제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003.5)

# IPSec 키 관리 시스템 관련 오픈 소스 프로그램 분석

김건웅*, 송병권**
*목포해양대학교 해양전자·통신공학부
**서경대학교 정보통신공학과
e-mail:kgu@mmu.ac.kr

# A Study on Open Source Programs for IPSec Key Management System

Geonung Kim*, Byung-kwen Song*
*Division of Electronics & Comm. Eng., Mokpo National Maritime University,
**Dept. of Information & Comm. Eng., Seokyeong University

요 약

망 차원에서의 보안 기능을 지원하기 위한 IPSec을 도입하기 위해서는 통신에 참여하는 각 호스트들의 인증에 필요한 키 관리 시스템 구축이 선행되어야 한다. 본 논문에서는 이러한 IPSec 키 관리 시스템 구축을 위해 현재 인터넷에 공개되어 있는 관련 오픈 소스 프로그램들을 분석하고, 이를 통한 구축 방안을 제시한다.

## 1. 서론[1]

인터넷 망을 통한 정보 교류가 증가함에 따라 인터넷 보안의 중요성도 날로 커지고 있다. 현재 인터넷 망 프로토콜인 IPv4에서 보안을 제공하기 위해 시작된 IPSec에 대한 연구는 차세대 망 프로토콜인 IPv6에 이르러 기본 프로토콜로 포함되기에 이르렀다. 따라서 IPv6가 도입되면 현재 트랜스포트 계층이나 응용 계층에서 개별적으로 제공되던 보안 서비스가 망 계층에서 제공되므로, 보다 신뢰할 수 있는 망 환경 구축이 가능할 것이다.

이러한 망 계층의 보안 기능을 제공하기 위해서는 망을 구성하는 호스트들의 운영체제에서 IPSec을 지원하는 것이 필수적이며, 또한 각 호스트들의 인증에 필요한 IPSec 키 관리 시스템도 도입되어야 한다. 현재 IPSec과 IPSec 키 관리 시스템 구현에 관련된 많은 연구들이 진행 중인데, 본 논문에서는 오픈 소스 형태로 진행 중인 연구들을 분석하여 IPSec 키 관리 시스템 도입 방안을 강구한다.

본 논문은 한국과학재단 목적기초연구(R05-2002-000-01055-0)의 중간 결과물입니다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 인터넷 보안 구조를 살펴보고, 다음 3장에서는 공개키 암호화를 이용한 키 관리 시스템에 대해 살펴본다. 다음 4장에서는 현재 공개 소스 형태로 진행되고 있는 관련 연구들과 상호 관계를 검토하며, 5장에서 결론을 맺는다.

## 2. 인터넷 보안구조

인터넷 보안 구조는 4가지 요소들 - 보안 프로토콜, 보안 연관(SA: Security Association), 키 관리 프로토콜, 그리고 인증이나 암호화에 쓰이는 다양한 알고리즘들 -로 구성된다[1][2].

보안 프로토콜에는 AH(Authentication Header)와 ESP(Encapsulating Security Payload)가 있는데, AH는 비연결형 무결성(connectionless integrity), 출발지 인증(data origin authentication)을 제공하며, 부수적으로 재전송 방지(anti-replay) 서비스를 제공할 수 있다. ESP는 기밀성(confidentiality)과 제한된 트래픽 흐름 기밀성(traffic flow confidentiality), 비연결형 무결성, 출발지 인증, 재전송 방지 서비스를 제공할 수 있다[3][4].

보안 연관은 논리적 연결로서, AH나 ESP 서비스는 이를 이용하며, 키 관리 프로토콜들의 역할은 이러한 SA를 생성하고 유지하는 것이다. 이러한 SA는 SPI(Security Parameter Index), IP 목적지 주소, 보안 프로토콜 식별자(AH 또는 ESP)로 구별되며, 서비스별로, 방향별로 하나씩 생성되어야 한다. 따라서 하나의 양방향 연결에서 AH와 ESP 서비스를 동시에 이용하고자 한다면 4개의 SA가 필요하다.

IPSec에서는 SA의 수동 설정과 자동 설정을 모두 규정하고 있는데, 보안 프로토콜과 독립적으로 운영되도록 되어있다. 수동 설정은 관리자가 직접 각 시스템의 키를 설정하는 방법으로 소규모의 고정된 망 환경에서 운영이 가능하다. 자동 설정은 IKE[5]를 비롯한 키 관리 프로토콜을 이용하여 사용자별, 세션별로 키를 생성하고, 유지하며, 폐기하는 것으로서 현재 IKEv2[6]와 JFK[7] 등이 활발히 논의 중이다. SA의 세밀성(granularity)은 이러한 자동 설정이 지원되는가에 대해 종속적인데, 미세한 SA를 지원하기 위해선 자동 설정이 필수적이며, 또한 규모가 큰 유동 망에서 보안 구조를 적용하고자 할 때에도 필수적이다.

마지막으로 인증이나 암호화에 쓰이는 다양한 알고리즘들인데, 인터넷 보안 구조에서는 이들에 대한 특별한 제약이 없이 모두를 수용할 수 있는 구조로 되어 있으며 상호 호환을 위해 최소한의 알고리즘을 반드시 포함하도록 규정하고 있다.

## 3. IPSec 키 관리 시스템

IKE에서는 2단계에 걸쳐 SA를 설정하는데, 첫 단계에서는 IKE 자체의 SA를 설정하고, 두 번째 단계에서 IPSec SA를 설정한다. 첫 단계에서는 IPSec SA 설정시 필요한 암호화 매개변수를 합의하고 공유 비밀키를 생성한다. 이때 쌍방간의 인증 방법으로 ①전자서명, ②공개키 암호화, ③수정된 공개키 암호화, ④미리 공유된 비밀키 등을 활용할 수 있다. 이들 중 가장 보안이 강화된 방법이 공개키 암호화 방법을 이용하는 ②와 ③이다[5].

그림 1은 ②의 1단계의 메인 모드(main mode)를 보여주고 있다. 여기에서는 6개의 메시지를 통해 인증을 수행하고, IKE SA를 설정한다. 이때 세 번째 메시지와 네 번째 메시지에서 각각 상대방의 공개키를 이용하여 신원(IDii, IDir)과 nonce(Ni, Nr)을 암호화하여 전송하고, 그것을 자신의 비밀키로 복호화하여 처리를 함으로써 인증이 이루어진다. 결국 이를 이용하기 위해서는 각자 신뢰할 수 있는 제3자를 통해 상대방의 공개키를 알아내야 한다.

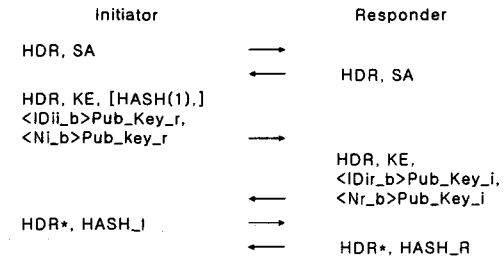


그림 1. 공개키 암호화를 이용한 메인 모드

인증 문제를 해결하기 위한 대표적인 방안이 전자인증서(digital certificate)와 이를 발급하는 인증기관(CA: Certificate Authority)의 도입인데, 대표적인 표준이 X.509 PKI(Public Key Infrastructure)이다. X.509는 1988년 발표된 이후, 보완과 추가를 통하여 1996년 v3까지 발표되었는데, IETF의 pkix 그룹에서는 X.509를 기반으로 하는 전자인증서의 구성과 전자인증서의 발급/보관/폐기, 전자인증 기관의 구성 등을 정의하는 작업을 하고 있다[8][9].

일반적인 공개키 기반 구조의 구성 요소는 ①인증서(Certificates), ②인증서 상태 확인 방법(Certificate Status Mechanism), ③인증기관(CA: Certificate Authority), ④등록기관(RA: Registration Authority), ⑤데이터 복구 에이전트(Data Recovery Agent), ⑥인증서와 CRL (Certificate Revocation List) 저장소, ⑦인증 정책(CP: Certification Policy)과 인증 적용 지침(Certification Practice Statement)이다. IPSec 키 관리 시스템도 일반적인 공개키 기반 구조의 요소들을 그대로 이용할 수 있는데, 다만 실제로 인증되어야 하는 대상이, 일반적인 사용자가 아닌, 호스트인 관계로 호스트가 스스로 키를 생성하고, 공개키를 등록하는 방법이 제공되어야 한다.

그림 2는 PKI를 기반으로 하는 IPSec 키 관리 시스템과 IKE 프로토콜 데몬인 IKEd와의 동작을 보이고 있다. IKEd는 보안 연관 설정시 IPSec 키 관리 시스템과 상호 동작하여 호스트의 인증 기능을 수행한다. 이러한 키 관리 시스템은 망 관리자를 위하여 구성원들의 인증 정보를 검색하고 관리하는 등의 일련의 작업을 수행할 수 있는 인터페이스를 웹을 통해 제공할 수도 있다.

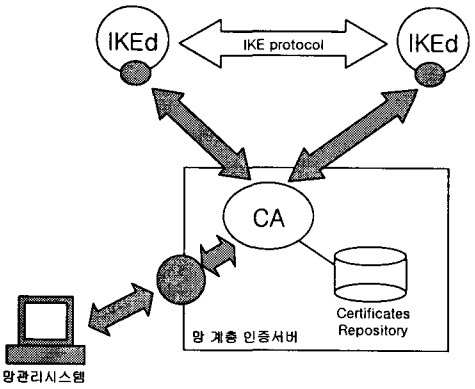


그림 2. PKI 기반 키 관리 시스템

## 4. 관련 오픈 소스 프로그램

### 4.1 암호화/인증 관련 프로그램

#### 4.1.1 OpenSSL

OpenSSL은 암호화 개발도구로서 SSL(Secure Sockets Layer v2/v3)과 TLS(Transport Layer Security v1) 프로토콜과 이를 지원하기 위한 암호화 표준들의 구현을 제공하고 있는데, 현재 API 형태로 214개의 함수를 지원한다. 특히 openssl 프로그램은 쉘에서 명령어 형태로 OpenSSL의 crypto 라이브러리의 암호화 함수를 이용할 수 있는데, 이를 통해 수행할 수 있는 기능은 다음과 같다. ① RSA, DH, DSA 키 파라메터 생성, ②X.509 인증서, CSR, CRL 생성, ③MD 계산, ④Ciphers를 이용한 암호화/복호화, ⑤SSL/TLS 서버와 클라이언트 테스트, ⑥S/MIME 또는 암호화된 전자우편 처리[10].

#### 4.1.2 Cryptlib

암호화 툴킷으로서 다양한 암호화 및 인증 알고리즘을 구현하여 제공하고 있으며, S/MIME, SSL/TLS, CA 서비스 들을 제공하고 있다. 특히, 미국내에서 개발한 제품이 아닌 관계로 미국 통상법의 저촉 없이, 전 세계 어디에서나 이용할 수 있다. 제공되는 암호화기능은 다음과 같다. ①AES, ②Blowfish, ③CAST, ④DES, ⑤triple DES, ⑥IDEA, ⑦RC2, ⑧RC4, ⑨RC5, ⑩Skipjack. 또한 ①MD2, ②MD4, ③MD5, ④RIPEMD-160 ⑤SHA, ⑥HMAC-MD5, ⑦HMAC-SHA, ⑧HMAC-RIPEMD-160, ⑨MAC, ⑩Diffie-Hellman, ⑪DSA, ⑫Elgamal, ⑬RSA 공개키 암호화 기능 등을 제공한다[11].

#### 4.1.3 Cyrus SASL(Simple Authentication and Security Layer)

SASL은 연결형 프로토콜에 인증(authentication)을 제공하기 위한 방법으로 사용자와 서버간 식별 기능과 인증기능을 제공한다. SASL이 이용되면 프로토콜과 연결 사이에 보안 계층이 추가된 것으로 볼 수 있다. Cyrus SASL 라이브러리는 Carnegie Mellon에서 개발한 SASL API의 구현으로 다음 방법들을 이용한 인증을 지원한다. ①anonymous, ② CRAM-MD5, ③KERBEROS_V4, ④plain, ⑤ GSSAPI(MIT Kerberos 5 or Heimdal Kerberos 5), ⑥DIGEST-MD5. ④를 제외한 나머지 방법을 지원하기 위해 데이터베이스로 gdbm 또는 ndbm을 이용한다[12].

### 4.2 디렉토리 관련 프로그램

#### 4.2.1 OpenLDAP

OpenLDAP은 Michigan 대학에서 구현한 LDAP을 기반으로 LDAP 서버를 구현한 것이다. 앞서 언급한 Cyrus SSL과 OpenSSL을 기반으로 하여 LDAPv3를 제공하며, Kerberos 기반 인증도 지원한다. 데이터베이스로 BDB를 주로 이용하며, GDBM도 가능하다. 이러한 OpenLDAP을 이용, 인증 관련 저장소와 인증 프로토콜을 구현할 수 있다[13].

### 4.3 PKIX 관련 프로그램

#### 4.3.1 OpenCA

OpenCA는 PKI 표준을 구현을 통해 PKI 인증기관을 구현하기 위한 작업으로 앞서 언급된 OpenSSH, OpenLDAP과 Apache 등의 오픈 소스 프로그램에 기반을 두고 있다. 현재 CA에서 이용될 보안 개요를 정의하는 작업과 도입과 이용이 용이한 CA 구현 작업이 진행 중이다[14].

#### 4.3.2 pyCA

X.509를 기반으로 하는 인증기관을 구현하기 위한 작업으로 앞서 언급한 OpenSSH를 기반으로 하고 있으며, 특히 인증서의 생성, 발급과 CRL을 가져올 수 있는, 사용자를 위한 웹 인터페이스를 제공하고 있다[15].

#### 4.3.3 SimpleCA

X.509를 기반으로 하는 인증기관을 구현하기 위한 작업으로 앞서 언급한 OpenSSH를 기반으로 하

고 있으며, 사용하기 용이한 GUI를 제공하는데, 설정에 일부 제한이 있다[16].

4.3.4 Jonah PKIX Freeware

Jonah PKIX는 pkix 표준들을 구현한 프리웨어로 역시 망 계층 인증 서버 구현 시 이용할 수 있으나 미국민으로 자유 이용 대상이 제한되어 있다. 이것은 Open Group의 ODE(Open Development Environment)를 기반으로 구현되었으며, Intel의 CDSA에서 제공하는 Crypto, LDAP, PKCS#11 인터페이스를 이용하고, Cylink사의 암호화 도구를 이용하였다. 그러나 Cylink사의 합병으로 관련 암호화 도구가 더 이상 공개 프로그램이 아닌 관계로 도입이 불가능하다[17][18].

이외에도 PKI를 구현한 공개 소스 프로그램이 다수 존재한다[19].

4.4 IPSec 및 IPSec + PKIX 관련 프로그램

4.4.1 FreeS/WAN

Linux에서 IPSec을 구현한 대표적인 공개 소스 프로그램인 FreeS/WAN은 IPv4에서 IPSec과 IKE를 지원하고 있다. 원래 인터넷 상에서의 VPN 제공을 목적으로 추진된 관계로 제한된 수의 호스트간 연결 지원에만 초점을 맞추고 있으며 따라서 호스트들의 인증을 담당할 키 관리 시스템의 도입은 배려하지 않고 있다[20].

4.4.2 FreeS/WAN with X.509

앞서 언급한 FreeS/WAN의 패치 형태로 PKI를 기반으로 한 인증과 OpenPGP를 기반으로 하는 인증 기능을 제공한다. PKI를 기반으로 하는 인증 기능을 위해 앞서 언급한 OpenSSH, OpenLDAP을 기반으로 하고 있으며, FreeS/WAN에서 구현한 IKE 코드에 인증 부분을 담당하는 코드가 삽입된 형태로 제공되고 있다[21].

## 5. 결론

본 논문은 대학 전산망에서 망 계층 보안 서비스를 지원하기 위한 방안 연구의 부산물로서 현재 공개 소스 프로그램 형태로 제공되고 있는 암호화/인증 프로그램들과 LDAP 서버 프로그램, PKIX와 IPSec을 구현한 프로그램들을 살펴보았다. 현재 이러한 검토 결과를 바탕으로 대학전산망의 호스트 인증에 이용할 망 계층 인증 서버와 이와 관련된 관리 기능을 제공하는 웹 인터페이스를 구현 중이다. 또한, 각 호스트 정보와 인증 정보를 체계적으로 보관하기 위한 디렉토리의 스키마 정의 작업이 진행 중이며, 아울러 인증 대상이 인간이 아닌 호스트인 관계로, 자동으로 IP와 인증 정보를 등록하는 방안이 모색 중이다.

## 참고문헌


[1] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, Nov. 1998

[2] A. Krywaniuk, "Security Properties of the IPSec Protocol Suite", draft-ietf-ipsec-properties-02, June 2002

[3] S. Kent, R. Atkinson, "IP Authentication Header", RFC 2402, Nov. 1998

[4] S. Kent, R. Atkinson, "IP Encapsulating Security Payload(ESP)", RFC 2406, Nov. 1998

[5] D. Harkins, D. Carrel, "The Internet Key Exchange", RFC 2409, Nov. 1998

[6] D. Harkins, C. Kaufman, S. Kent, T. Kivinen, R. Perlman, "Proposal for the IKEv2 Protocol", draft-ietf-ipsec-ikev2-02, April 2002

[7] W. Aiello, S.M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A.D. Keromytis, O. Reingold, "Just Fast Keying", draft-ietf-ipsec-jfk-04, April 2002

[8]http://www.ietf.org/html.charters/pkix-charter.html

[9] A. nash, W. Duane, C. Joseph, D. Brink, "PKI Implementing and Managing E-Security", McGraw-Hill, 2001

[10] http://www.openssl.org/

[11]http://www.cs.auckland.ac.nz/~pgut001/cryptlib/

[12] http://asg.web.cmu.edu/cyrus/

[13] http://www.openldap.org

[14] http://www.openca.org/

[15] http://www.pyca.de/

[16] http://users.skynet.be/ballet/joris/SimpleCA/

[17] http://web.mit.edu/pfl//

[18] http://www.cylink.com/

[19] http://www.pki-page.org/

[20] http://www.freeswan.org

[21] http://www.strongsec.com/freeswan