# IP계층에서의 VPN 전송성능에 관한 연구

정회원 임 형 진*, 권 윤 주* 정 태 명*

# Secure VPN Performance in IP Layers

Hyung J. Lim*, Yoon J. Kwon, Tai M. Chung* *Regular Members*

요 약

본 논문에서는 IPSec을 리눅스에서 구현하여 IP 계층에서 AH, ESP 프로토콜 사용시 노드간 성능을 측정하여 네트웍에서의 처리 성능에 영향을 미치는 인자에 대하여 분석을 하였다. IPSec에서 사용하는 AH와 ESP는 인증 데이터의 계산, 비교, 암호화에 의해서 IP프로토콜의 전송성능에 영향을 준다. 이에 AH, ESP 프로토콜에 대하여 응용 프로토콜(FTP, Telnet, SMTP)을 이용해 전송 데이터량을 증가시키며 Non IPSec과 IPSec의 처리성능을 평가하여 보았다. 성능평가 결과 전송패킷의 크기와 터널링에 사용되는 암호, 인증 함수, 호스트의 CPU속도, IPSec 구현방식이 전송성능에 영향을 주는 인자로 나타났으며, 대용량의 트래픽에서는 Non IPSec에 비하여 10 여배 이상의 전송지연이 발생하여 IPSec 전송에는 적합하지 않았다.

ABSTRACT

This paper analyzes Security Performance and Processing Performance to measure performance between nodes by using AH and ESP protocol. IPSec VPN provides application with security service implemented in IP Layer while traffic cost and packet processing time is increased by encryption, decryption and authentication in AH and ESP. We measured overall packet processing time and IPSec module processing time. The result of the efficeiency test showed that the factors of influencing electrical transmission efficiency were the size of electrical transmission packets, codes used for tunnelling, authentication functions, CPU velocity of hosts, and the embodiment of IPSec; for a high capacity traffic, IPSec transmission was not appropriate, because transmission velocity was delayed by more than ten times in comparison with Non-IPSec.

## Ⅰ. 서 론

90년대 이후에 인터넷은 개인뿐만 아니라 기업체들에게 기업간 혹은 기업내부간의 새로운 통신매체로서 자리잡아 가고 있으며 21세기 정보화 시대에 있어 정보화 사회의 기반 구조가 되어 가고 있다. 이러한 인터넷에는 많은 다른 기술들이 융합되어 있고, 신뢰성과 보안이 요구되어지는 통신영역기술들로서 계속 발전되어가고 있다. 기업측면에서는 분산된 기업내의 통신을 위하여 통신 사업자의 전송장비를 임대하여 사설 통신망을 구축해 왔으며 이는 그 범위, 규모에 따라 상당한 구성의 어려움과 많은 비용이 소요되고 있었다.

전용선에 의한 사설망이 아닌 공중망 VPN으로의 변화는 저렴한 가격에, 유연성과 확장성을 제공할 수 있는 서비스를 제공할 수 있으나, 반면에 공중망의 특징인 보안상의 위험, 공중망의 성능을 예측할 수 없다는 단점을 갖고 있다. 이에 대하여 터널링 프로토콜의 사용을 통해 공중망을 통과하는 사설망의 트래픽들에 보안성을 제공하며, 성능에 있어서는 RSVP, MPLS와 같은 기술을 통하여 보장된 대역폭을 확보하고자 하고 있다.[9,10,13,14,21]

본 논문은 IP기반에 적용 가능한 IPSec을 리눅스에서 구현하여 AH, ESP프로토콜을 사용하여 노드

* 성균관대학교 전기 전자 및 컴퓨터 공학부
논문번호 : 010199-0725, 접수일자 : 2001년 7월 25일

간 네트워크의 영향을 측정하고자 하며 이는 향후 기존 네트워크에 VPN 도입시 보안성능 대 처리성능에 있어서 적절한 정책결정에 기반이 될 수 있을 것이다. 이를 위하여 본 논문에서는 관련연구로서 2장에서는 계층별 VPN 프로토콜의 특성과 VPN 구성을 통해 성능과 관련한 고려사항을 살펴보고, 3장에서는 현재 IETF 표준인 IPSec에 대하여 기술한다. 4장에서는 IPSec에서 사용되는 암호, 인증 함수의 분석을 통해 성능에 영향도를 비교해보며, 5장에서는 현재 구현된 IPSec 제품군의 성능을 비교하며, 6장에서는 본 논문에서 구현된 리눅스상의 IPSec과 성능 테스트 방식, 성능 테스트 결과와 분석을 보여준다.

## Ⅱ. VPN 도입

VPN을 형성할 수 있는 프로토콜은 각 계층별로 정의되어진다. 네트워크 프로토콜을 터널링 프로토콜에 캡슐화하는 계층 2 터널링 ( L2TP : Layer 2 Tunneling)이 있으며, 네트워크 프로토콜을 직접 터널링 프로토콜로 캡슐화 하는 계층 3 터널링 (Layer 3 Tunneling )이 있다. 이러한 터널링 프로토콜에 의한 보안 정책 형태는 통신의 각 계층에서 이루어질 수 있다. 계층 3 터널링 프로토콜인 IPSec은 IETF 표준 프로토콜로서 IP 프로토콜을 사용하면서 키 관리, 인증, 암호화 기능을 제공하며, IPv4에서 구현이 가능하며 대규모의 인터넷 환경에 적합한 보안 프로토콜이다.[9,11,15]

네트워크 계층에서의 VPN 보안 정책의 구현은 두 가지 형태를 가질 수 있는데, 이는 데이터에 대한 보안에 관한 책임이 주어지는 위치에 따른 구분이다. 보안 정책이 호스트간 end-to-end방식으로 구성되어지거나 방화벽이나 라우터와 같은 다른 네트워크 자원간에 이루어지는 node-to-node방식이 있다.

end-to-end 방식의 보안 정책에서는 각 호스트가 직접 전송과 수신자의 역할을 하고 node-to-node 보안 방식보다 더 신뢰적일 수 있다. 하지만 end-user가 node에서 처리되어질 수 있는 암/복호화, 인증의 잠재적인 요소들을 처리해야 하므로 성능에 있어서 고려할 사항이 있다.

VPN은 보안책임에 관한 정책에서 뿐만 아니라 [그림 1]과 같이 VPN의 구성에 따라서 여러 형태를 지닐 수 있다. LAN에서 공중망으로의 연결은 ISP로 접속을 통해서 이루어지게 되며, 이 때의 경로는 LAN에서 방화벽과 라우터에 이어 CSU/DSU를 따르는 형태가 될 것이다. 여기서 VPN 하드웨어와 소프트웨어는 LAN에서 ISP까지의 네트워크의 경로 중 다양한 위치에 자리할 수 있게 된다. 완전하게 구현된 VPN 형태는 암호화, 인증, 터널링 서비스를 제공하며 아래와 같은 다양한 위치에서 구현되어질 수 있다([그림 1] 참고).

- CSU/DSU와 라우터 사이 또는 라우터와 방화벽 사이 구현
- 방화벽 혹은 라우터의 한 기능으로 구현
- WAN 링크의 번들링, 라우팅, 방화벽, VPN 서비스들의 통합된 형태로 구현
- NT서버나 Netware와 같은 NOS에서 제공하는 소프트웨어로서 구현

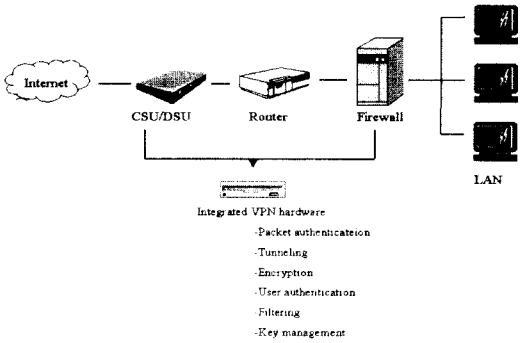



그림 1. 다양한 VPN 적용지점

네트워크 사이에 방화벽은 많은 트래픽을 처리하고 있기 때문에 암호화나 키관리 등의 기능이 부가될 경우 성능에 있어서 저하가 이루어질 수 있다. 방화벽에 VPN 기능을 추가하는 것은 성능에 있어서 고려인자가 될 수 있다. 또한 VPN의 라우터 구현은 라우터의 패킷 포워딩 성능과 관련이 있으며 보안을 요구하지 않는 패킷에 대하여 IPSec에 영향을 받지 않아야 하며 많은 구현에 있어서 암호, 인증 연산을 위한 추가적인 하드웨어의 지원이 요구되고 있다.

이상과 같이 VPN 전송성능은 구현, 구성방식에 따라, 계산상의 오버헤드가 예상되는 암호화와 메시지 인증이 처리되는 지점에 따라 영향을 받는다고 할 수 있다. VPN의 도입이 기존에 구축된 망에 네트워크의 진화 모델의 적용이라고 할 때, 이러한 정보보호를 보장하기 위한 잠재비용은 검증되어 적절한 보안 정책으로 적용되어야 할 것이다.[8,9,11,13,15]

## Ⅲ. IPSec의 개요

IPSec은 IETF에 의하여 표준화된 암호화와 관련된 시스템구조 및 키 관리 프로토콜로 IP계층의 확장된 형태로서 두 개체간에 통신에서 기밀성과 인증 기능을 제공한다.

주요 구성요소에 의한 처리과정으로 IPSec은 시스템으로 하여금 보안 프로토콜을 선택하고, 암호화 알고리즘을 결정하며, 암호화 키를 결정할 수 있게 함으로서 IP계층에서 보안서비스를 제공할 수 있도록 한다. IPSec은 두 호스트 사이, 두 보안 게이트웨이 사이, 또는 보안 게이트웨이와 호스트사이의 통신을 보호하기 위해 사용할 수 있다. IPSec이 제공할 수 있는 보안 서비스는 접근제어, 무결성, 데이터 출처 인증, 재연된 패킷의 거부, 기밀성 그리고 제한된 트래픽 흐름 기밀성이다.[11]

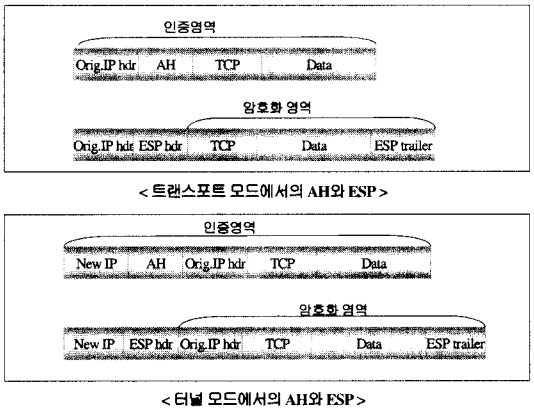



그림 2. IPSec의 보안 프로토콜 및 전송모드

[그림 2]에서 보는 바와 같이, IPSec의 보안프로토콜인 AH(Authentication Header), ESP(Encapsulation Security Payload)는 보호되는 데이터 영역에 따라 터널 모드(Tunnel Mode)와 트랜스포트 모드(Transport Mode)의 두 가지 운용 모드를 갖는다. 트랜스포트 모드는 일반적으로 보안 호스트 구현시 사용되며 상위 계층 프로토콜에 대한 보호 서비스를 제공한다. 터널모드의 경우 터널이 형성되는 시작점과 끝점을 인식할 수 있도록 하기 위해서 보안 호스트 및 게이트웨이 구현시 모두 적용되며 외부 IP헤더를 새로이 생성하여 내부 IP헤더를 포함한 IP 패킷 전체에 대한 보호 서비스를 제공한다.[1,2,11]

IPSec 보안프로토콜인 AH는 전송패킷에 대하여

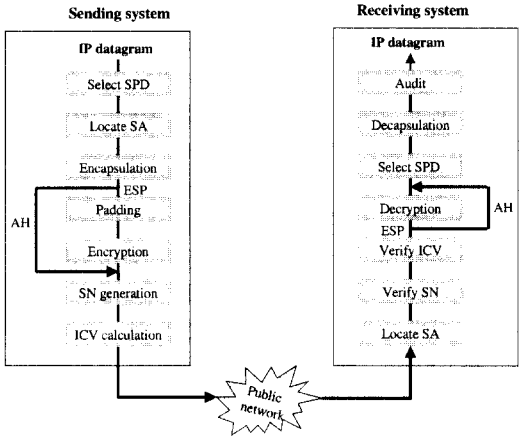



그림 3. IPSec 처리 과정

리플레이 방지와 IP 데이터그램에 대한 인증, 무결성을 제공한다. 무결성은 MD5와 같은 메시지 압축 알고리즘의 메시지 인증 코드에 의해 생성되는 체크섬에 의하여 보장되고, 인증은 인증 되어지는 데이터에 대한 비밀 공유키를 제공함에 의해 보장되어진다. 또한 리플레이 방지는 AH 헤더안에 있는 순서 번호에 의해 보장되어진다. IPSec은 이러한 세 가지 기능을 인증(Authentication)으로서 제공하고 있다. 인증 데이터 필드의 내용은 규정된 인증 알고리즘(MD5, SHA-1)으로 생성하고 있으며, 인증 데이터는 전송 중 그 내용이 변경될 수 있는 영역을 제외한 전체 패킷에 대하여 계산된다.

ESP는 IP 패킷의 비밀성과 무결성, 인증, 리플레이 방지 기능을 제공한다. ESP 암호화는 공유 대칭키를 사용하며 이를 통신 당사자간에 서로 교환하여 암호화 함수(DES)를 암호화와 복호화에 사용한다. ESP의 인증은 AH에서와 같은 인증(MD5, SHA-1)알고리즘을 사용한다.[4,5,11]

IPSec 처리과정은 [그림 3]과 같이 나타낼 수 있고, 이는 IPSec헤더 생성과정과 메시지 인증 및 암호화 과정으로 구분된다. IPSec헤더 생성은 IP와 TCP/UDP 사이에 위치해야 하고, AH인 경우 생성된 헤더의 인증 필드에 AH상위 계층에 대한 인증값을, ESP인 경우 생성된 헤더의 페이로드 필드에 ESP헤더 상위 계층에 대한 암호화한 값을 포함해야 한다. 위의 두 과정은 새로운 네트워크 프로토콜인 IPSec의 구현에서 헤더 생성순서를 유지할 수 있기는 하지만, 커널내 수행 위치가 분리되어 있어 IPSec 모듈을 분산화하여 IPSec 프로토콜 구현의 복잡도를 증가시키게 된다. 본 논문에서는 트랜스포

트 계층에 대하여 독립적이고, 향후 IPSec 구현의 완성을 위한 SA협상, 키 관리 관련 모듈로 쉽게 확장 가능하도록 IPSec헤더 생성과 암호, 인증에 대한 처리를 새로운 버퍼의 할당을 통한 일괄적 처리 구현 방식을 사용한다.[3,6,7,11,21]

## Ⅳ. 암호, 인증 함수의 분석

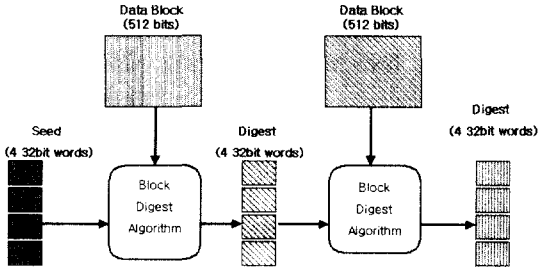
이 장에서는 IPSec에서 데이터보호를 위해 사용되고 있는 암호, 인증 함수로서 현재 출시되는 대부분의 IPSec의 제품에서 제공하고 있는 DES와 MD5의 주요특성과 알고리즘, 계산처리시간에 대하여 비교해보고자 한다.

### 4.1. MD5의 주요특성

MD5는 해쉬 함수로서 MD2, MD4에 이어 MD5가 제안되었다. MD5는 임의의 길이의 서명문을 128비트의 해쉬 값을 출력시키며 임의의 입력 서명문을 512비트 단위로 처리한다. MD5를 동작 시키기전에 서명문을 512비트의 배수가 되도록 패딩(padding)한다.AH에서는 전송패킷의 내용에 대하여 128비트 메시지 압축을 계산한다.

[그림 4]에서와 같이, 각 512비트 블록은 4라운드 기본단계의 압축이 이루어지고, 각 라운드에서 16번의 반복을 통하여 전체 64단계에 처리가 이루어진다. 1라운드에 대한 각 단계는 데이터 블록과 상수, 중간 다이제스트를 통하여 128비트 축적다이제스트 중에 32비트 한 단어를 변경한다.4 라운드의 기본구조는 다음과 같다.(<< lotate)

A = B ((A + f(B, C, D) + x[i] + T[i]) << c2),
B = C + ((B + f( B ,C ,D) + x[i] + T[i]) << c2),
C = D + ((C + f( B ,C ,D) + x[i] + T[i]) << c2),




MD5 block-chained digest algorithm

그림 4. MD5의 블록 체인 다이제스트 알고리즘

D = A + ((D + f( B ,C ,D) + x[i] + T[i]) << c2),

A, B, C, D - 버퍼의 4단어
f() - 비 선형함수 F, G, H, I에 하나
x[i] - x[q*16 + I] 메시지 q번째 512비트 블록에서 k번째의 32비트 단어
T[i] - 행렬 T에서 I번째 32비트 단어

각 라운드는 [그림 5]와 같이 4개의 논리적 함수 f()에 의해서 16회 수행되며 아래와 같다.(^는 XOR이다.)

F(x, y, z) = (x & y) | ((~x) & z)
G(x, y, z) = (x & z) | (y & (~z))
H(x, y, z) = (x ^ y ^ z)
I(x, y, z) = y ^ (x | (~z))

각 단계는 이전단계의 중간 압축 결과를 사용해야하는 순차처리가 요구되기 때문에 쉽게 병렬처리를 하지 못한다. 이전 단계의 결과는 다음단계의 압축에 영향을 준다.

MD5의 계산처리 비용은 각 입력단어에 대한 4라운드의 처리이다. 이 각 라운드가 분석됨에 의해서 전체 성능을 결정할 수 있다. 각 라운드에서 처리시간에 영향을 주는 인자로는 로테이션 연산, 논리 연산, XOR 연산, 메모리 참조가 있다.[18,20]

### 4.2. DES의 주요특성

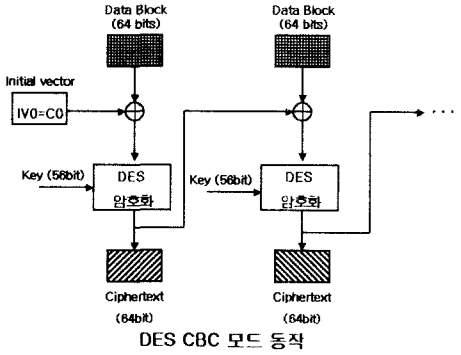
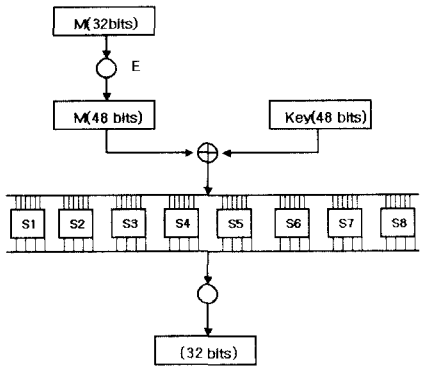



그림 5. DES CBC 모드 동작

DES는 64비트 블록단위로 암호화되며 64비트의 키 블록 중 56비트가 암호화 및 복호화에 사용된다. DES는 기본적으로 치환(P-box), 대치(S-box), 및 키 스케줄을 이용하여 16라운드 동안 동일한 동작과정

의 반복으로 이루어진다. 암호화 과정은 64비트의 평문을 초기 치환(IP) 시킨 후, 좌우 32비트씩 양분되며, 오른쪽 32비트는 다음 라운드의 왼쪽부분을 차지하게 되고 전체 16라운드의 오른쪽 부분과 암호함수 f를 적용한 결과가 전 라운드의 왼쪽부분과 XOR되어 새로운 라운드의 오른쪽 부분을 차지하게 된다. 16라운드의 처리 이후 다시 초기 순열의 역배열로 바꾸는 최종 치환부분으로 이루어진다. 암호화 함수 f는 [그림 6]과 같이 비트 확장표E 와 치환표 P 및 비선형 구조를 갖는 lookup table형식의 S-box로 구성되어 있다. 암호화 과정을 수식으로 표현하면 다음과 같다.

Li = Ri-1
Ri = Li-1 XOR f(Ri-1,Ki) (단,i는 라운드수)




F 함수구조

그림 6. F 함수구조

DES의 계산처리 비용은 각 입력단어에 대한 16라운드의 처리이다. 이 각 단계가 분석됨에 의해서 전체 성능을 결정할 수 있다. 1라운드에서 처리시간에 영향을 주는 인자로 치환, ADD 연산, 메모리 참조가 있다.[19,20]

### 4.3. DES와 MD5의 계산처리 비용 분석

DES와 MD5는 블록 단위의 처리를 수행하는 알고리즘이다. IPSec에서 데이터의 전송시 DES는 64비트마다, MD5는 512비트마다 계산을 수행하게 된다. 네트워크로 전송되는 패킷의 크기가 커질 경우 DES의 계산처리비용은 MD5에 비해 8배에 가까운 수행횟수를 실행하므로 그만큼의 호스트 자원이 소요된다. 계산처리비용을 분석하는데 있어서 DES에서 전치와 치환은 단지 비트 순열의 위치만을 변경해주는 부분이며 키 스케줄러에 의한 라운드마다의 키 생성은 최초 데이터 전송 시 한번의 수행으로 생성가능하기 때문에 데이타 전송 처리에 소요되는 계산처리비용중 수학적 계산을 요구하는 XOR연산과 메모리 참조만을 비교하여 보았다. MD5는 4라운드에 대한 총 64단계처리에 대하여,DES는 16라운드에 대하여 다음과 같은 계산처리를 수행한다.

전송 데이타 길이를 L(bit)이라고 했을 때 MD5 계산 처리비용 T(md5) 과 DES 계산 처리비용 T(des) 은 다음과 같다.

| | MD5 | DES |
|---|---|---|
| 메모리 참조 (Tm) | 1 라운드 당 T[i] 참조 16회 총 64회 | S-box 처리 시 1 라운드 당 8회 총 128회 |
| XOR (Tx) | f함수를 포함한 각 라운드 당 64+64+96+80 = 304회 | 1 라운드 당 2회식 16라운드 총 32회 |

T(md5) = L / 512 * ( 1Tm + 9.5Tx)
T(des) = L / 64 * ( 2Tm + 1Tx)

데이터의 전송크기가 증가할수록 DES에서의 블록처리 회수가 증가함에 따라 그에 따른 계산비용은 처리 횟수만큼 증가하게 되고 따라서 IPSec에서의 ESP 전송지연 증가율은 AH에서의 증가율보다 많아지게 된다.

## V. IPSec 제품의 성능비교

IPSec 제품은 IPSec을 지원하는 호스트 시스템, 게이트웨이 시스템, 라우터 VPN 서버, 방화벽 장비 및 이들 장비를 제어하는 소프트웨어를 포함한다. 이러한 장비들은 복잡성, 규모, 범위 목적에 따라 다양하기 때문에 VPN 구현시 표준안에 근거하는 적용기술과 이들에 대한 적절한 조합을 고려해야 한다. 현재 많은 수의 인터넷 장비 제조 업체에서 IPSec을 지원하는 제품들을 생산 판매하고 있으며 ICSA의 인증 제품 중심으로 그 성능을 비교해 보았다. ICSA는 항 바이러스 제품, 방화벽 IPSec/VPN 장비, 암호화 장비 및 모니터링 장비 등 네트워크 보안과 관련된 장비에 대한 인증 서비스를 해주는 곳으로 인터넷에서 다중 벤더의 VPN 네트워크가 가능하도록 표준에 기반한 IPSec VPN 기술에 대한 상호 운용성을 보증하는데 목적을 두고 있다. 현재 Cisco를 비롯하여 IBM, Lucent등 대규모 인

터넷 장비공급업체에서 생산된 IPSec 제품들이 ICSA의 장비 보증 서비스를 통해 인증 받고 있다. IPSec 버전 1.0A 장비 제품 인증 프로그램은 총 6가지로 인증 제품간 상호 운용성, IKE, IPSec 프로토콜, 암호, 인증 항목에 관하여 상세 요구, 선택기능항목을 테스트하여 현재 1.0A를 기반으로 한 IPSec장비 인증을 마친 상태이며 현재는 범주 1.0A의 선택 항목을 요구 항목으로 하여 범주 1.0B에 대한 테스트를 진행하고 있다.

ICSA에서 IPSec에 대한 인증 절차를 통과한 제품들은 기본적으로 인증, 무결성 및 비밀성 제공을 위한 기본적인 요구사항들을 만족하고, 인증 레벨별로 3-DES의 지원여부나, 다른 암호화함수 지원 여부, 압축기능여부와 최종적으로 인증 처리까지 테스트가 진행되고 있다.

대부분의 VPN장비들은 기본적으로 인터넷 표준 프로토콜인 IPSec을 지원하고 있고, 이러한 장비업체들은 보안관련 상품을 개발하는 곳으로서 기존의 방화벽 시스템에 부가적인 기능으로서 IPSec기능을 부가하거나 네트워크 장비업체로서 기존의 라우터에 IPSec기능을 추가하고 있다. 사용자 인증을 위해서 공개키 알고리즘을 사용하고 있으며 이를 위해 다른 PKI를 제공하는 회사들(VeriSign, Entrust등)로부터 서비스들과 연동되고 있다. 암호화에 사용되는 알고리즘으로는 주로 56bit-DES를 사용하고 강한 인증을 위하여 168bit-3DES가 제공되고 있으며 일부 제품에서는 암호전용의 프로세서를 지원하고 있다. 기본적으로 사용되는 인증 함수로서는 MD5 (56bit), SHA1을 지원하고 있으며 또한 키 관리에 있어서는 ISAKMP/Oakly, SKIP가 기본모드로 지원되고 있다.[16,17]

일부 제품명세서에서는 최대 사용 가능한 터널수가 명기되어 있으나 이는 전용프로세서 여부나 터널에 사용된 암호 알고리즘, 제품에서 제공하는 대역폭에 따라 다르기 때문에 절대적 수치는 되지 않으며 단지 현재 판매되는 제품들에 대하여 한정된 대역폭에서 터널사용자수를 가늠하여 볼 수 있게 한다. 방화벽 장비와 통합된 모듈에서는 non-IPSec 트래픽에 대한 bypass를 통하여 성능을 고려하였고, 일부 제품에서는 주 혹은 보조 게이트웨이를 통하여 single point failure에 대한 백업기능을 제공하고 있다. 본 논문에서 조사된 ICSA 인증 제품들 이외에도 IPSec 제품들은 여러 업체에서 다양한 규모와 사양의 제품들이 생산되고 있다.[16,17]

## Ⅵ. 성능평가 및 결과

### 6.1. IPSec 구현방식

본 논문에서는 리눅스 커널에서 IPSec의 일괄처리를 위해 새로운 버퍼를 할당하는 방식을 사용하였다. [그림 7]은 IPSec의 처리방식을 설명하는 것으로 tcp_sendmsg가 응용계층으로부터 tcp계층까지 내려오면 do_tcp_sendmeg는 ip헤더생성모듈ip_build_header()과 tcp헤더 생성모듈tcp_build_header()을 호출하게 되며 실제 메시지를 버퍼내에 복사하고 tcp_send_skb를 호출한다.

ip 헤더 생성모듈에서 정해지지 않은 IP헤더의 tot_len필드와 checksum필드는 ip_queue_xmit를 통하여 결정된다. IPSec의 처리에 필요한 값들은 앞의 함수들에 호출을 통하여 리눅스 소켓버퍼인 sk_buffer에 포함되어 이후에 IPSec 처리모듈이 호출된다.

#### 6.1.1. 일반적 일괄 처리방식

일반적 일괄처리방식에 있어서 select SPD 과정이 ip_build_header전에서 발생하는데 IPSec 구현시 기존의 ip_build_header, tcp_build_header,가 IPSec 처리에 맞게 수정되어야한다. 이는 IPSec헤더가 IP헤더 뒤에 위치해야 하므로 IP헤더가 생성된 다음 위치에 일정공간을 할당받아야 하며 IPSec헤더 생성모듈이 IP헤더와 TCP헤더 생성모듈 사이에서 호출됨을 의미한다. 또한 IPSec의 처리에 있어서는 터널모드일 경우 IP 데이타그램이 IPSec의 보호영역이므로 단편화가 이루어진 후에 IPSec 처리가 수행된다. 반면에 트랜스포트 모드에서는 프래그멘테이션 전에 IPSec 처리가 이루어진다.[6,7]

#### 6.1.2. 새로운 버퍼할당을 통한 일괄 처리방식

본 논문의 테스트를 위해 사용되는 방식은 [그림 7]과 같이 기존의 IP, TCP 헤더 생성모듈에 수정이 없이 IPSec 모듈이 단편화 여부를 결정하기 전 시점에서 헤더처리가 일괄적으로 처리되어 진다.

우선 그 패킷의 IPSec 적용 여부를 결정하기 위해 'Select SPD' 과정을 거친다. 'BYPASS' 정책이 선택되는 경우, 본래 순서대로 단편화가 필요하다면 ip_frag() 함수를 호출하고, 필요하지 않다면 dev_queue_xmit()함수를 호출한다. 'IPSEC 적용' 정책이 선택되는 경우, 그 패킷에 해당하는 SA를 찾는다. SA는 그 패킷에 적용될 IPSec 운용 모드와 IPSec의 보안 프로토콜(AH/ESP), 그리고 그 보안

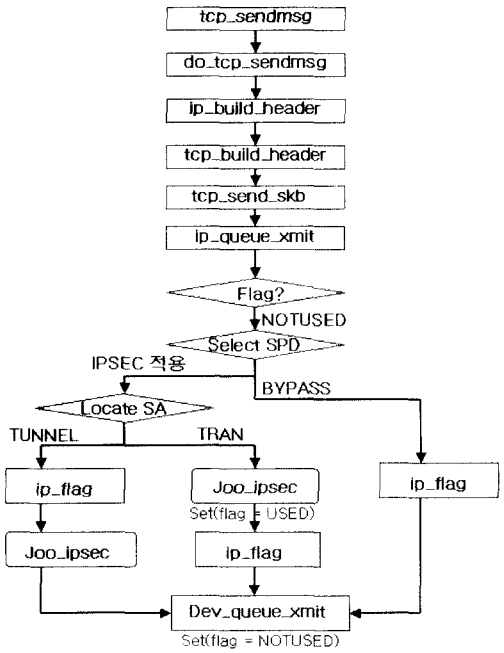



그림 7. 새로운 버퍼할당을 통한 일괄적 처리방식

프로토콜이 사용하게 될 알고리즘 등으로 구성되어 있다. 일반적 일괄처리방식에서와 같이 전송모드에 따라서 IPSec처리 모듈을 먼저 수행할 것인지를 결정하는데, 터널모드인 경우 단편화된 IP 데이터 그램을 캡슐화하므로 단편화 과정을 거친 후에 IPSec 처리를 수행하게 되고, 트랜스포트 모드인 경우 IPSec 처리를 끝낸 후 단편화 과정을 수행한다.

리눅스 소켓버퍼는 [그림 8]과 같이 채워져 있으므로 Joo_ipsec()의 첫번째 과정인 IPSec 헤더 생성을 위해서 리눅스 소켓 버퍼를 재 할당해야 한다. 재할당된 버퍼는 IPSec 헤더를 제외하고는 그 이전 버퍼의 필드들을 모두 복사한다. 특히 ESP인 경우에는 필요하다면 MSG뒤에 패딩을 하고 ESP 트레이러를 붙인다.

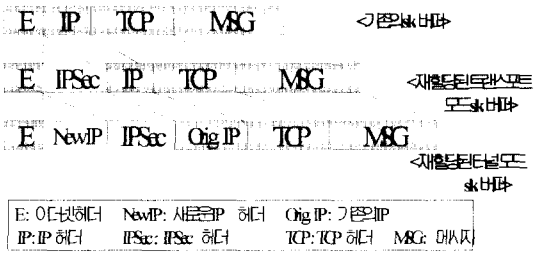



그림 8. 리눅스 소켓버퍼

두 번째 과정은 IPSec 보안 프로토콜 모드가 AH인 경우에는 인증을, ESP인 경우에는 암호화를 한다. AH인 경우 IP 헤더내의 변할 수 있는 값을 가진 필드들과 AH내의 authentication data 필드를 제외하고는 IP헤더 혹은 New IP헤더부터 메시지까지의 모든 영역을 SA내에 선택되어져 있는 알고리즘으로 압축하고, 그 압축한 값을 AH내의 authentication data필드에 넣음으로써 AH처리를 종료한다. ESP의 경우 IP헤더 혹은 TCP헤더부터 ESP 트레일러까지의 모든 영역을 SA내에 선택되어져 있는 알고리즘으로 암호화하여 재 할당된 버퍼에서는 암호화되기 이전 값들을 제거하고 암호화된 값을 ESP 헤더 뒤에 붙임으로서 ESP처리를 종료한다.

구현에 사용된 새로운 버퍼할당을 통한 처리방식의 특징은 호스트 구현방식으로 Bump in The Stack 방식과 유사하며, 순차적 헤더 생성순서를 유지하는 방식에 비해 성능에 있어서는 거의 차이를 보이지 않지만 기존의 커널 소스에 변경을 최소화하면서 트랜스포트 계층과의 독립성을 보장하고 IPSec헤더 생성 모듈과 처리모듈의 통합으로 인한 IPSec모듈의 확장성을 보장하고 있다.

### 6.2. 성능평가 환경

본 장에서는 위에서 기술한 리눅스에서 구현된 IPSec을 이용한 성능측정 방식에 대하여 기술한다. 성능 측정은 다음과 같은 환경에서 평가되어진다.

- 리눅스 커널 버전 : 2.0.35
- 구현에 사용된 언어 : C언어
- Network Interface Card Driver :
  3Com EtherLink 3 3c509TP
- 네트워크 환경: Shared 10Mbps
- Host CPU: Pentium-3 550MHz

향후 IPSec 기술에 가장 많은 응용서비스들로 사용이 예상되고 있는 SMTP, Telnet, FTP프로토콜들에 대하여 전송 패킷의 크기에 따르는 전송에 대한 처리속도를 1:1 호스트간 연결 상에서 IPSec 전송모드와 프로토콜 그리고 사용되는 암호, 인증 함수를 측정인자들로 하여 테스트하였으며 non IPSec 트래픽에 대한 상대적 처리속도를 비교하였다.[8,16]

성능평가는 트랜스포트모드에서 AH, ESP 프로토콜에서 1byte부터 4Mbyte까지의 전송크기의 점진적 변경에 따라 커널에서 IPSec모듈내 처리 시간과 전체 데이터 처리시간을 검사했으며, 커널에서의 시간 측정은 [그림 7]에서 select SPD의 전후와 각 암호, 인증 함수를 호출하는 부분의 전후에서 측정하였다.

또한 인증에 사용되는 해쉬 함수로서 128bit 키를 사용하는 SHA-1,MD5를 사용하였으며, 암호함수로서는 64비트의 키를 사용하는 DES-CBC를 사용하였다.

IPSec 구현에 있어서 SA와 키 관리 부분은 수동설정 방식을 사용하였으며, 각 전송프로토콜 별, 모드별로 SAD와 SPD의 파일을 수정하였다. SPD에는 선택자로서 송, 수신지 주소, 전송프로토콜, 송수신 포트가 기술되고, SAD에는 전송에 사용되는 암호, 인증알고리즘, 모드등의 정보들이 기술되며 데이터 전송시 관련정보를 통해 SAD를 참조하여 IPSec모듈이 동작된다.

### 6.3. 성능 평가 결과 및 분석

전송성능 측정에 대하여 평균 응답시간을 통한 처리속도를 비교수치로서 사용하였으며 커널에서 수행시간은 IPSec헤더 처리시 측정된 시간을 계산하였다.

성능 측정을 통해서 데이터 전송시 non-IPSec과 IPSec 전송특성의 비교와, IPSec 전송시 BYPASS, AH와 ESP프로토콜간 네트워크에서 처리속도에 미치는 영향을 통해 성능특성을 알아보고자 한다.

non-IPSec 전송은 IPSec 전송처리를 위한 모듈이 추가되지 않은 기존의 리눅스 커널을 사용하였다. IPSec 전송은 위에서 제안된 새로운 버퍼할당을 통한 일괄 처리방식의 모듈이 커널에 추가된 형태로서, BYPASS 정책은 전송되는 데이터에 대하여 암호, 인증의 처리가 이루어지지 않은 방식을 의미한다.

IPSEC 정책 적용은 AH 또는 ESP 프로토콜을 사용할 수 있는 데, AH 프로토콜은 인증 알고리즘으로서 MD5, SHA1를 사용하고 ESP 프로토콜은 암호알고리즘으로서 DES를 사용하였다. 데이터전송은 텍스트파일로서 1byte부터 수 메가byte까지의 전송크기로 나누어 성능을 비교해 보았다.

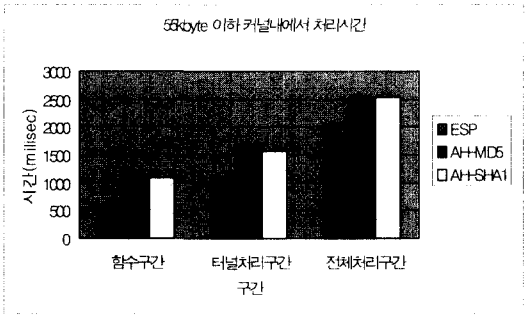



그림 9. 커널내에서의 처리시간

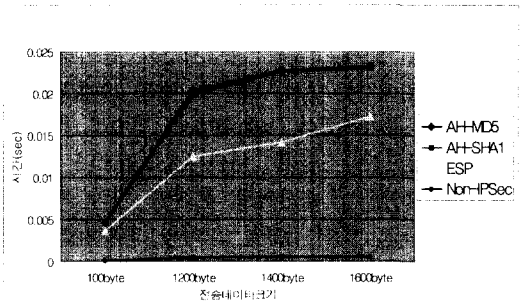



그림 10. 56Kbyte이하 FTP 처리시간

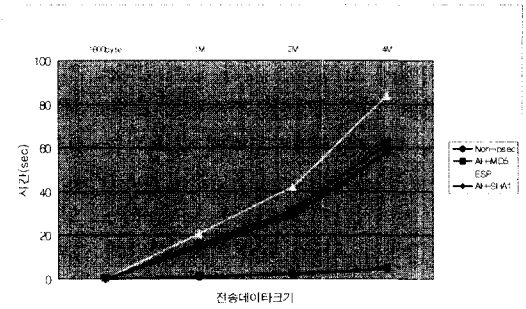



그림 11. 56Kbyte이상 FTP처리시간

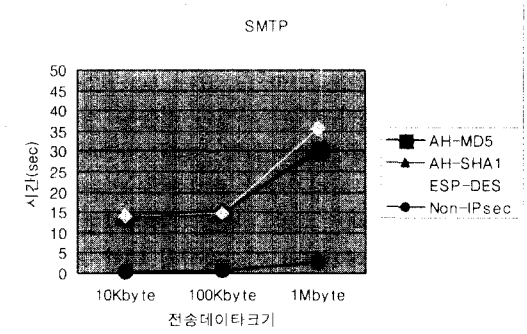



그림 12. SMTP에서의 처리시간

[그림 10], [그림 11], [그림 12]은 데이터 전송시 전송 데이타의 응답시간의 변화를 보이고 있으며 세로축은 응답시간(sec)을 가로축은 전송 데이타의 크기를 나타낸다. [그림 10], [그림 11]은 AH, ESP 전송성능에 있어서 변화를 보이는 지점으로서 전송 데이타의 크기에 따라 두 개의 그림으로 나누어 놓았다.

FTP 전송에 있어서 100byte단위부터 전송 테스트하였다. [그림 10]에서 데이터 전송방식에 대한

응답시간 비율을 보면 non-IPSec에 대하여 AH, ESP가 각각 53배,42배의 응답시간이 증가하였고, 1000byte단위에서는 AH, ESP가 각각 40배,30배의 응답시간이 증가하였다. non-IPSec 전송에 비하여 IPSec 응답시간이 높게 나타나고 있으나 이는 milisecond단위로서 초단위로 본다면 거의 차이가 나지 않는다고 할수 있다. 그러나 1000byte에서 56kbyte까지 전송량을 증가하며 전송함에 있어서 그래프에서처럼 non-IPSec전송은 응답시간의 증가율이 적은 반면에 IPSec 전송은 전송크기가 증가함에 따라 데이터에 대한 AH, ESP처리로 인해 증가율이 점차 늘어 Non IPSec전송에 비해서 응답시간이 길어지고 있다.

[그림 9]에서 보여주는 것과 같이 IPSec 전송 데이터가 IPSec 처리 모듈을 거치면서 IPSec 헤더 생성과 암호, 인증에 대한 처리에 소요되는 시간차가 데이터 전송량에 따라 영향을 주어 non-IPSec 전송에 비하여 처리시간이 증가함을 나타내주고 있다.

[그림 10]에서 AH와 ESP프로토콜간에 그래프에서는 DES보다 MD5와 SHA1에서 더 큰 응답시간이 나타나고 있다. 이는 DES 암호함수에서보다 인증함수인 MD5에서 IPSec처리를 위해 더 많은 계산처리가 이루어지고 있음을 보여주고 있으며 MD5와 SHA1간에는 거의 같은 응답시간을 보여주고 있다.

[그림 11] 에서는 전송데이타 크기가 56Kbyte 이상일 때 처리시간을 보여주고 있다. non-IPSec 전송의 경우는 수 메가크기의 파일이 전송될 경우도 전송시간이 크게 증가하지 않고 있다. 반면에 IPSec 전송에 있어서는 처리시간의 증가율이 전송데이타의 크기가 증가함에 따라 점점 높아지고 있다. 56Kbyte이상의 데이터의 전송방식에 대한 응답시간 비율을 보면 non-IPSec에 대하여 AH, ESP가 12.7배,17.6배의 처리시간의 증가를 보여주고 있다. 이는 non-IPSec전송시간에 대한 상대적 응답시간 비율로서 실제 응답시간을 보면 non-IPSec 전송 시 수 메가 파일을 전송할 때 수 초후에 전송이 완료가 이루어지는 반면 IPSec 전송시에는 수십 초 후에야 데이터 전송이 완료되고 있다. 전송 데이타의 크기가 증가할수록 non-IPSec 전송에 비하여 IPSec 전송에서 현저한 성능저하를 보여주고 있다. 위에서 언급했던 암호, 인증알고리즘의 분석에서 보여주는 것과 같이 전송 데이타가 증가함으로서 커널 내부에서 IPSec 헤더 생성과 암호, 인증 함수의 처리 횟수의 증가로 전송 패킷당 IPSec처리로 인해 지연시간이 누적되는 것이 원인이다.

[그림 11]에서는 1000byte단위의 전송결과에서와는 다르게 ESP프로토콜에서의 DES암호함수 처리시간이 AH프로토콜 MD5의 처리시간보다 많이 소요되고 있다. 커널에서 IPSec처리에 있어서 DES는 56비트 블록단위로, MD5는 512비트 블록 처리를 하기 때문에 전송 데이타의 크기가 증가함에 따라 DES에서 계산시간을 요구하는 더 많은 처리가 수행되어 AH 프로토콜 인증함수인 MD5에서 보다 많은 처리시간이 소요되고 있음을 보여주고 있다.

그림에는 나타나지 않지만 10Mbyte 정도의 데이터 전송시는 전송호스트의 Hanging현상도 나타났으며 이는 ESP, AH프로토콜을 위한 계산처리를 위해 CPU의 모든 자원이 점유됨으로써 야기되는 현상이다.

[그림 12]에서는 SMTP 전송 시 전송 데이타 크기에 따른 성능 변화를 보여주는 것으로 non-IPSec 전송에 비하여 IPSec 전송시간이 AH, ESP 경우 35배, 30배의 처리시간이 증가하였으나 1M의 파일을 전송하였을 경우 11배,13배의 처리시간이 증가하였다. FTP에서와 같이 전송파일의 크기가 증가할수록 non-IPSec전송에 비하여 IPSec모듈의 처리에 대한 심한 지연이 나타났다.

## Ⅶ. 결 론

본 논문은 IP계층의 VPN인 IPSec에서 전송 프로토콜과 모드에 따라 응용계층 프로토콜들의 전송에 관한 성능을 테스트하였다. IPSec은 노드간 또는 호스트간 정보보호와 안전한 운용에 대하여 IP 계층에서 구현된 보호서비스를 이용할 수 있게 한다. 하지만 IPSec에서 사용하는 AH와 ESP의 사용은 IP 프로토콜의 처리비용의 증가로 인해 보안성능은 증대시키지만 처리성능을 감소시킬 수 있다. 이점을 감안하여 성능 테스트를 한 결과 non-IPSec전송에 비하여 ESP와 AH에서 사용하는 암호/인증 함수에 의해 전송에 대한 처리지연이 나타났으며 수백 바이트 범위에서는 인증 함수에서, 수천 바이트 이상에서는 암호함수에 의한 더 많은 처리 지연이 나타났다. 이는 전송량이 증가함에 따라 MD5에 비해 DES에서의 IPSec처리를 위한 수행횟수의 증가율이 높기 때문이다. 증가율의 차이로 전송량이 증가할수록 IPSec전송은 non-IPSec 전송에 비하여 더욱 심한 지연차이를 나타냈다.

성능테스트 결과 전송패킷크기와 사용하는 암호, 인증함수, 호스트의 CPU속도, IPSec의 구현방식이

IPSec 전송성능에 영향을 주는 인자로 나타났다. telnet 경우 1byte단위의 통신방식을 사용하기 때문에 IPSec전송에 의한 처리시간 지연은 적다. 그러나 파일전송의 경우 전송용량이 증가하게되면 10여배 정도의 처리지연이 나타나고 있다. 축적 전송방식의 SMTP 경우에서도 전송량이 증가할수록 데이타를 물리계층으로 내보내기까지의 지연시간이 증가했다.

본 논문의 성능테스트에서처럼 IPSec의 구현이 호스트 기반, 라우터 기반 구현됨에 있어서 전송량의 증가에 따라 CPU가 암호, 인증 처리로 인해 모든 CPU자원을 점유하게되어 hanging현상을 유발할 수 도 있다. IPSec의 도입이 기존 네트워크에 부가적인 서비스형태로 제공되어질 때 보안 강도의 유지와 기존 서비스들에 영향을 최소화하는 CPU성능을 유지하거나 부가적인 전용 암호화 모듈,VPN 전용장비 등이 필요하다.

IPSec 모드에는 AH, ESP, Bypass 모드가 있으나, 본 논문에서는 IPSec을 통한 전송 데이타의 보호에 있어서 처리시간을 non-IPSec트래픽에 대하여 비교하였다. non-IPSec트래픽과 Bypass 모드의 전송처리시간은 구현방식에 따라 차이가 날수 있음을 고려해야 한다. 본 논문에 구현된 Bypass 모드는 Non IPSec 전송에 비해 3-4배의 전송시간이 더 걸렸으며 이는 IPSec 모듈의 추가에 따른 커널 루틴의 변경 때문이었다.

대용량의 트래픽 전송에서는 Non IPSec에 비하여 10여배 이상의 전송지연이 발생하였다. 그러므로 기존 네트워크에 VPN적용은 수백 바이트 이하 단위의 인터넷 트래픽에 적절한 방식이며 지연에 민감한 동영상 등의 대용량 전송방식에는 적합하지 않다.

## 참 고 문 헌


[1] S. Kent, R. Atkinson, "IP Authentication Header", RFC 2402, November 1998

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

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

[4] C. Madson, R. Glenn, "The Use of HMAC-MD5 within ESP and AH", RFC 2403, November 1998

[5] C. Madson, R. Glenn, "The Use of HMAC-SHA1 within ESP and AH", RFC 2404, November 1998

[6] Michael Beck, Harald Bohme,Mirko Dzidzka, Ulrich Kunitz, Rover Magnus,Dirk Verworner, "Linux Kernel Internals", SecondEdition, Addison-Wesley, pp227-247, 1998

[7] Gary R. Wright ,W.Richard Stevens, "TCP/IP Illustrated, Volume 2(1)", Addison Wesley, pp205-246, Jaunary 1995

[8] Charlie Scott, Paul Wolfe&Mike Erwin, "Virtual Private Network",pp135-161 O Reilly, 1999

[9] Dave Kosiur, "Virtual Private Network", WILEY, 1998

[10] Steven Brown,"Implementing Virtual Private Networks", McGraw-Hill, pp97-140, 1999

[11] Naganand Doraswamy, Dan Harking, "IPsec", Prentice Hall PTR, 1999

[12] 권윤주, 김현주, 정태명, "리눅스상에서의 IPSec 구현에 관한 연구", 한국 정보처리학회 추계 학술 발표 논문집 제6권 2호,1999

[13] Brian Quirton, "The CASE from packeting VPNs", IEEE computer Society (TELEPHONY), 1999

[14] Wray West, "The ABC of Remote Access VPNs", IEEE Computer Society (BUSINESS COMMUNICATIONS REVIEW), 1999

[15] Thomas J. routt and Jerry A. keterly, "Securing VPNS: Architectural Approach", COMMUNICATIONS Review, 1999

[16] http://etlars,etri.re.kr/Etlars/industry/jungidong/952/95304

[17] http://www.icsa.net/html/communities/ipsec/certification/certified_products/index.shtml

[18] Larry D. Barchett, Arindam Banerji, John M. Tracey and David L. Cohn, "Problems using MD5 with IPv6", PERFORMANCE EVALUATION An international Journal, 1996 [19]

[19] Eli Biham, "A Fast New DES Implementation in Software", Technion-Israel Institute of Technology,1997

[20] William stallings,"Network and internet Security Principles and practice, Prentice-Hall,

1995
[21] 이경근, 오채형, "인터넷 VPN서비스 기술 동향", 한국통신학회지 THE PROCEEDINGS OF THE KOREAN INSTITUTE OF COMMUNICATION SCIENCES 1999, 8 v.16, n.8, pp.81-91

정 태 명(Hyung J. Lim)
1981년 : 연세대학교 전기공학(학사)
1984년 : University of Illinois Chicago, 전자계산학과 학사
1987년 : University of Illinois Chicago, 컴퓨터공학과 석사
1995년 : Purdue University, 컴퓨터공학 박사
1985년~1987년 : Waldner and Co., System Engineer,
1987년~1990년 : Bolt Bernek and Newman Labs., Staff Scientist
현재 : 성균관대학교 전기 전자 및 컴퓨터공학부 부교수
<주관심 분야> 실시간 시스템, 네트워크관리, 네트워크 보안, 침입 탐지 시스템, 시스템보안, 전자 상거래 보안

임 형 진(Yoon J. Kwon)
1998.2 : 한림대학교 컴퓨터공학과(학사) 졸
2001.8 : 성균관대학교 정보통신대학원(석사) 졸
2001.7 : 하이콤정보통신 기술연구소
<전공> Network Security

권 윤 주(Tai M. Chung)
2000.2. : 성균관대학교 정보공학과(학사) 졸
2000.3~현재 : 성균관대학교 전기전자 및 컴퓨터공학부 석사 과정
<전공> Network Security