2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

IPSec을 위한 새로운 키 복구 시스템

김정범 이윤정 박남섭 현은실 김태윤
고려대학교 컴퓨터학과
{qston, genuine, nspark, eunsil, tykim}@netlab.korea.ac.kr

New Key Recovery System For IPSec(NKRS_IPSec)

Jeong-Beom kim, Yun-Jeong Lee, Tai-Yun Kim
Dept. of Computer Science and Engineering, Korea University

요 약

최근 네트워크 보안에 대한 사용이 증가하고 있다. 그러나 암호는 본래 가지고 있는 키 관리의 어려움 때문에 여러 가지 문제가 발생할 수 있다. 이러한 암호의 사용이 야기하는 역기능을 해소하고 순기능을 조장하기 위하여 키 복구 개념이 도입되었다. 이러한 키 복구 방법 중 본 논문에서는 캡슐화 방식을 사용한다. 하지만 이 방식에서의 문제점은 키 협상 부분에서 키 복구에 대한 부분도 협상해야 한다. 그렇기 때문에 복잡한 키 협상 과정이 더욱 복잡해지며 마찬가지 이유로 SPD(Security Policy Database) 역시 복잡해진다. 그리고 키 복구에 대한 정보를 정해진 시간을 주기로 계속 보내야 하기 때문에 네트워크 소스 역시 많이 소비된다. 이러한 점을 해결하고자 본 논문에서는 TOS(Type Of Service)의 4bit를 이용하여 한 번의 키 복구 정보를 보내고, 사용자가 이러한 서비스를 자신이 사용하고자 할 경우에만 선택할 수 있도록 함으로써 해결하였다.

1. 서론

IPSec(IP Security)[1]은 IETF에 의해서 IP 계층 보안을 위한 개방 구조로 설계되고 있다. IPSec은 네트워크 계층의 보안에 대해서 안정적이고 영구적인 기초를 제공한다. IPSec은 오늘날의 암호화 알고리즘을 수용할 수 있을 뿐만 아니라 새로운 알고리즘을 수용할 수 있다.

하지만, 네트워크에서 암호의 사용은 정보의 누출 및 오용을 방지하고 상대의 신원 확인을 가능하게 함으로써, 온라인 상에서의 전자상거래나 전자계약을 가능하게 하는 등 많은 장점을 가지고 있다. 그러나 암호는 본래 가지고 있는 키 관리의 어려움 때문에 다음과 같은 문제가 발생할 수 있다. 첫째, 키의 분실이나 손실로 인해 사용자가 자신의 키(또는 암호문)에 접근할 수 없는 경우이다. 이 경우에는 자신이 적법한 소유자임에도 불구하고 자신의 정보에 대하여 접근을 할 수 없음으로 해서 많은 손실을 가져올 수 있다. 둘째, 국가가 범죄 수사 등의 적법한 이유로 키에 접근해야 할 필요성이 있을 경우에 발생하는 문제점이다. 범죄자는 암호문을 사용함으로써 합법적인 수사를 방해할 수 있다. 셋째, 암호가 오용됨으로써 발생할 수 있는 잠재적인 위협이다. 사업장에서 피고용인이 중요한 정보를 암호화하고 키를 담보로 금품을 요구할 수도 있으며, 키의 도난이나 손상 등의 위협이 항상 존재한다.

지금까지 논의한 바와 같이 암호의 사용이 야기하는 역기능을 해소하고 순기능을 조장하기 위하여 키 복구 개념이 도입되었다.

현재까지 제안된 암호 키 관리 방식은 크게 위탁(escrow) 방식과 캡슐화 방식, TTP(Trusted Third Party) 방식으로 나눌 수 있다.

기존의 IPSec에서는 이러한 종류의 키 관리 방식이 도입되지 않았고, 또한 SPD(Secure Policy Database)를 두어서 관리자의 의도대로 사용자는 따를 수밖에 없는 상황이었다. 이에 본 논문은 관리자가 만들어 놓은 서비스가 아닌 사용자의 입장에서 IPSec을 개발하는게 목적이며, 키 복구 방식을 도입하게 된 경위도 사용자의 입장에서 보다 안전하다는 확신을 줄 수 있게 하기 위해서다.

본 논문에서 제안한 IPSec을 이용한 터널 구조는 이러한 키 복구 개념을 도입하여 사용자 측에서 간단한 방법에 의해 원하는 보안 서비스를 받을 수 있도록 하기 위해서 IP의 4bit를 이용함으로써 사용자 측에서 서비스를 나누어서 사용할 수 있게끔 하였

고, 키 관리 방식 3개중 캡슐화 방식을 이용하여 구상 및 설계를 하였다.

2. 관련연구

2.1 KRH(Key Recovery Header)[4]

KRH란 KRB(Key Recovery Block)[5]를 IP 데이터그램에 키 회복 정보를 덧붙임으로 인해서 키 회복 능력을 제공하기 위해서 사용된다.

KRH를 네트워크의 하부구조의 변경 없이 사용하기 위해서 key 복구 데이터를 IP 페이로드에 넣어서 전송된다. 그렇기 때문에 KRH를 사용하지 않는 시스템은 KRH를 무시한다. 이러한 KRH의 위치는 IPv6를 사용할 경우 KRH는 일반적으로 End-to-End와 AH(Authentication Header)[2] 뒤, 그리고 ESP(Encapsulation Security Payload)[3]와 transport Header전에 위치한다. IPv4일 경우 AH 뒤, ESP 앞에서 사용한다. 이러한 KRH는 키 복구를 수행하기 위해 덧붙여진 KRB를 네트워크를 통해서 전송하고, 받을 수 있게 설계되었다. 이러한 이유로 KRH는 ESP에 의해서 암호화되어서는 안 된다. 즉, KRH는 암호화되어진 부분인 ESP 부분에 대한 SA(Security Association)의 정보를 담고 있다. 그러므로, KRH는 ESP와 항상 함께 결합하여 사용되어진다. KRH는 ESP Header를 포함하고 있지 않는 데이터그램에서는 전송할 수 없다. 그리고 이것을 비인증된 변형으로부터 보호하기 위하여 AH를 사용할 것을 권고한다. 하지만 반드시 AH와 함께 사용되어지는 것은 아니다.

ESP를 위한 SA를 확립한 엔티티들은 KRH에 대한 SA 역시 확립해야만 한다. 이러한 KRH에 SA 확립이란, KRH가 모든 IPSec 패킷의 부분으로 덧붙여져서 전송되는 것이 아니라 IPSec 디바이스의 로컬 정책에 따라 전송 빈도수가 정해지는 것이기 때문에 이러한 정책에 대한 SA 확립을 의미한다.

우선적으로 sender가 KRH를 포함하는 패킷을 보내기 위한 KRH의 AD(authentication data)를 산출한다. 그 다음에 receiver가 그것을 받고 KRH의 AD를 검출한다. 즉, KRH를 받은 IPSec 엔티티는 KRH의 무결성을 검증하여 무결성이 깨졌다면 타당한 KRH가 도달 할 때까지 그 뒤의 IPSec 패킷을 폐기한다.

3. IPSec을 위한 새로운 키 복구 시스템 (NKRS_IPSec)

앞에서 언급한 대로 KRH는 SA 과정을 복잡하게 하고, 제한 시간마다 KRH를 보낸다. 이 과정은 네트워크 리소스를 많이 소비하므로 매우 비효율적이며, 이 서비스를 로컬 정책에 의해 강제로 시행한다는 것은 사용자 측면에서 역시 비효율적이다. 이 문제를 해결하기 위한 방안으로 본 논문은 IP Header의 TOS(Type Of Service) 필드의 4비트를 이용한다. 4비트로 구분한 이유는 사용자가 원하는 보안 서비스를 받을 수 있게 하기 위함이며, KRH 서비스를 시행할 경우 오직 한 번의 패킷을 보냄으로써, 네트워크 리소스를 효과적으로 이용할 수 있기 때문이다.

즉,TOS의 맨 앞 비트는 패킷이 Host에서 서비스를 요청하는 것인지, 아니면 SG에서 IPSec이 처리된 패킷인지 구분하기 위함이며, 두 번째 비트는 설정 여부에 따라 AH 서비스의 유무를 뜻하고, 세 번째 비트 역시 동작원리는 같으며 KRH를 의미하는 것이고, 네 번째 비트 역시 동작원리는 같으며 ESP를 의미한다. 본 문에서는 KRH를 쓸 경우 보안을 위하여 AH를 꼭 쓰게 만들어서 AH가 KRH의 무결성을 검증할 수 있도록 하였다. TOS가 0이나 이 외에 설정은 일반 패킷으로 간주한다.

기존의 키 복구는 사용자가 원하지도 않는데 키 복구 정보를 매 주기마다 계속 보냄으로 인하여 사용자가 신뢰할 수 없었다. 따라서 본 논문에서는 사용자가 서비스를 요구하면 해당 서비스에 대한 헤더를 페이로드에 삽입하여 전달함으로써 모든 정보가 세션이 끝날 때까지 키 복구 정보를 이용할 수 있다. 이로 인해 네트워크 리소스의 효율적인 사용과 보안 문제점에 대한 것을 해결하였다. 가장 중요한 이점은 사용자가 자신의 데이터에 대한 자신이 보안 서비스를 평가하여 키 복구등과 보안 서비스등과를 할 수 있기 때문에 키 복구에 대하여 더욱 더 사용자가 신뢰를 가질 수 있다는 점이다. 제안한 키 복구 기반의 IPSec의 전체적인 동작원리를 플로우차트를 써서 나타내면 그림1,2와 같다.

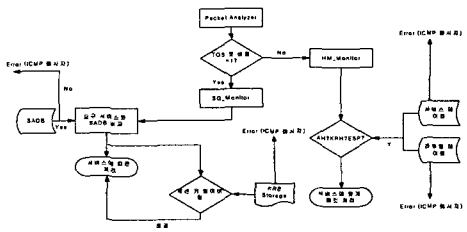

그림 1 TOS 설정에 따른 패킷 흐름도

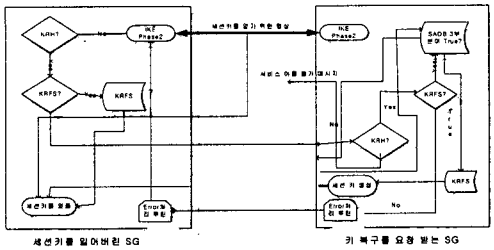

그림 2 키 복구 시의 메커니즘

3.1 NKRS_IPSEC을 위한 Security Gateway(SG) 구조

기존의 IPSec의 SG에서 필요한 부분을 살펴보면, 보낸 패킷의 TOS 설정 부분을 조사하기 위한 PA(Packet Analyzer) 모듈이 있다. 여기서 해야 할 일은 TOS 필드 중 SG를 가리키는 비트 설정을 보고, Host에서 보내온 패킷인지, 아니면 SG에서 보내온 패킷인지를 판단하여, 이 패킷을 HM(Host_Monitor)로 보낼 것인지, SM(SG_Monitor)로 보낼 것인지를 결정한다. 이 동작은 일반 패킷과 IPSec 처리된 패킷들을 구분하기 위함이고, IPSec 서비스 중에서 어떤 서비스를 이용할 것인지를 구분하기 위함이다. PA는 들어오는 패킷들을 분석하여 HM을 호출할 것인지, SM을 호출할 것인지가 결정된다. 이로 인해 두 번째로 필요한 모듈은 HM과 SM 부분이다. HM은 PA에 의해 이곳으로 경로가 설정 된 패킷들은 TOS의 비트 중 SG 부분이 0인 패킷들이다. 이 패킷을 분석하여 IPSec을 적용할 수 있는지를 판단한다. 라우팅 테이블과 서비스 테이블을 보고 이 서비스를 적용시킬 SG가 라우팅 테이블 안에 있는지를 검사하고, 서비스 테이블(표 1)을 보고 정당한 서비스를 요구하는지를 판단한다.

만약에 라우팅 테이블 참조 에러나 서비스 요구 에러가 나면 HM은 서비스를 요구한 Host에게 ICMP 메시지를 이용하여 에러 메시지를 보내고 그 패킷을 폐기한다.

SM 역시 PA에 의해 이곳으로 경로가 설정된 패킷들은 TOS의 비트 중 SG 부분이 1인 패킷들이다. 이 패킷들은 서비스가 처리된 패킷이기에 바로 인증이나 복호화 하는 것이 아니라 이 작동에 들어가기 앞서 서비스 처리 대상이 되는지를 판별한다. 그 패킷의 IPSec 적용에 대한 SA를 구별하는 방법은 목적지 IP, SPI로 판별을 하게 된다. 즉 패킷에 대한

Decimal	sender	SG	AH	KRH	ESP	비고
0	H O S T	0	0	0	0	일반패킷
1		0	0	0	1	ESP
2		0	0	1	0	Error
3		0	0	1	1	Error
4		0	1	0	0	AH
5		0	1	0	1	AH+ESP
6		0	1	1	0	Error
7		0	1	1	1	AH+KRH+ESP
8	S G	1	0	0	0	Error
9		1	0	0	1	ESP
10		1	0	1	0	Error
11		1	0	1	1	Error
12		1	1	0	0	AH
13		1	1	0	1	AH+ESP
14		1	1	1	0	Error
15		1	1	1	1	AH+KRH+ESP

표 1 Service Table(ST)

인증을 하기 위해 사용된 인증 알고리즘과 키, 암호화를 하기 위해 사용된 암호화 알고리즘과 키가 있다면 이것을 다시 인증 확인 또는 복호화 키를 알아야 하고, 또한 암호화 알고리즘과 키를 알기 위한 정보가 SADB(SA DataBase)에 들어 있다. 이러한 것들을 확인함으로써 서로간에 협상 된 규칙에 의한 IPSec인지를 판별한다. 비교 후에 맞는 것이 없다면, 불법적인 패킷으로 간주하고 에러 메시지를 보낸다.

세 번째로 키 복구를 위한 KRH 모듈 부분이 필요하다. 이 서비스를 포함한 패킷이 먼저 HM에 의해서 분류가 됐다면, 목적지 IP와 SPI를 보고, 그것에 대응하는 SA를 SADB에서 찾는다. SADB에서 찾은 SA들의 내용으로 KRF를 만들고, 다시 그것의 보안성을 위해 KRB를 만든다. 이것을 다시 IPSec 프로토콜에 맞게 보내기 위해 KRH 형식으로 최종적으로 만든다. 그러고 나서 인증을 위한 AH를 삽입하고, ESP에 대한 키 복구 정보를 가진 KRH를 ESP 앞에다가 삽입해서 보내게 된다.

다음으로 SM에 의해 이 서비스를 포함한 패킷이 분류되었다면, SM은 이 패킷에 대한 KRH의 인증과 무결성을 검증하고 정당한지를 살피고, 이것을 임시 기억 복구 저장소인 KRFS(KRF Store)에 세션이 끝날 때까지 저장한다. 세션이 이어져 있는 동안 세션 키를 잃어버렸을 경우 저장된 세션 키를 이용한다.

마지막으로, KRF에 대한 KRFS가 필요하다. 이 부분은 소스 SG가 패킷에다가 KRH를 포함하여 보낸 경우, KRH를 디캡슐화해서 KRB에 있는 KRF를

세션이 끝날 때까지 임시 저장해두는 곳으로서, 세션이 끝나게 되면 폐기된다. KRF에는 세션 키의 정보들이 저장되어 있다. 만약 패킷을 받다가 ESP로 암호화 한 것을 복호화 할 수 없는 경우, KRFS에 저장해 둔 KRF가 있는지 조사하고 이 정보에 따라 세션 키를 얻은 후 복호화 할 수 있다.

3.2 키 복구 기반이 있는 IPSec에서 세션 키를 얻을 때의 성능 예측

ISAKMP[6]는 6개의 메시지로 구성된 Phase1 과정과 3개의 메시지로 구성된 Phase2 과정으로 구성되어 있다. phase2과정을 통해서 세션 키를 얻은 후 데이터 전송을 한다. 이 과정에서 세션 키를 분실할 경우 암호화 된 패킷을 복호화 할 수 없기 때문에 phase2 재협상을 통해 세션 키를 얻은 다음에 데이터를 재전송한 후 복호화 한다. 이 과정의 지연시간은 $K_t=t_{p1}+2(t_{p2}+nt_f)$이다. 하지만 사용자가 KRH 서비스 요청을 할 경우 이 세션 키를 저장해 둔 후, 데이터를 전송하게 되는데 전송하다가 세션 키를 잃어버린 경우 그 패킷을 받은 SG는 저장된 세션 키를 이용하여 그 패킷을 복호화 할 수 있다. 이때의 지연시간은 $K_t'=t_{p1}+t_{p2}+(n+1)t_f+t_{proc}$이다. 따라서 $K_t-K_t'=t_{p2}+(n+1)t_{frame}-t_{proc}\simeq t_{p2}+(n-1)t_f$ 만큼 지연시간을 단축할 수 있다.($\because t_{proc}\ll 1$ 이라고 가정)

(K_t:키 복구 기반이 없는 경우 걸리는 시간, K_t':키 복구 기반이 있는 경우 걸리는 시간, t_{p1}:키 협상 중 phase1에 걸리는 시간, t_{p2}: 키 협상 중 phase2에 걸리는 시간, t_f:하나의 프레임을 보내는데 걸리는 시간, nt_f:n개의 프레임을 보내는데 걸리는 시간, t_{proc}: 하드웨어 내부 연산 시간)

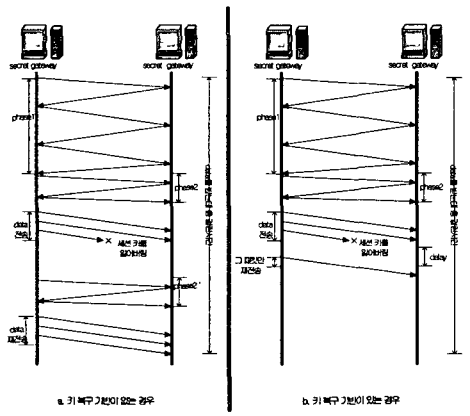

그림 3 성능 비교

4. 결론 및 향후 연구 과제

본 논문에서는 기존의 IPSec에서 정책서버를 복잡성을 줄이고 사용자의 신뢰 확보와 안정성을 지원할 수 있는 IPSec을 제안한다. 임의대로 불필요한 보안 서비스를 정책적으로 시행하는 것보다는 사용자 자율성을 고려하여 자신의 전송할 데이터의 보안 등급을 결정하고 그에 따른 자신의 서비스를 선택할 수 있다. 또한 암호화 된 데이터 양이 많을 경우 세션키 분실시 키 복구 메커니즘을 사용하는 경우 많은 지연 시간을 줄일 수 있다.

향후 연구 과제는 무선 환경 하에서 사용자의 신뢰성을 확보할 수 있는 캡슐화 방식의 키 복구 기반을 제공할 수 있는 IPSec을 구현하는 방법을 연구하고자 한다.

5. 참고 문헌

[1] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, NRL,

[2] Atkinson, R., "IP Authentication Header", RFC 2401 August 1998.

[3] Atkinson, R., "IP Encapsulating Security Payload", RFC 1827, NRL, August 1995.

[4] Tom Markham, Key Recovery Header for IPSec, Computer & Security 2000, PP. 86-90.

[5] Sabari Gupta, A Common Key Recovery Block Format:promoting Interoperability between dissimilar key recovery schemes

[6] Internet Security Association and Key ManagementProtocol(ISAKMP),Douglas Maughan, Mark Schertler, Mark Schneider, Jeff Tunner,INTERNET-DRAFT draft-ietf-ipsec-isakmp-08.txt,.ps,July26, 1997.