사용자에 대한 강력한 인증과 SA 협상을 위한 OTIAS 시스템 개발에 대한 연구

김희승*, 한영주*, 정태명***
*성균관대학교 컴퓨터공학과
***성균관대학교 정보통신공학부
e-mail : hskim@imtl.skku.ac.kr, yjhan@imtl.skku.ac.kr, and tmchung@ece.skku.ac.kr

A Study on Development of OTIAS System for strong Authentication and SA negotiation of user

Hee-Seung Kim*, Young-ju Han*, and Tai-Myung Chung**
*Dept. of Computer Engineering, Sungkyunkwan University
**School of Information & Communication Engineering, Sungkyunkwan University

요 약

다양한 망 접속 기술이 개발되고 다양한 컨텐츠들이 온라인 환경으로 옮겨가게 되면서, 네트워크 접근 제어 기술과 VPN 에 대한 요구사항이 생겨났다. 무선랜 기반 구조 상에서는 포트 기반의 네트워크 접근 제어 기술이 나오게 되었지만, 이 접근 제어 기술은 VPN 에 대한 고려가 되지 않았다. PANA 는 기반 구조에 무관하게 네트워크 계층 위에서 작동하고 사용자를 인증한 후, VPN 통신을 위한 SA 를 협상한다. 본 논문에서 제안하는 OTIAS 시스템은 PANA 의 구동원리를 참조하고, EAP-TLS-RADIUS 연동 프로토콜을 수정, 보완하여 기반 구조에 무관하게 네트워크 계층 위에서 작동하며, 인증서를 이용하여 상호 인증을 수행하고 이후 IPSec 터널링을 위한 SA 를 협상을 가능하게 한다.

1. 서론

WLAN, xDSL, Mobile IP, Wibro, Ethernet, Dial-up PPP 등 다양한 망 접속 기술이 개발되면서 기존에 많이 사용하는 포트 기반의 네트워크 접근 제어 기술보다 새로운 IP 기반의 네트워크 접근 제어 기술을 필요로 하게 되었다. 또한 다양한 통신인프라의 발전이 거듭하게 됨에 따라 많은 서비스 컨텐츠들이 새롭게 탄생하게 되었으며 기존의 오프라인에서 운용되던 대다수의 서비스 컨텐츠들이 온라인 환경으로 옮겨가게 되는 일이 잦아졌다. 이에 서비스 컨텐츠에 대한 클라이언트의 접근 제어 뿐 아니라 통신 채널의 기밀성과 무결성을 보장해야 하는 Remote Access 형태의 VPN 구성이 요구사항으로 추가되었다. 따라서 새롭게 제안되어야 하는 IP 기반의 네트워크 접근 제어 기술은 클라이언트와 게이트웨이 (또는 Authenticator, Application Server) 간에 VPN 을 형성하기 위한 SA 협상 과정을 포함하는 프레임워크가 되어야 한다. 이러한 연구는 IETF 워킹 그룹인 PANA 그룹에서 처음 시도 되었고, 현재 PANA 프로토콜을 드레프트로 제안되고 있다[1]. 본 논문에서는 PANA 프로토콜과 거의 비슷하고, 기존의 EAP-TLS 을 수정하여, IP 기반의 네트워크 접근 제어가 가능하며, IPSec 터널링을 형성하기 위한 SA 협상을 간단하게 수행할 수 있는 OTIAS (One Touch Integrated Authentication Service for IPv6) 시스템에 제시하고자 한다. 본 논문의 2 장에서는 기존에 네트워크 접근 제어에 사용된 포트 기반의 네트워크 접근 제어 방식과 새로이 제안되고 있는 PANA(Protocol for Carrying Authentication for Network Access)에 대해 알아보고, 3 장에서는 본 논문에서 제안하는 OTIAS 시스템의 세부 요소들과 프로토콜에 대해 설명한다. 마지막으로 4 장에서는 결론과 향후

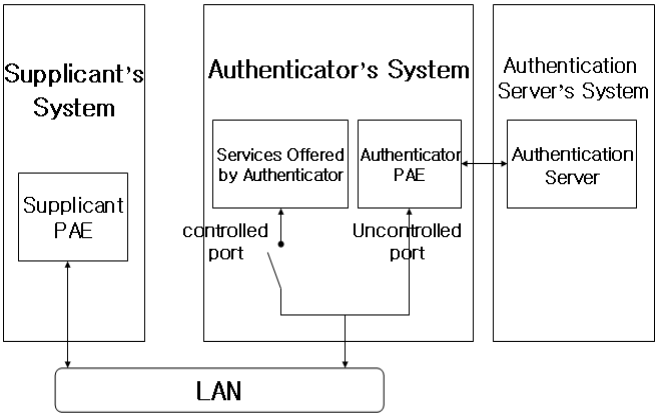
본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성・지원사업의 연구결과로 수행되었음

연구 계획에 대해 기술한다.

2. 관련연구

2.1. 포트 기반의 네트워크 접근 제어

IEEE 802.1X 에서는 무선랜 기반구조 상의 임의의 사용자가 액세스 네트워크에 접근하려고 할 때, 포트를 이용하여 브리지 또는 무선 AP 에서 인증을 수행한 다음에 망에 접근할 수 있도록 하는 포트 기반의 네트워크 접근 제어 메커니즘을 제안하고 있다. 이 메커니즘은 서비스를 제공하고자 하는 포트에 대하여 인증을 수행하는 Authenticator, authenticator 에서 제공하는 포트의 인증을 받고자 하는 Supplicant, Supplicant 의 신분을 인증하여 Authenticator 가 서비스를 제공할 수 있도록 알려주는 Authentication Server 로 구성된다. 다음 [그림 1]은 포트 기반의 네트워크 접근 제어 메커니즘에서 각 시스템의 역할을 보여준다.



[그림 1] 포트 기반 네트워크 접근 제어 메커니즘

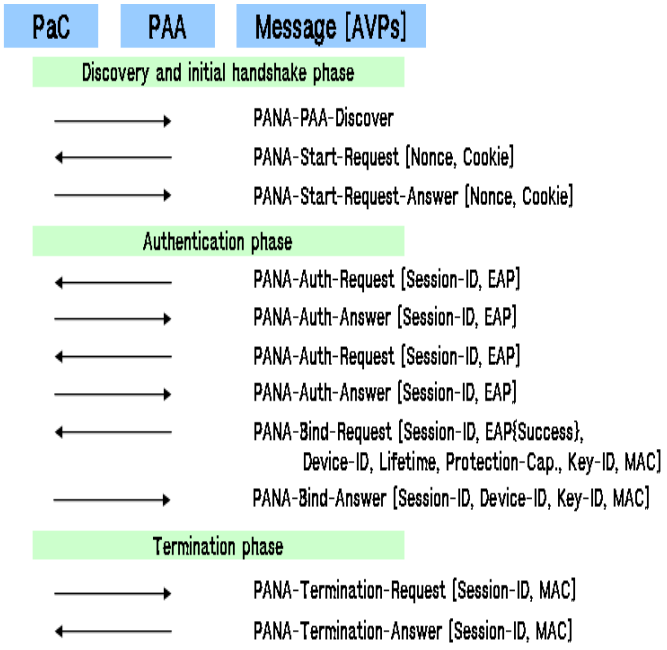
Authenticator 는 제어포트와 비제어포트를 가지고 있으며, Supplicant 는 서비스를 제공받기 위해 Uncontrolled Port 를 통해 사용자의 인증 정보를 전송하여 Authentication Server 에서 인증을 수행한다. 인증에 성공하면 Controlled Port 를 통해 망 접속이 허용된다. 하지만 이 메커니즘은 점 대점 연결 특성을 가진 랜 포트에 연결된 장치에게만 서비스를 제공하고, 인증 이후 controlled port 를 통해 통신하는 데이터에 대한 비밀성, 무결성은 책임지지 않는 단점이 있다[2].

2.2. PANA

PANA 는 포트 기반 네트워크 접근 제어 메커니즘과 마찬가지로 안전한 네트워크 접근 서비스를 제공하기 위해 클라이언트와 접근 네트워크 간에 인증 정보를 주고 받으면서 인증과 권한부여에 기반한 접근 제어를 수행하는 어플리케이션 레벨의 프로토콜을 포함하는 프레임워크이다. 하지만 포트 기반 네트워크 접근 제어 메커니즘과는 다르게 네트워크 계층 위에서 동작하기 때문에 접속 환경에 무관하게 동작할 수 있고, 접근 제어 성공 후에 SA 협상을 수행하여 VPN 설정의 요건을 만족시켜준다. 이를 위해 PANA 는 PaC (PANA Client), PAA (PANA Authentication Agent)로 구성되어 있고 이들 간에 인증 정보와 SA 협상 정보를 주고 받는다. PaC 는 프로토콜의 클라이언트 측면으로 호스트 장치에 위치해 있으면서 인증을 위해 인증서를 전달하는 역할을 한다. PAA 는 액세스 네트워크 측에 위치해 있으면서 PaC 가 전달한 인증서를 확인하고 네트워크 접근 서비스를 PaC 에게 부여하는 역할을 한다.

PANA 는 다양한 EAP 메소드를 이용하여 PaC 를 인증하게 되고, 이를 위해 크게 세 단계의 과정을 거치게 된다. 첫 번째 단계는 "발견과 초기 연결" 단계로 PaC 는 연결된 네트워크의 PAA 를 찾기 위해 PANA-PAA-Discover 메시지를 멀티캐스팅하고, 이 메시지를 받은 PAA 는 상태 유지를 하지 않고 세션 유지를 하기 위해 쿠키와 함께 "PANA-Start-Request 메시지를 PaC 에게 전송하여 초기 연결을 설정한다. 두 번째 단계는 "인증" 단계로 인증 수행 후, 인증이 성공하면 PANA SA(Security Association)를 협상하는 과정이다. 세 번째 단계는 "종료" 단계로 세션을 맺고 있는 PaC 와 PAA 가 세션을 종료하는 과정이다.

이 세 과정을 세부적으로 알아보면 다음 [그림 x]와 같이 표현할 수 있고, 이 메시지들은 모두 UDP 상에서 전송된다.



[그림 2] PANA 프로토콜의 동작과정

① PANA-PAA-Discover

PaC 가 액세스 네트워크의 PAA 를 찾기 위한 메시지로 링크 로컬 멀티캐스트 주소로 이 메시지를 전송한다. PaC 가 이미 PAA 의 주소를 알고 있더라도 PaC 는 PAA 의 주소로 이 메시지를 전송한다.

② PAA-Start-Request

PAA 가 PANA-PAA-Discover 메시지에 대한 응답으로 PaC 에게 보내는 메시지로 UDP 상에서 가상 세션을 생성하기 위한 쿠키가 포함된다. 또한 쿠키를 이용하여 DoS 공격에 안전하게 된다.

③ PAA-Start-Request-Answer

PaC 가 PANA-Start-Request 메시지에 대한 응답으로 PAA 에게 보내는 메시지로 PAA-Start-Request 메시지

에서 포함된 Cookie 를 그대로 복사하여 전송한다.

④ PAA-Auth-Request, PAA-Auth-Answer

PAA-Auth-Request 와 PAA-Auth-Answer 는 EAP 를 주고 받으면서, EAP 메소드를 이용하여 PaC 를 인증하고 PANA SA 를 협상하는데 사용되는 AAA-Key 를 도출해내는데 사용되는 메시지이다. 이 때 EAP 메시지는 PAA 와 연결된 백엔드 AAA 서버로 전송되어 인증이 수행된다.

⑤ PAA-Bind-Request, PAA-Bind-Answer

PAA-Bind-Request 와 PAA-Bind-Answer 는 위에서 도출된 AAA-Key 를 이용하여 PANA SA 를 협상하고 PANA 세션을 형성한다. 이후 서로 간에 주고 받는 메시지는 모두 무결성이 보장되고 재사용 공격에 안전하게 된다.

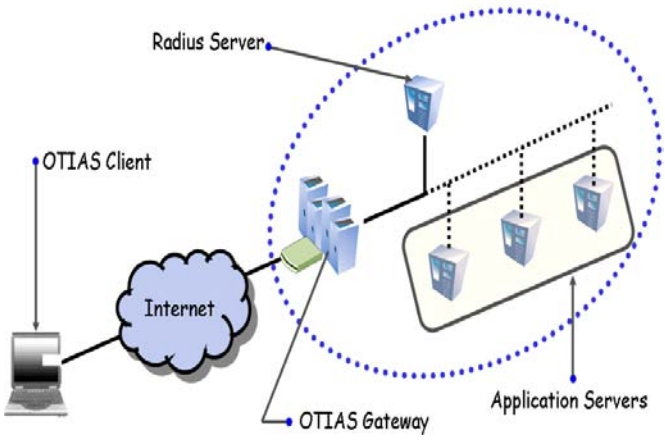
⑥ PAA-Terminatin-Request,PAA-Terminatin-Answer

PAA-Termination-Request 와 PAA-Termination-Answer 는 PANA 세션을 끊고 PANA SA 를 삭제하는 메시지이다[2].

3. OTIAS 시스템

OTIAS 시스템은 IPv6 가상 사설망 기반의 통합 사용자 인증 시스템으로써 IPv6 IPSec 기반의 VPN 환경을 기반으로 한다. OTIAS 시스템은 공인인증서를 기반으로 단일 사용자를 인증하기 때문에 사용자 인증의 보안성이 향상되고, 인증 후 IPv6 의 기본 확장 헤더로 제공되는 IPSec 에서 필요한 SA 협상 정보를 수행하여 사용자 인증과 SA 협상을 통합적으로 관리하여 그 효율성이 향상된다는 장점을 가진다. 이를 위해 OTIAS 시스템에서는 OTIAS 프로토콜을 이용한다. 이 프로토콜은 PANA 프로토콜 동작과정의 원리를 이용하고, 기존의 EAP-TLS 프로토콜을 수정한 EAP-IAPTLS(Integrated Authentication Protocol TLS)를 사용하여 인증서 정보 전달시의 과부하를 감소시키고 이후 IPSec 에서 사용할 SA 협상, 재협상 기능까지 수행할 수 있도록 기능을 첨가한 프로토콜이다[3][4][5].

OTIAS 시스템은 [그림 3]와 같이 PaC 의 역할을 하는 IAPTLS 클라이언트, PAA 의 역할을 하는 EAP-Forwarder, 백엔드 AAA 서버인 Radius 서버로 구성되어 있다.



[그림 3] EAP-IAPTLS 프로토콜의 구성요소

OTIAS 클라이언트는 OTIAS 게이트웨이와 EAP 프로토콜을 이용하여 통신하게 되고, OTIAS 게이트웨이와 Radius 서버는 Radius 프로토콜을 이용하여 통신하게 된다. EAP 메시지 내부의 속해 있는 EAP-IAPTLS 메시지는 OTIAS 게이트웨이가 포워딩하여 논리적으로 OTIAS 클라이언트와 Radius 서버가 서로 교환하는 것으로, 이 메소드를 이용하여 인증서를 교환하여 인증을 수행하고 SA 협상을 수행한다. 이 과정을 모두 마치게 되면 OTIAS 클라이언트와 OTIAS 게이트웨이는 IPSec 터널링이 구성되어 비밀성과 무결성을 만족하는 데이터 통신을 할 수 있게 된다[3][4][5].

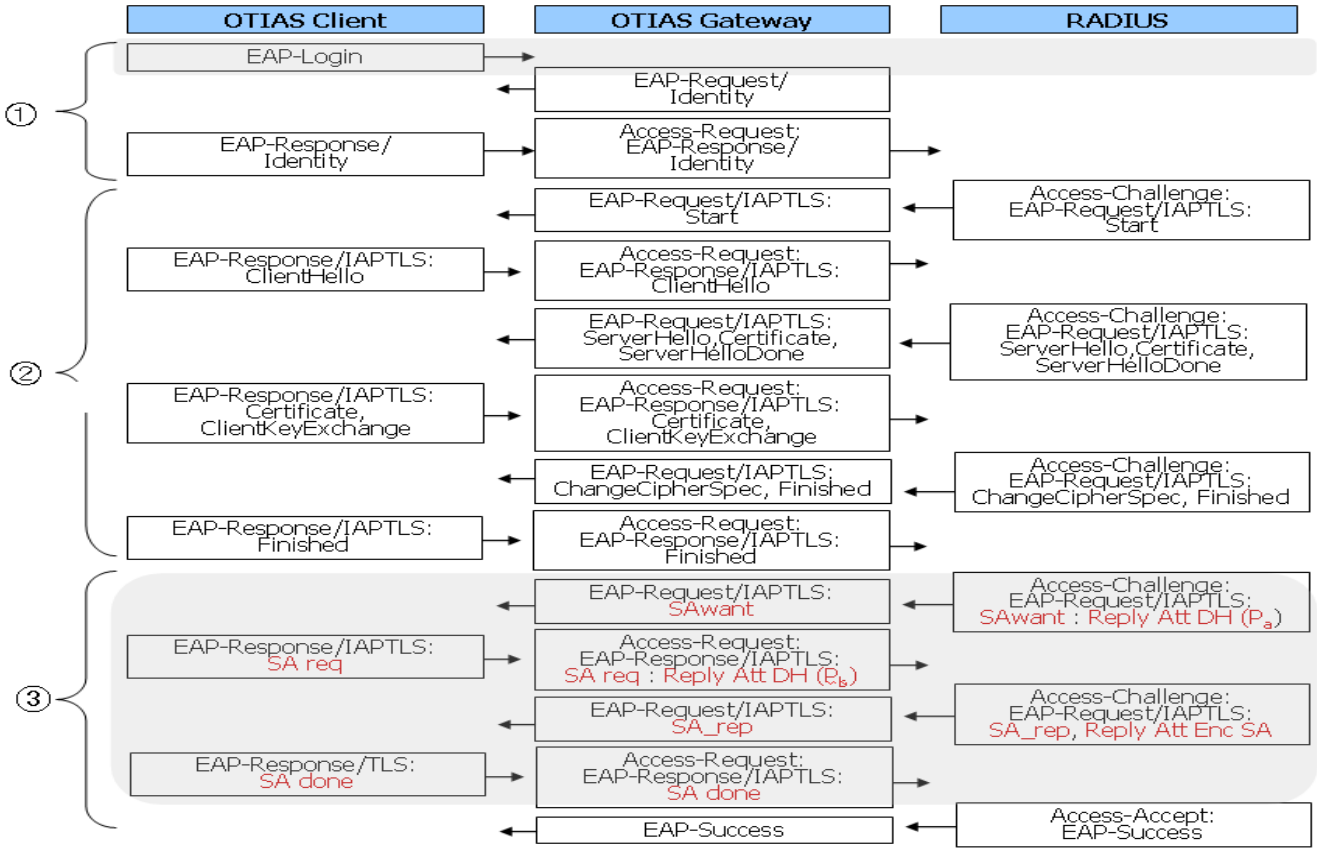
우선 OTIAS 프로토콜 내부에 EAP 프로토콜에서 IAPTLS 메소드를 이용하기 위해 사용하는 타입은 30 으로 정의한다. 또한 IAPTLS 메소드의 기본 메시지 형식은 TLS 메소드의 메시지 형식과 동일하고 IAPTLS 메소드의 Start, ClientHello, ServerHello, Certificate, ServerHelloDone, ClientKeyExchange, ChangeCipherSpec, Finished 메시지의 형태는 기존의 TLS 메소드의 메시지들과 동일하다. 하지만 IAPTLS 에서 사용하는 플래그는 TLS 의 플래그와 약간의 차이를 가진다. IAPTLS 의 플래그는 다음 [그림 4]와 같다.

0	1	2	3	4	5	6	7
L	M	R	IS	SA	O	R	R

L = Length included
M = More fragments
IS = EAP-IAPTLS Start
SA = Security Association
O = Logout
R = Reserved

[그림 4] IAPTLS 의 flag 사용 형식

OTIAS 프로토콜의 세부적인 과정은 [그림 5]와 같다. OTIAS 프로토콜은 크게 로긴 과정, 인증과정, SA 협상 과정으로 구분된다. ① 과정은 로긴 과정으로 OTIAS 클라이언트가 네트워크로 접근하기 위해 OTIAS 게이트웨이로 단순히 '0' 이라는 EAP-Login 메시지를 전송하여 로긴하려는 의도를 OTIAS 게이트웨이에게 알린 후에 자신의 Identity 를 전송하는 단계이다. 이 단계에서는 일차적으로 접근이 가능한 사용자인지 판단한다. ② 과정은 EAP-TLS 와 동일하게 공인 인증서를 주고 받으면서 상호 인증을 수행하는 인증 과정이다. ③ 과정은 SA 협상 과정으로 TLS 메소드에는 없는 SAwant, SA req, SA rep, SA done 메시지를 IAPTLS 를 추가하여 사용하고, 이 메시지를 이용하여 OTIAS 클라이언트와 Radius 서버가 SA 협상을 수행한다. 협상이 끝나면 협상된 SA 를 안전하게 OTIAS 게이트웨이로 통보하기 위해 Radius 프로토콜에 Vendor Specific Attribute 로 새로 정의한 DH, EncSA Attribute 를 사용한다. SA 협상은 Radius 서버에서 OTIAS 클라이언트로 보내는 SAwant 메시지로 시작된다. SAwant 메시지를 전송하면서 이후에 협상될 SA 를 안전하게 OTIAS 게이트웨이로 보내기 위해 사용할 암



[그림 5] OTIAS 프로토콜

호화 키를 생성하는데 필요한 디피헬만 값 하나 P_a 를 DH Attribute 에 저장하여 OTIAS 게이트웨이로 전송한다. OTIAS 게이트웨이는 P_a 를 저장하고 SAwant 메시지를 OTIAS 클라이언트로 로워딩한다. OTIAS 클라이언트는 자신이 협상하기 원하는 SA 리스트를 몇 개 선정하여 SA req 메시지에 담아 전송한다. OTIAS 게이트웨이는 이 메시지에 디피헬만 값의 또 다른 하나 P_b 를 DH Attribute 에 추가하여 Radius 서버로 전송한다. 이제 Raidus 서버는 OTIAS 게이트웨이와 안전하게 통신할 수 있는 대칭키 K 가 생성되고, SA req 메시지에서 하나의 SA 를 선택할 수 있다. 선택된 SA 는 SA rep 메시지에 저장되고, 또한 대칭키 K 로 DES 암호화한 SA 를 EncSA Attribute 에 저장하여 전송한다. OTIAS 클라이언트는 EncSA Attribute 에서 암호화된 SA 를 추출하고 복호화여 SA 를 저장한 후, 메시지를 OTIAS 클라이언트로 전송한다. OTIAS 클라이언트는 SA rep 에서 SA 를 추출하여 SA 를 저장한다. 저장이 완료되면 SA done 메시지로 SA 협상이 끝났음을 통보한다. 이후 OTIAS 클라이언트와 OTIAS 게이트웨이 사이에서는 IPsec 터널이 형성되며 OTIAS 클라이언트와 액세스 네트워크 사이 통신은 모두 비밀성, 무결성이 만족된다.

4. 결론 및 향후 연구 계획

본 논문에서는 망 접근 기술에 무관하며 강력한 인증을 수행하고 안전한 IPsec 통신을 위한 SA 협상을 하는 OTIAS 시스템과 프로토콜에 대하여 알아보았다. OTIAS 프로토콜은 기존의 EAP, TLS, RADIUS 프로토콜의 연동 과정을 수정하고 추가하여 개발되었다. 이 프로토콜은 기존의 TLS 프로토콜의 기술을 이용하여 공인 인증서를 이용한 상호 인증을 수행하며, 인증 후 새로 정의 된 SAwant, SA req, SA rep, SA done 메시지와 DH, EncSA Attribute 를 이용하여 SA 협상을 수행한다. 현재 본 시스템은 구현이 완료되었으며, 추가적으로 디버깅 작업과 성능 개선을 위한 연구를 계획하고 있다.

참고문헌

[1] D. Forsberg, "Protocol for Carrying Authentication for Network Access (PANA)", draft-ietf-pana-pana-07 (work in progress), December 2004.

[2] IEEE, “Port-Based Network Access Control”, IEEE Std 802.1X-2001, June 2001.

[3] B. Aboba, “PPP EAP TLS Authentication Protocol”, RFC 2716, October 1999.

[4] B. Aboba, “RADIUS Support For EAP”, RFC 3579, September 2003.

[5] B. Aboba, "Extensible Authentication Protocol (EAP) Key Management Framework", draft-ietf-eap-keying-04 (work in progress), November 2004.