OPENSSL을 다운받아 적당한 폴더에 압축을 푼다.

tar -xvzf openssl -<version>.tar.gz

OPENSSL 0.9.7b

압축을 푼 openssl-0.9.7b 폴더로 들어가서 현재 기본적으로 설치되어 있는 openssl의 경로와 맞추어 다음과 같이 설치한다.

./config --prefix=/기본dir/openssl shared

./make로 에러 없이 make 되면 **./make install**을 통해 설치를 마무리한다. (Redhat 8.0의 기본 dir은 **/usr** 이었다.)

OPENSSL SNAP-20030625

압축을 푼 openssl-SANP-0030625 소스폴더로 들어간다. SNAP 버전의 설치 경로를 다음과 같이 지정하고 잘 기억해 두도록 하자.

./config --prefix=/usr/local/openssl shared

./make와 **./make install**로 설치가 끝나면 **cd /usr/local/openssl/lib** 로 가서 **libssl.so** 와 **libssl.so.0**에 의해 심볼릭 링크된 **libssl.so.0.9.8**과 **libcrypto.so**와 **libcrypto.so.0**에 의해 심볼릭 링크 된 **libcrypto.so.0.9.8**이 제대로 생성되었는지 확인한다.

OPENSSL 0.9.7-beta3

압축을 푼 openssl-0.9.7-beta3 소스폴더로 들어간다. SNAP 버전의 설치 경로를 다음과 같이 지정하고 잘 기억해 두도록 하자.

./config -- prefix = /usr/local/openssl -certgen shared

./make와 **./make install**로 설치가 끝나면 **cd /usr/local/openssl-certgen/lib** 로 가서 **libssl.so** 와 **libssl.so.0** 에 의해 심볼릭 링크된 **libssl.so.0.9.7** 과 **libcrypto.so**와 **libcrypto.so.0**에 의해 심볼릭 링크 된 **libcrypto.so.0.9.7**이 제대로 생성되었는지 확인한다.

FREERADIUS 0.9.0-pre3

tat -xvzf freeradius-0.9.0-pre3.tar.gz

압축을 푼 폴더로 들어가서 **./configure--sysconfdir=/etc**를 실행한다.

cd /src/modules/rlm_eap/types/rlm_eap_tls

Makefile에서 SNAP 버전이 설치된 곳으로 경로를 바꾸어 준 후(**/usr/local/openssl**) 소스의 root 부분으로 돌아와 **./make**와 **./make install**을 실행한다. make 할 때 수정한 rlm_eap_tls 부분에서 경고 메시지나 에러가 났다면 SNAP 버전 openssl을 설치했던 경로의 include나 lib dir로 경로 지정이 올바르게 되었는지 다시 확인하고 Makefile을 수정하여 준다. 차후 연동 시 이 부분에서 SSL_set_msg_callback 등의 에러 메시지가 보이면서 RADIUS의 작동이 중단되거나 한다면 openssl의 EAP-TLS 모듈 경로 설정이 SNAP 버전으로 되지 않은 것으로 예측 할 수 있다.

2. 인증서 생성

Windows XP Client를 인증하기 위한 인증서 생성을 위해선 openssl-beta 버전을 사용한다. (OID를 이용한 client인증) 기존에 설치된 안정 버전의 openssl 에서는 Windows XP client가 필요로 하는 만큼의 확장된 Attribute를 가진(확장키 : enhanced OID) 인증서를 생성 할 수 없고, 확장키를 안정버전으로 생성하려 한다면 "header too long"이라는 에러 메시지가 출력 될 것이다. SNAP 버전은 확장키를 지원하기는 하지만 인증서를 생성하는 과정 중에 PKCS12 인코딩/ 디코딩을 할 때 에러가 나서 인증서를 성공적으로 생성할 수 없다. 따라서 인증서를 생성 할 때 openssl-beta 버전을 사용하여야 하는 것이다.

cd /usr/local/openssl-certgen (beta 설치dir)

cd ./ssl로 이동하여 **openssl.cnf** 파일을 수정한다. 파일 중 **./demoCA** 가 생성되는 경로는 수정하지 말고 **req_distinguished_name** 부분을 구현자의 정보와 일치하게 작성하여 두면 차후에 인증서 생성 할 때 일일이 내용을 작성하는 번거로움을 덜 수 있다. openssl.cnf 파일은 인증서 생성 script가 인증서를 생성하기 위해 사용된다.

인증서 생성 script

인증서 생성을 위하여 rootCA와 client, server 각각에 대한 script를 만들어야 하고(**cert_script**), 또한 OID 확장을 위하여 또 다른 script가 추가로 필요하다.(**xpextensions**) 이전에 언급했듯이 OID 확장을 위해 openssl-beta 버전을 사용해야 하는데, 인증서를 생성하는 script(cert_script) 내의 SSL 경로를 beta 버전이 설치되었던 경로와 일치시켜 주어야 사용하고자 하는 올바른 인증서의 생성이 가능하게 된다. (/usr/local/openssl-certgen) 인증서 생성 시 필요한 패스워드는 default로 whatever로 되어있지만 그 부분을 수정하여 관리자만이 알고 있는 암호로 바꾸어 주어야 한다. 차후에 패스워드가 필요하니 잘 기억해두자. 인증서 생성을 위한 cert_script와 xpextension 두 가지의 script는 지면 관계상 생략하니 후반부의 **참고 문헌[1]**을 참조하길 바란다.

인증서 생성과 용도

cert_script와 xpextension의 두개의 스크립트가 만들어졌다면 다음으로 예제로 사용 가능한, 키가 미리 생성되어 있는 cert.tgz 파일의 압축을 풀고 **(tar -xvzf cert.tgz)** 압축 풀린 결과물을 구분이 용이하도록 /etc/1x로 이동시켜 사용한다.**(mv cert /etc/1x)** 하지만, 이 예제 파일에 있는 인증서들은 이미 사용 기한이 지났으므로 사용이 불가능하고 **dh** 파일과 **random** 파일만 제외하고 다른 인증서들은 위의 cert_script를 이용하여 다시 생성해 주어야 한다. cert.tgz 파일의 출처는 본 논문의 **참고 문헌[2]**를 참고하면 된다. 인증서를 생성하는 방법은 압축을 풀고 이동시킨 /etc/1x 디렉토리에 위에서 만든 cert_script와 xpextension 의 두 개의 스크립트를 복사하고 실행이 가능하도록 모드를 700으로 바꾸어 준 후(**chmod 700 cert_script**) 스크립트를 실행시키면(**./cert_script**) 인증서 내부의 세부 내용 입력을 통한 rootCA와 client, server 각각에 대한 인증서를 만들 수 있게 되며, 생성된 인증서에 대한 확인은 ls -al 명령을 통해 파일이 생성 된 날짜로 파악이 가능하다. 인증서의 생성의 끝나면 여러 가지의 파일들이 생성되는데, 대부분은 rootCA와 서버 내에서 사용되는 것들이고, client에게 인증을 위해 설치하여야 하

는 파일은 **root.der**과 **cert-clt.p12** 두 가지의 파일이다.

3. FreeRADIUS 설정

FreeRADIUS의 설정(configuration) 은 4개 파일 정도를 수정하는 것으로 기본적인 설정이 가능하므로 비교적 단순하다고 볼 수도 있다. 그렇지만 기본 설정이 아니라 LDAP이나 PAM 모듈을 사용한다든지 ploxy나 mysql 등을 사용하려고 할 때에는 해당 항목에 대한 설정 파일들을 더 수정하여 주어야 한다. 이것에 대해서는 FreeRADIUS 내의 설정 파일들을 참고하면 되겠다. RADIUS를 처음 설치 할 때 **--sysconfig=/etc** 라는 인자를 주고 설정 파일을 수행하였으므로 설정 부분에서 수정해줘야 할 파일들은 **/etc/raddb** 내에 위치하게 되며, 그것은 **client, client.conf, radiusd.conf, users** 이렇게 총 4가지이다.

client - RADIUS와 연동할 AP의 IP 주소와 secret 공유 키를 설정

client.conf - client 등록 AP와 동일한 IP, secret과 ssid에 해당하는 shortname을 추가 등록

radiusd.conf - RADIUS의 인증 방법, DB 연동 등 대부분의 설정이 여기에서 이루어지며, TLS를 지원하기 위해서 TLS 해당 부분의 주석을 풀고 인증서가 위치한 부분의 경로와 일치시킨다.

users - 실제 인증하게 될 windows XP의 client를 등록시킨다. 주의할 점은 client 인증서의 common name과 users에 등록된 user name이 동일해야 인증서의 대조가 가능하다. radtest를 통하여 RADIUS 서버의 작동 유무를 local에서 확인하는 경우는 인증 방식이 local이지만 EAP-TLS를 사용하기 위해서는 사용자의 인증 방식을 EAP바꿔 주어야 한다. 그밖에 세부적인 세팅 예는 **참고문헌[1]**을 참조하도록 한다.

4. FreeRADIUS 실행 및 Test

EAP-TLS를 구현하기 위해 지금까지의 인스톨과 설정 부분들이 분명히 정리된 상황이라 가정하고 이번 단계를 진행한다. RADIUS를 실행시키기 위해서는 또 하나의 script를 만들어야 하는데, 이 script(**run-radius**)의 역할은 RADIUS가 처음 동작할 때 사용되는 **SSL library들을 SNAP 버전의 openssl로부터 추가로 사용**할 수 있도록 한다.

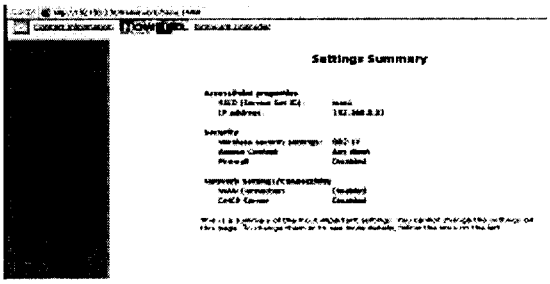
LD_LIBRARY_PATH=/usr/local/openssl/lib

LDPRELOAD=/usr/local/openssl/lib/libcrypto.so

위의 두 부분을 스크립트에 추가해서 RADIUS 데몬을 실행시켜 주면 된다. 작성한 script를 RADIUS 관련 실행 파일이 모여 있는 **/usr/local/ sbin** 폴더로 복사하고, 파일의 실행이 가능하도록 권한을 700 으로 바꾸어 준다. (**./chmod 700 run-radiusd**) 이렇게 하면 RADIUS를 실행하기 위한 모든 사전 준비는 끝난 셈이다. 마지막으로 **/usr/local/sbin** 폴더로 이동하여 **./run-radiusd -X -A** (디버깅 모드로 실행)을 실행시킨다.

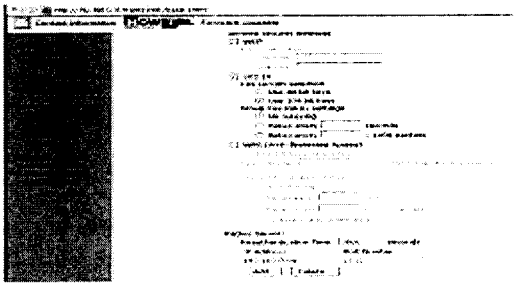
5. Access Point 설정

AP는 802.1X port-base 인증과 EAP-TLS가 지원되는 제품이어야 한다. 본 구현에 사용된 AP는 intersil의 APDK 3.0 버전을 사용한 802.11b 기반의 1X/WPA 기능을 갖는 (주)하우텔의 howap 5100 이다. web interface를 통하여 설정을 하며, 그 화면은 아래와 같다.



[그림 7] howap 5100 web interface

위의 그림은 전체 부분에 대한 종합이며, 왼쪽의 wireless security setting 부분을 클릭하면 사용할 RADIUS 서버에 대한 비밀 공유키의 등록이나 WEP 설정 등 무선 랜의 보안 설정을 할 수 있다.



[그림 8] howap 5100의 보안 설정

설정을 자세히 살펴보면 WEP을 자동으로 하되 802.1X 기능에 의한 104bit(IV 포함 128bit) WEP을 자동 할당하여 사용하는 세팅을 통해 EAP-

TLS가 성공적으로 이루어 졌다. 또한 RADIUS 서버 세팅하는 부분에서는 재 인증시간을 3600초 즉 1시간으로 제한하였으며, RADIUS 서버의 IP 주소와 서버와의 통신에 사용되는 UDP포트 번호, secret 공유 키 등을 등록하여 추가하거나, 이미 등록된 서버의 삭제가 가능하도록 되어 있다. 다른 세팅으로도 TLS 통신이 가능할 수 있겠지만, 현재는 위와 같은 세팅으로 TLS가 성공적으로 동작했다. 또한 처음 등록된 RADIUS를 primary로 인식하여 RADIUS가 여러 개 등록된 경우 순서대로 패킷을 보내는 것을 확인 할 수 있었는데, 이것은 여러 개의 RADIUS 서버에 같은 비밀키를 가진 AP와 client가 등록되어 있으면 어느 한 서버가 다운되더라도 다른 RADIUS 서버에 의해 재인증을 받을 수 있는 상호 보완적인 기능이 가능하다는 것을 알 수 있었다.

6. Windows XP 사용자 설정

사실 이 부분은 위에서 인증서를 만들고 RADIUS 서버의 run-radiusd script를 실행시키기 전에 선행되어야 하는 부분이다. 위의 인증서 생성 부분에도 언급했듯이 Windows XP client에 RADIUS 에서 생성한 인증서 중 **root.der (rootCA 인증서) 와 cert-clt.p12 (client 인증서)** 두 개를 ftp 등으로 다운받아 설치하여 주면 된다. 자세한 순서와 설치 방법은 **참고 문헌 [3]**을 참고하도록 한다.

Ⅳ.결론 및 향후 연구 방향

본 논문에서는 최근 네트워킹 분야의 새로운 화두로 떠오르고 있는 무선 네트워크의 보안성을 증가시키기 위하여 X.509 인증서를 사용하여 서버와 클라이언트가 상호 인증하는 802.1X 기반의 EAP-TLS 분석과 함께, 리눅스용 공개 소스를 이용하여 실제로 구축하는 내용을 다루어 보았다. 지면 관계상 구현 과정을 자세히 언급하지 못한 부분이 있지만 참고 문헌을 참조하면서 이 논문에서 기술한 대로 구현한다면 아마도 EAP-TLS가 적용된 안전한 무선 네트워크를 기존의 PC를 이용하여 만들어 내는 일도 그리 어려운 것만은 아닐 것이다. 또한 실제로 구축되어 있는 무선 네트워크 상에서 사용한다면 고가의 무선 랜 인증 서버 못지 않게 안전하고 편리하게 사용 할 수 있을 것이다. 또한 802.1X 기반의 EAP 레벨에서 보안성이 더욱 강화된 기술을 개발하기 위해서는 EAP-TLS와 같은 기존의 기술들에 대한 세부적인 이해와 분석 또한 동시에 이루어져야 하는 것이니 만큼 본 연구 결과가 관련 연구 분야 및 향후 연구 목표에 자그마한 초석이 될 수 있기를 기대한다.

참고 문헌

[1] Raymond McKay's FreeRADIUS EAP/TLS -WinXP HOWTO
http://www.impossiblereflex.com/8021x/eap-tls-HOWTO.htm

[2] Adam Sulmicki's HOWTO on EAP/TLS authentication between FreeRADIUS and XSupplicant
http://www.missl.cs.umd.edu/wireless/eaptls

[3] Ken Roser's HOWTO: EAP//TLS Setup for FreeRADIIUS and Windows XP Supplliicant
http://www.freeradius.org/doc/EAPTLS.pdf

[4] Jungho Park's FreeRADIUS Server Installation Manual
http://palace.xwow.net

[5] Keung Hee Oh's Access Control of Wireless LAN Access Point Based on IEEE 802.1X - 2002 CISC

[6] Joseph Davies, Microsoft Corporation, Enterprise Deployment of IEEE 802.11 Using Windows XP and Windows 2000 Internet Authentication Service
http://www.microsoft.comWindowsXP/pro/techinfo/deplovment/wireless/80211corp.doc