ECC기반 VPN 터널링 프로토콜 설계

최은실*, 이병관**, 정은희***
*관동대학교 전자계산공학과
**관동대학교 컴퓨터공학과
***삼척대학교 경제학과
tosil17@hanmail.net, bklee@kwandong.ac.kr, jeh@samcheok.ac.kr

VPN Tunneling Protocol Design based on ECC

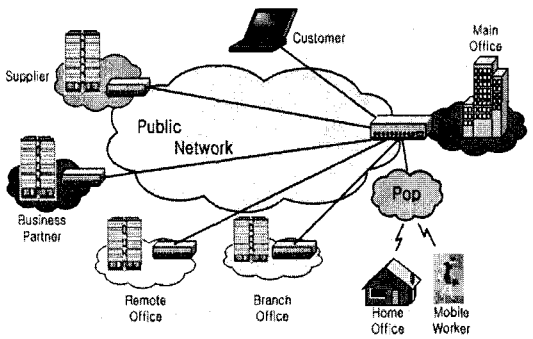
Eun-Sil Choi*, Byung-Kwan Lee**,Eun-Hee Jung***
*Dept of Computer Science, Kwandong University
**Dept of Computer Engineering, Kwandong University
***Dept of economic Science, Samcheok National University

요 약

본 논문은 네트워크계층에서 IP패킷을 송·수신자간에 사전에 합의된 알고리즘을 이용하여 암, 복호화하는 통신 터널링 프로토콜로, 공유비밀키 생성을 위해 ECC((Elliptic Curve Cryptosystem)알고리즘을 사용함으로서 키 생성시간의 단축과 보안 강도을 강화시켰다. 또한, 공유비밀키 교환을 위해서 타원곡선을 이용한 EC-DH(Elliptic Curve Diffie Hellman)알고리즘을 사용하고, IP패킷의 무결성 검증과 인증을 위해 HMAC-SHA-1알고리즘, 패킷을 암·복호화하기 위한 대칭키 알고리즘인 DES를 사용하였다.

1. 서론

오늘날 통신 환경의 급속한 발달과 기업내의 외부 네트워크와의 정보 교환 수요의 증가, 재택 근무자나 이동중의 업무처리의 증가로 인하여 네트워크 범위가 확대되어가고 있으며 그에 따른 안전한 통신을 위한 보안 문제와 QOS(Quality of Service)제공이 크게 대두되고 있다. 이러한 시점에서 기업이 여전히 전용망을 구축하여 운용하는 일은 많은 비용이 들고, 분산된 기업 구조와 환경에서 여러 지점간(intranet)이나 협력사를 연결(extranet)하는 경우에 더욱 강력한 보안이 필요하게 되었다. 따라서 많은 기업들과 통신 사업자들은 공중망상에서 터널링 프로토콜을 적용한 가상사설망(VPN)을 구축하게되었다. VPN은 공중 통신망 기반시설을 터널링 프로토콜과 보안 서비스를 사용하여 개별기업의 목적에 맞게 구성한 데이터 네트웍이다. 이것은 공중망을 이용하기 때문에 네트워크 구축비용을 줄일 수 있으며 회선 관리도 공중망의 운용자가 맡게 되어 전체적인 비용 절감 및 인력 절감 효과를 얻을 수 있다. 하지만 이러한 공중망 형태의 네트워크에서도 해킹 공격이나 바이러스에 노출될 수 있으며, 언제든지 예상하지 못했던 장애가 발생할 수 있다. 따라서 이러한 문제를 해결하기 위해 본 논문에서는 네트워크계층에서 송·수신자 사이에 사전에 합의된 알고리즘으로 IP패킷을 암호화하여 전송함으로 더욱 강화된 VPN 터널링 프로토콜을 제안한다.

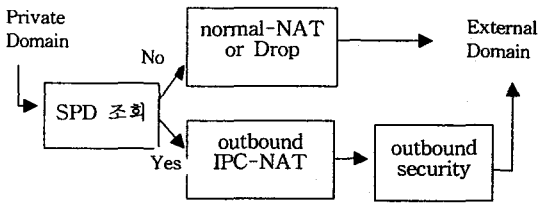


[그림 2-1] 공중망을 이용한 VPN 구조

2. IPC-NAT(Network Address Translation)

기업은 ISP(Internet Service Provider)로부터 할당받은 한정된 IP주소를 이용하여 인터넷에 접속한다. 이러한 경우에 기업내에서 내부(private) IP의 보호 차원에서 private 주소를 쓰는 경우가 많다. 기업내부에서 사용하는 내부 IP주소는 내부에서는 통신이 가능하지만 외부망(external network)으로는 라우팅이 안된다. 따라서 내부 IP주소를 외부 IP주소로 바꾸어 주는 장치인 NAT가 필요하게된다. NAT는 NAT 테이블에 의하여 IP헤더의 소스 IP주소를 외부 IP주소로 변환시켜 목적지로 전송하고 들어오는 패킷의 목적지 IP주소는 다시 내부주소로 변환하여 내부망의 목적지로 전달된다. 본 논문에서 제안하는 IPC-NAT는 네트워크계층에서 IPsec 기능을 사용할 수 있는 IPsec Policy Controlled NAT라고 하며, 양단의 게이트웨이의 터널모드로 동작한다.

터널모드 송신 패킷의 IPC-NAT 수행절차는 [그림 2-2]와 같다.



[그림 2-2] 송신 패킷에 대한 IPC-NAT

① Outbound SPD를 조회하여 내부망에서의 IP패킷이 보안이 적용될 패킷인지를 확인한다.

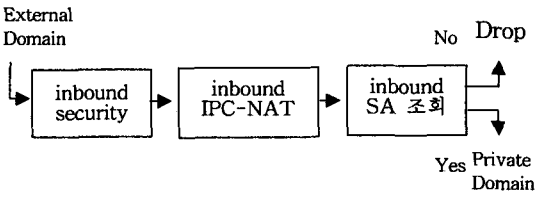
② 보안이 적용될 패킷이라면 내부 IP주소를 사용하여 IPC-NAT을 적용한 후, 외부 IP주소를 New IP 주소로 사용한다. 보안이 적용되지 안는 패킷이라면 normal-NAT를 수행하거나 Drop한다.

③ outbound 보안을 위해 IKE(Internet Key Excange)가 송·수신자간에 SA(Security Association)를 협상하여 협상값을 각각 SAD(Security Association Database)에 저장한다.

④ 송·수신자가 상호 합의한 인증알고리즘으로 인증값을 계산하고, 암호화 알고리즘으로 IP패킷을 암호화한다.

⑤ 암호화된 IP패킷을 New IP header의 목적지 주소의 외부망으로 전송한다.

송, 수신자간의 데이터를 전송하기 위한 터널모드 수신 패킷의 IPC-NAT 수행절차는 [그림 2-3]과 같다.



[그림 2-3] 수신 패킷에 대한 IPC-NAT

① 외부망에서 들어오는 패킷은 먼저 바깥쪽의 IP헤더를 제거한다(Detunnel).

② 바깥쪽 헤더가 제거된 IP패킷의 내부 IP를 이용하여 IPC-NAT를 적용한다.

③ 내부 IP주소를 보고 사전에 송·수신자간에 합의된 SA를 조회하여 만족하는지에 따라 내부망으로 라우팅 할지를 결정한다.

3. ECC 공유비밀키 생성 알고리즘

$GF(2^m)$상의 타원곡선 $E: y^2+xy=x^3+ax^2+b$, $a, b \in GF(2^m), b \neq 0$의 방정식을 만족하는 $GF(2^m)$상의 모든 점 (x, y)와 무한 원점 0로 구성되고, 이들에 대해 다음과 같은 덧셈 법칙을 만족한다.

(1) $0+0=0$

(2) 모든 점 $P \in E$ 에 대해 $0+P=P+0=P$를 만족한다.

(3) $P=(x_p, y_p)$일때, $-P=(x_p, x_p+y_p)$이고 $P+(-P)=0$이다.

(4) $P=(x_1, y_1), Q=(x_2, y_2)$라고하면 $P+Q=(x_3, x_3)$다음과 같다

$$\text{단계 [1]}\ L = \begin{cases} (x_1+x_2)^{-1}.(y_1+y_2) & P \neq Q \\ x_1^{-1}.y_1+x_1 & P=Q \end{cases}$$

$$\text{단계 [2]}\ x_3 = L^2+L+(x_1+x_2)+a$$

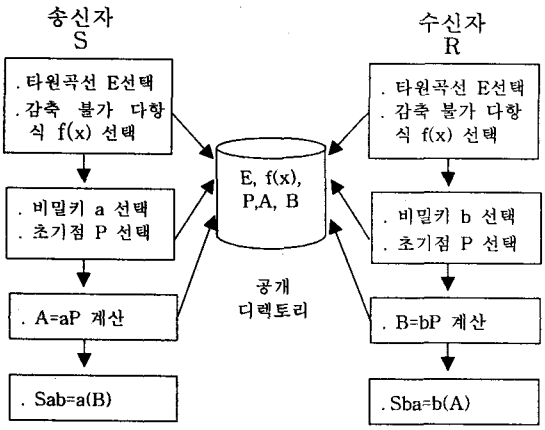
$$\text{단계 [3]}\ y_3 = \begin{cases} L.(x_1+x_3)+x_3+y_1 & P \neq Q \\ x_1^2+(L+1).x_3 & P=Q \end{cases}$$

2^m개의 원소를 포함하는 유한체 $GF(2^m)$은 주어진 m에 대해서 다항식 기저(polynomial basis)로 원소를 표현할 수 있다. 다항식 기저는 m차의 감산 다항식 f(x)에 의해 유일하게 결정되는 $\{x^{m-1}, x^{m-2}, \ldots, x, 1\}$형태의 다항식 집합이다. 감산 다항식으로는 구현상의 효율을 위해 3개의 항으

로 구성된다.

4. EC-DH 알고리즘

EC-DH 알고리즘은 키 교환을 위해 타원곡선알고리즘을 이용하여 공개키와 비밀키를 생성한 후, 상대방의 공개키를 이용하여 공유비밀키(shard secret key)를 생성하는 알고리즘이다. 즉 송·수신자는 타원곡선 알고리즘을 이용하여 비밀키와 공개키를 각각 생성한 후, 공개 디렉토리에 공개키를 등록하면 송·수신자의 공개키를 각각 이용하여 자신의 비밀키와 덧셈연산 하여 공유 비밀키를 생성한다. 아래의 [그림 2-4]은 EC-DH 알고리즘의 공유 비밀키 생성 과정을 나타낸다.



[그림 2-4] EC-DH 공유 비밀키 생성과정

[S,R] 송신자 S와 수신자 R 양쪽은 타원곡선 E; $y^2+xy=x^3+\alpha x^2+1$을 선택하고, 감축 불가 다항식 $f(x)=x^7+x+1$을 선택한다.

[S→R] 송신자 S는 타원곡선 알고리즘을 이용하여 비밀키 a와 초기점 P를 생성한 후 a만큼 덧셈 연산하여 aP를 계산한 후 공개 디렉토리에 등록한다.

[R→S] 수신자 R는 타원곡선 알고리즘을 이용하여 비밀키 a와 초기점 P를 생성한 후 b만큼 덧셈 연산하여 공개키 bP를 계산한 후 공개 디렉토리에 등록한다.

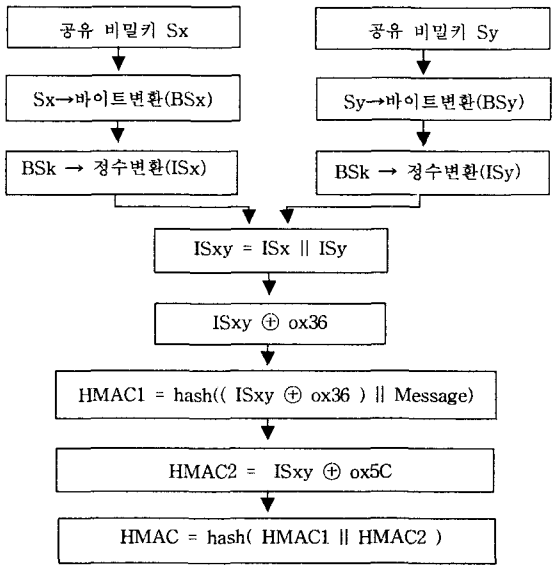
[S] 송신자 S는 수신자 R가 등록한 B를 이용하여 송신자의 비밀키 a만큼 덧셈 연산하여 공유 비밀키 S_{ab}를 계산한다..

[R] 수신자 R는 송신자 S가 등록한 A를 이용하여 수신자의 비밀키 b만큼 덧셈 연산하여 공유 비밀키 S_{ba}를 계산한다.

[S,R] 송신자 S와 수신자 R는 공유 비밀키 S_{ba}를 안전하게 공유한다.

5. HMAC-SHA-1 알고리즘

해쉬 함수가 단지 무결성만을 확인 하는데 비해 HMAC-SHA-1 알고리즘은 메시지와 공유 비밀키를 결합하여 메시지 인증 코드(MAC)를 생성한다. 따라서 상대방의 신원을 확인할 수 있는 인증기능이 포함되었다 할 수 있다. 전자상거래를 수행하기 전에 키 생성 알고리즘을 합의하고, 합의한 알고리즘으로 공유비밀키를 생성한다. 송신자는 본인의 공유비밀키와 메시지를 결합하여 메시지 인증 코드를 생성한 후 상대방에게 전송한다. 수신자는 전송된 메시지에 해쉬함수를 이용해 메시지 인증코드를 생성하여 전송된 MAC와 비교한다. 만약 두 값이 일치하면 전송중에 메시지에 변조가 없다는 것을 검증할 수 있다. MAC의 흐름도는 [그림 2-5]와 같다.



[그림 2-5] HMAC-SHA-1알고리즘 흐름도

① F_2^m으로 생성된 공유비밀키 Sx, Sy를 바이트로 변환하여 BSx, BSy 변수에 기억시킨다.

② 바이트로 변환된 공유비밀키 BSx, BSy를 다시 정수로 변환하여 ISx, ISy 변수에 기억시킨다.

③ ISx, ISy값을 각각 연결하여 ISxy 변수에 기억시킨다.

④ 정수로 변환된 ISxy가 해시 함수의 1Block 크기인 64바이트보다 작으면, 64바이트가 되도록 끝

에 0을 채운다. ISxy가 64바이트 보다 큰 경우에는 ISxy를 해시함수를 수행하여 20바이트가 출력되도록 한 후, 나머지 44바이트를 0으로 채운다. 그 후 16진수 36과 Exclusive-OR 연산을 수행한다.

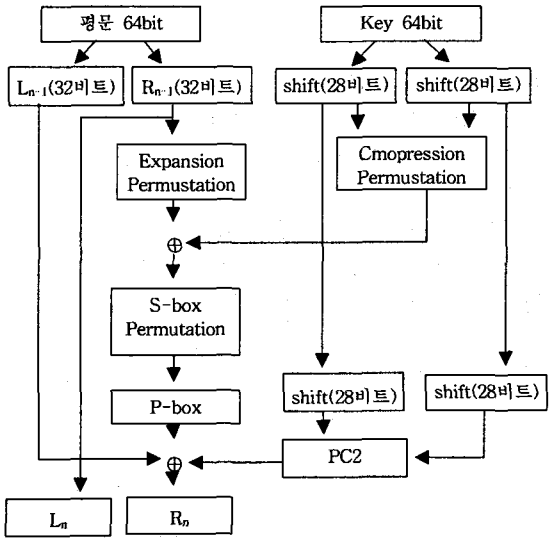
⑤ ④번의 결과와 원래의 데이터를 결합시킨 후 해시 함수로 다이제스트 연산을 수행하면 160비트의 메시지 다이제스트 값 HMAC1이 생성된다.

⑥ 정수로 변환된 ISxy변수 값과 16진수 5C와 Exclusive-OR 연산을 수행하고 메시지 다이제스트 값 HMAC2를 생성한다..

⑦ 마지막으로 ⑤의 결과값과 ⑥번의 값을 다시 한번 해시 함수(sha-1)로 다이제스트 연산을 수행하여 최종값인 160비트, 즉 20바이트의 메시지 인증 코드값이 생성된다.

6. DES 알고리즘

DES의 구조는 크게 암호화 처리 과정과 암호화 키를 생성하는 부분으로 구분할 수 있다. 먼저 키 생성과정은 최초의 64비트키가 PC-1(Permuted Choice-1)표에 따라 전치(permuted)되면 56비트로 키가 생성되며, 이것은 각각 28비트 크기의 C0와 D0로 나뉘어져 좌측 시프트를 수행하고, PC-2테이블을 적용하여 1라운드의 키(k1)로 사용될 C1과 D1으로 구해진다. 암호화 과정은 먼저 입력된 평문을 초기 치환 IP(Initial Permutation)테이블에 따라 각각 32bit L0와 R0로 나누고 16회의 라운드 함수를 적용한다. 라운드 함수인 f함수는 먼저 32비트 입력 데이터를 확장 테이블(PE)를 사용하여 48비트로 선형 확장한다(E(R0)). 그 후 키 스케줄에 따라 입력된 48비트 라운드 키(Kn)와 선형 확장된 48비트 데이터를 Xor 비트 연산을 하며 연산이 끝난 48비트 데이터는 각각 6비트 입력에 4비트 출력을 갖는 8개의 s-box를 거쳐 32비트로 변환 출력된다. 마지막으로 16라운드 처리 후 출력된 최종 데이터는 결합되어 64비트 데이터의 형태로 변환된 후 역 초기 치환 테이블(IP-1)을 거쳐 출력됨으로 1블럭의 암호문으로 완성된다. 복호화는 암호화과정과 같은 알고리즘을 사용하고, 라운드 키를 적용한다. [그림2-5]은 DES의 1라운드 암호화 과정을 보여준다.



[그림 2-6]DES 1라운드 흐름도

7. 결론

네트워크 계층에서 터널링 프로토콜을 사용하여 VPN을 형성하기 위해서는 먼저 IKE가 송·수신자 사이에서 SA값을 협상해야한다. 합의된 SA정보를 이용하여 IP패킷에 대한 인증값과 암호화를 위해서 먼저 공유비밀키를 생성하는데, 본 논문에서 사용한 ECC알고리즘은 RSA의 키 크기가 1024비트인데 비해 단지 160비트의 크기만을 요구하기 때문에 그만큼 대역폭을 적게 차지하게 되며, 이산대수문제를 기반으로 덧셈연산을 수행하므로 곱셈의 멱승으로 키생성을 하는 RSA보다 키 생성 시간이 10배 이상 빠르다. 따라서 ECC 알고리즘으로 생성된 공유비밀키를 사용하는 HMAC-SHA-1과 DES알고리즘의 수행속도 역시 빨라지게 되며, 보안 강도도 향상되었다.

참고문헌

[1] http://rfc-2663.rfc-list.com/rfc-2663-12.htm.

[2] http://www.faqs.org/rfcs/rfc2709.html.

[3] http://arcane.4wish.net/mirror/kordoc/network/network_whitepaper-3.html.

[4] 공학기술논문집 Vol.7 1998.8, "타원곡선 공개키 알고리즘의 효율성", pp.295.

[5] 한국항공대학교 논문집 제38집 "타원곡선 암호 시스템에 관한 연구", pp.205~211.

[6] 한국정보통신기술협회, "TTAS.KO-12.0015",pp. 8~11, pp.14, pp.16~20.

[7] 신성규, "전자서명을 위한 HMAC 알고리즘 구현", pp.27~28.