通信情報保護學會學會誌
第 9 券 第 4 號, 1999. 12

멀티캐스트 적용을 위한 인터넷 키관리 프로토콜 SKIP 분석

김대호*, 박응기*, 김영수*

요 약

안전한 멀티캐스트 통신을 위하여 인증과 프라이버시 기능을 네트워크 계층에 두는 것이 효율적인 경우가 있다. IPv4나 IPv6과 같은 네트워크 계층 프로토콜에서 공통으로 사용되는 대부분의 방식은 비세션 프로토콜(session-less datagram oriented protocol)이다. 인터넷 보안구조에서 정의하고 있는 키 관리 프로토콜인 SKIP은 IPv4와 IPv6에 적합하도록 정의된 IP 보안 프로토콜인 AH(authentication header)와 ESP(encapsulating security payload)에서 동작되도록 설계되어 있으며, IPv4나 IPv6과 같은 비세션 프로토콜에 연결되어 사용되도록 정의된 키 관리 기법이다. 본 논문에서는 SKIP에 대하여 분석하며, 멀티캐스트 환경에 적용하기 위한 고려사항을 검토하고 정리한다.

I. 서 론

인터넷의 사용이 폭발적으로 증가하고 상업용으로 사용이 급속히 확대되면서, 보안 문제로 관심이 집중되고 있으며, 인터넷을 잘못 사용하면 가시적인 손실을 입게 된다는 인식으로 보안 기능은 기본적인 기능으로 이해하게 되었다.

인터넷에서의 보안 요구가 강해짐에 따라 패킷 필터링이나 방화벽과 같은 보안 문제의 해결 방안이 대두되기 시작하였으나, 이러한 해결 방안은 대부분 독립적이며, 개념적으로 서로 다른 방안을 사용하므로 연동성이 부족하다. 독립적인 해결 방안 중 일부는 폐쇄된 기업 내에서만 응용 프로그램들과 동작되며, 또 일부는 운영체계에 연결되거나 링크계층에 삽입되어 동작되고 있다.[1]

본 논문에서는 인터넷의 보안 구조를 기반으로 제시된 키 관리 방안의 하나인 SKIP (simple key-management for internet protocols)기법을 소개하고 분석한다. SKIP은 유니 캐스트를 중심으로 하여 연구된 키 관리 기법이지만 멀티캐스트 서비스가 등장하면서 기능을 확장하는 방안이 검토되고 있다. 또한 멀티캐스트 서비스에서 중요한 문제점인 완벽한 전방보호 기능을 SKIP을 이용하여 처리하는 연구 내용을 소개한다. 또한, 본 논문의 주된 목표는 확장성 있는 방법으로 인터넷 서비스에서 인증과 암호가 어떻게 사용되는 가를 보인다.

II. 배 경

1. 인터넷 기술의 변화

1) 동적 라우팅 기능

라우팅 테이블 목록을 작성하여 라우팅이 수행되는 정적 라우팅 프로토콜은 네트워크 규모가 작거나 다른 네트워크와의 접속 점이 하나 혹은 경로가 이중화되어 있지 않은 경우에 적합하다. 그러나 이러한 조건 중에 어느 것이라도 적합하지 않은 경우 동적 라우팅 프로토콜을 사용하게 된다. 동적 라우팅이란 라우터가 인접한 라우터와 네트워크에 관한 정보를 교환할 때

* 한국전자통신연구원 정보보호기술연구본부(dhokim, ekpark, yskim@etri.re.kr

사용되는 프로토콜이며 네트워크 상태 변화에 따라 대처하는 기능이 뛰어난 특징을 갖고 있다. 가장 기본적인 동적 라우팅 프로토콜로는 RIP(routing information protocol)가 있으며 RIP의 문제점을 보완하여 OSPF(open shortest path first), BGP(border gateway protocol) 등이 개발되어 널리 사용되고 있다.[2]

널리 알려진 프로토콜인 OSPF는 링크-상태(link-state) 프로토콜이다. 기존에 사용되던 RIP는 거리-벡터(distance-vector) 프로토콜인데 반하여, OSPF는 인접한 라우터의 연결 상태를 점검하고 그 정보를 상호 교환하는 프로토콜이다. 링크-상태 프로토콜은 라우터의 다운이나 회선 절단 등의 변화로부터 회복되는 시간이 상대적으로 짧은 특성을 갖고 있다.

2) 인터넷 키관리 기술

인터넷에서 정보보호 서비스를 제공함에 있어서 키 관리가 중요한 문제로 등장한다. 정보를 안전하게 전송하기 위해서는 상대에게 자신을 확인시키고 데이터 암호를 위한 키를 서로 공유할 수 있어야 한다.

인터넷에서 사용되는 노드의 수만큼 확장할 수 있는 키 관리 기법은 근본적으로 인증된 공개키에 기반한 구조에 기초하여야 한다. 이들 공개키는 공개키 보증서, 디렉토리 서버 등 다양한 메커니즘을 사용하여 인증된다. 이 모든 메커니즘은 기본적으로 식별자와 그에 해당하는 공개키를 안전한 방법으로 하나로 묶는 기능을 제공한다. 기존 대부분의 응용프로그램에서 인증되어지는 키는 RSA와 같은 공개키 암호시스템에서 사용하는 키로 가정하고 있다.

IP(internet protocol) 계층에서 보안에서의 목표는 최종 시스템 사이 또는 방화벽을 통한 완벽한 네트워크 사이에서 비밀과 인증이 보장된 통신을 하는 것이다. 이런 네트워크에서는 재 라우팅, 부하의 분배 및 복구기능과 같은 IP의 기존 특성을 잘 유지하며 최소의 오버헤드만으로 네트워크 트래픽을 안전하게 전송할 수 있는 구조를 갖고 있어야 한다. 동시에, 융통성과 확장성도 가져서 사용자 또는 포트 당 키 제공, 스마트카드 사용, 안전한 멀티캐스트, IPv6로의 전이, 새로운 알고리즘의 채용이 가능한 구조를 갖도록 연구가 진행되고 있다.

현재 키 관리 방안이 IPSEC의 IETF워킹 그룹에서 연구되고 있다(photuris 키 관리, oakley 키 관리, SKIP 키 관리). 이들은 기능적으로 공통부분을 갖고 있지만, 키 관리에 관한 문제를 해결하기 위한 기본적인 방향은 매우 다르다.[1]

3) 차세대 인터넷 프로토콜(IPv6) 개발

IPv6에서는 라우팅 번지 영역을 확장하는 것 이외에도 다양한 선택사항을 두어 서비스를 제공하고 있으며 QoS(quality of service), 인증, 보안 기능 등이 기본으로 제공되고 멀티캐스트 서비스 기능도 지원한다.[1]

마이크로소프트에서도 2002년에는 IPv6을 지원하기로 결정한 바 있으며 미국의 주요 ISP, 벤더들은 올 초에 IPv6을 위한 포럼을 구성하여 이를 준비하고 있다. 일본에서 생산되는 가전제품도 주소 체계를 받아들여 제품에 구현하여 출시하는 등 세계 각국은 차세대 인터넷을 통한 통합을 준비하고 있어 이제 IPv6은 선택사항이기는 하지만 도입 시기만이 문제로 인식되고 있다. 앞으로 서비스는 당분간 IPv4와 IPv6이 조화를 이루며 사용될 것이 분명하며 키 관리 서비스도 이를 반영하여 발전하여 나갈 것이다.

4) 비 세션성 프로토콜

기존의 키 관리 서비스는 IP와 같은 데이터그램 프로토콜에 인증 기능과 프라이버시 기능을 제공하는 방법이 사용되었다. 세션 키를 설정하기 위하여 통신이 이루어지기 이전에 프로토콜을 사용하여 인증된 세션 키를 먼저 설정하는 대역 외(out-of-band) 통신방식이다. 세션 키는 IP 데이터 트래픽을 암호하거나 인증하기 위해 사용된다. 이러한 기법에는 비세션(sessionless) 프로토콜인 IP 계층을 지배하는 의사(pseudo) 세션 계층을 설치하고 관리해야 하는 단점이 있었다. 세션 키가 바뀔 필요가 있을 경우에도 발신측, 착

신측 IP는 이를 위해 다시 통신해야만 한다. 이런 통신과정에서는 키 교환을 위하여 계산이 복잡한 공개키 연산동작을 매번 포함하여야만 한다.

IP 계층에서 세션성(session oriented) 키 관리를 수행하는 데에는 또 다른 커다란 결점이 있다. IP 프로토콜의 중요한 특징이 중간 노드나 링크에서 고장이 발생해도, 그 주위에서 동적으로 재 라우팅 하여 중간 노드나 링크의 문제를 복구해 주는 기능이다. 원래 IP 계층을 설계한 기본 동기는 군사적 공격을 받아 네트워크의 일부가 파괴되는 경우에 스스로 복구하여 신뢰성을 갖추는 것이었다. 상용 환경에서도 네트워크 액세스와 서비스의 가용성을 높이기 위해 이러한 요구 조건들은 필요하다.

이러한 네트워크 문제를 복구하기 위해서는 통신 중 발생하는 모든 상태를 중간 노드가 계속적으로 유지하고 있을 필요가 없으므로 IP의 사용을 촉진하게 되며 각 IP 패킷은 독립적으로 라우팅 할 수 있다. 그러므로 문제가 발생된 라우터나 링크 주변에서 IP 트래픽을 간단하게 재 라우팅 하는 것이 허용된다. IP 라우터나 링크 상의 문제 때문에 재 라우팅 하는 것 이외에, 목적지까지 복수의 경로가 존재하면 각 경로에 균등한 트래픽을 분배하는 기능인 부하조절 기능(load balancing)을 수행하여 네트워크 트래픽의 효율을 높이도록 동적인 라우팅을 수행하기도 한다. 그러나 기존의 방식으로 IP에서 세션성(session-oriented) 키 관리를 수행하게 되면 IP의 이런 특성들을 해치게 된다. 이런 이유로 비 세션성(sessionless)이며 비 상태성(stateless) 방법에서 동작하는 키 관리 기법이 연구되고 있다.

2. 인터넷 프로토콜 보안구조 등장

IP 보안구조는 1995년 8월 표준화되었으며 1998년 8월 개정되어 RFC 2401로 발표된 바 있다. 많은 문서가 제정되고, 일반 구조와 AH(authentication header), 공통 인증 알고리즘(MD5) 및 ESP(encapsulating security payload), 암호알고리즘(DES - CBC모드)에 대해 권고 내용들을 다루고 있다. IP 보안구조에서는 트래픽 보호를 위해 AH, ESP를 사용하며 키 관리 부분은 연구가 계속 추진 중이다.[3][4]

* AH: 비 접속형 네트워크에서의 무결성, 데이터 발신자 인증을 기본으로 제공하며, 선택적으로 리플레이 방지 서비스도 제공한다.
* ESP: 트래픽에 대한 기밀성과 제한된 범위의 트래픽에 대한 기밀성도 제공한다. 또한 AH에서 제공되는 서비스도 지원한다.

이 프로토콜은 독자적으로 사용될 수 도 있으며, IPv4나 IPv6 환경에서 요구되는 서비스를 제공하기 위하여 결합하여 사용하기도 한다. 프로토콜의 사용에는 두 가지 모드가 있다. 하나는 전송(transport)모드이며 다른 하나는 터널(tunnel)모드이다. 전송모드에서 프로토콜은 상위 계층의 프로토콜에 보호기능을 제공하며, 터널모드에서의 프로토콜은 IP패킷을 터널링 시키는데 사용된다.

SA(security association)는 일련의 보안 정보로서 목적지 번지와 SPI(security parameter index)에 의해 유일하게 정의된다. SA에는 다음과 같은 파라미터를 포함한다.

* 인증 알고리즘, 암호 알고리즘
* 인증 키, 암호 키
* 암호용 초기벡터
* 키의 유효기간
* SA의 유효기간
* 발신자 번지

IPSEC SA는 수신자에 의해 확인된다. 멀티캐스트 그룹은 멀티캐스팅 될 때에는 수신자로 간주된다. 그러나 실제로는 여러 개의 수신자가 존재하므로 그 중에 하나의 수신자가 SA와 SPI들을 정의하며 그룹 소유자(owner) 또는 그룹 발신자(originator)가 될 수 있다.

유니 캐스트 환경에서 통신 호스트들은 정보보호 서비스 제공을 위해 보안 파라미터를 협상할 수 있다. 그러나 참여자가 많은 멀티캐스트 환경에서 보안 파라미터를 협상하는 것은 비효율적이다. 따라서 멀티

캐스트 환경에서 세션을 시작하는 사람은 안전한 세션을 위한 보안 파라미터를 정의하여 그룹 멤버들에게 분배하며, 안전한 세션을 위한 보안 파라미터는 SA 형식으로 저장된다. 안전한 한 그룹 내에 여러 개의 SA 즉, SPI가 존재할 수 있다. 세션이 시작되기 전에 그룹 멤버들은 SA를 알아야 하며, 세션을 시작하는 사람은 세션을 시작하기 이전에 SA를 안전하게 그룹 멤버들에게 분배하여야 한다.

III. 인터넷 키 관리 프로토콜 SKIP 분석

1. SKIP 동작 개요

SKIP은 DH 키 교환방식을 이용하여 공개 보증서를 상대에게 제공하는 방식에 근거한 키 관리 기법이다.[1] SKIP에서는 비 상태성 IP계층을 통하여 상태 정보를 재 전송하지 않으므로 비밀 값을 공유하는 방식에서 이러한 일들이 분명하게 일어나도록 설계되어야 하며, 각 참여자는 통신 상대에 대한 공개 값이 포함되어 있는 보증서를 액세스하여 데이터베이스로부터 그 값을 구해내게 된다. DH에서는 통신하고자 하는 쪽이 공유 비밀 값을 만들기 위해 비밀 값 i를 생성한 후 이 값을 이용하여 공개 값 $g^i \bmod p$를 계산하여 상대방에게 전송한다. 통신 상대에 대한 공개 값을 얻어내면 이 값은 DH 기법을 사용하여 시스템의 비밀 값과 결합되어 새로운 비밀 값을 계산하며, 동일한 방법으로 양쪽에서 새로 계산한 비밀 값을 공유하게 된다. 이 경우 공유 비밀 값은 다음과 같다. 여기서 상대방의 비밀 값은 j라 한다.

$$(g^i)^j \bmod p = g^{ij} \bmod p$$

이 공유 비밀 값은 장기간 사용되는 것으로서 데이터를 직접 암호 하는데 사용하지 않으며, 데이터를 암호하기 위하여 랜덤한 트래픽 키를 발생하여 사용하며, 이 암호 키는 장기간 사용되어야 하는 공유 비밀 값을 사용하여 암호 되며, 대역 내 통신 방식으로 알고리즘 선택정보와 대량 트래픽에 관한 keying material을 함께 전송한다.

SKIP에서는 수신자의 인증된 DH 공개 값 $g^{ij} \bmod p$를 쉽게 이용할 수 있어야 하며, 디렉토리 서비스나 수동 키 분배에 의한 전달이 가능하도록 구성한다. 이 공유 비밀 값은 수신자의 공개 값과 발신자의 비밀 값으로 계산되어지므로, 발신자 입장에서는 수신자와 관계없이 계산할 수 있다.

키를 암호 하는 키로서 공유 비밀 값을 사용한다. 암호나 인증에 사용되는 실제 키는 공유 비밀 값으로 암호 되거나, 공유 비밀 값과 현재 시각 정보(time stamp)를 이용한 해쉬 값으로 주어진다.

SKIP은 아직 완성된 것이 아니며 IETF IP security 워킹 그룹에서 photuris와 함께 개발 중이다. SKIP은 발신자가 SA를 선택하기 때문에 IP Security 모델에 정확하게 부합되지는 않는다. 몇 가지 문제가 있으나, 가장 심각한 취약점은 전방보호(forward secrecy)가 안 되는 것이다.[5] 만일 범죄자가 어떤 노드의 비밀키 정보 i를 알아낸다면 연결된 모든 노드의 SKIP 트래픽을 복호 할 수 있게 된다.

SUN에서는 SPARC Solaris 2.4에 SKIP을 구현하여 배포한 바 있으나, 미국의 수출허가제에 걸려 북미 지역으로 제한되어 있다. 그러나, SKIP을 구현하는 것은 GNU public license에 의해 제약을 받지 않는다.

SKIP을 구현하기 위해서는 IP 방식의 발신측과 착신측 모두는 인증된 DH 공개 값을 갖고 있어야 한다. 이 공개 값은 여러 방법으로 인증 받을 수 있다. DH 공개 값을 인증하기 위한 몇 가지 방법으로 X.509 보증서의 사용, 안전한 DNS(domain name server)의 사용 등을 들 수 있다. 이들 보증서는 RSA나 DSA(digital signature algorithm)와 같은 서명 알고리즘을 이용하여 서명될 수도 있다.

각 단말에서 대칭 키를 계산하기 위해서 다른 단말의 인증된 공개 값을 알기만 하면 된다. DH 공개 값은 별도로 보호할 필요가 없으며, 관련된 인증 공개 값을 발견하는 기본적인 방법은 디렉토리 서비스나

certificate discovery protocol을 이용하여 이 값을 배포하는 것이다.

공유 비밀 값은 IP 패킷 단위의 인증과 암호 기능을 제공하기 위한 키 암호용 키(key encrypting key)의 기초로 사용된다. 그러므로 $g^{ij} \bmod p$를 long- term secret라 부르며, 이 값으로부터 공유 비밀 키(K_{ij})를 얻는다. K_{ij}는 DES나 RC2, IDEA와 같은 블록 대칭 키 암호시스템용 키로서 사용된다. K_{ij}는 $g^{ij} \bmod p$로부터 하위 비트의 키를 취하여 사용한다. $g^{ij} \bmod p$는 최소 512비트가 되어야 하며, 보안성을 증대하기 위해서는 1,024비트 또는 그 이상이 되기도 하므로 블록 암호시스템용 키로 사용되는 K_{ij}는 항상 충분히 많은 비트를 얻을 수 있다. 일반적으로 사용하는 키 사이즈는 40에서 256비트에 해당된다. 여기서 K_{ij}는 간접적인 공유 비밀 키 쌍이 된다. 이 값은 별도의 패킷이나 대역외 방식으로 전송되고 결정될 필요는 없다. 착신측 IP 노드에서는 발신측 IP의 인증된 공개 값을 알기만 하면 간단히 공유 비밀 키(K_{ij})를 계산할 수 있다. 이 키는 간접적으로 마스터키로 사용되므로 별도의 오버헤드가 없이도 키 사이즈를 원하는 길이 만큼 얻을 수 있다. 강력한 암호시스템에 충분한 크기의 K_{ij}를 사용하면 키를 분석하는 것은 불가능해진다.

K_{ij}는 패킷 키(K_p)로 불리는 임시 키를 암호하기 위해 사용된다. K_p는 패킷을 암호 하거나 인증하는 키로서 사용된다. 이는 long-term 키 K_{ij}를 사용하여 암호하는 데이터의 실질적인 양을 제한하는 효과가 있으며, IP 데이터 트래픽은 K_{ij}를 사용하여 암호하지 않는다. 대신, long-term 키를 사용하여 K_p를 암호하고, IP 데이터 트래픽을 암호하고 인증하기 위하여 K_p를 사용한다. K_p가 상대적으로 적은 양의 데이터에 적용되므로, long-term 키를 오랜 기간에 걸쳐 유지한다 해도 long-term 키를 사용하여 암호되는 데이터의 양은 상대적으로 적은 양으로 제한되는 효과가 있다. K_{ij}가 트래픽 암호용이 아니라 다른 키를 암호하는 용도로만 사용되므로 마스터키로 불려진다. K_{ij}와 K_p는 대칭 키 알고리즘용 키로 사용한다.

발신측 노드(I)에서 암호용 키(K_p)를 변경하면 착신측 노드(J)는 공개키 계산을 거치지 않고도 이 사실을 알아낼 수 있다. 그러면 저장된 K_{ij}를 사용하여 암호된 새로운 패킷키 K_p를 복호 할 수 있다. 착.발신간의 별도의 통신이 필요 없으며, 공개키 연산 동작을 수행하지 않아도 패킷 암호 및 인증 키를 발신측에서 변경하고, 착신측에서는 계산 할 수 있다(실질적으로 K_p는 두개의 서로 다른 키, 즉 암호용과 인증용 키를 얻는데 사용된다).

2. SKIP 헤더 구성

1) 일반적인 SKIP 헤더 구성

SKIP 헤더는 대역 내 키, 알고리즘, 패킷에서 사용될 프로토콜 등을 나타내며 표 1과 같다. 첫 번째 필드는 버전(Ver)을 나타낸다. 버전 다음에 정의되는 4비트는 SKIP에 대한 미래에 사용될 버전을 예비(rsvd)해두는 필드로서 SKIP 버전 1에서는 zero로 설정하고 무시한다. Non-zero 발신측 NSID는 패킷이 발신측 MKID값을 갖고 있다는 것을 나타낸다. Non-zero 발신측 NSID의 값은 패킷이 발신측

표 2. SKIP 헤더구성

0			15	16			31
Ver	Rsvd	Source NSID		Dest NSID		Next Header	
Counter n							
K_{ij} Alg		Crypt Alg		MAC Alg		Comp Alg	
K_p encrypted in K_{ijn} (typiclly 8 to 16 bytes)							
Source Master Key-ID (If Source NSID is non-zero)							
Destination Master Key-ID (If Destination NSID is non-zero)							

표 3. ESP 적용시 구조

IPv4 Header	SKIP Hdr	Auth Hdr	Inner Protocol(e.g.,IP,TCP,

표 4. AH 적용시 SKIP의 상세 구조

0			15	16			31
Ver	Rsvd	Source NSID		Dest NSID		Next Header=AH	
Counter n							
K_{ij} Alg		Crypt Alg		MAC Alg		Comp Alg	
K_p encrypted in K_{ijn} (typically 8 to 16 bytes)							
Next Header		Length		Reserved			
SKIP SPI							
Authentication Data computed using A K_p(Variable Length)							

MKID를 갖고 있다는 뜻이다. 발신측 NSID의 값은 MKID를 갖고 있는 네임 스페이스를 나타낸다. non-zero 착신측 NSID는 SKIP헤더가 착신측 MKID 값을 갖고 있다는 것을 나타낸다. 착신측 NSID의 값은 MKID를 갖고 있는 네임 스페이스를 나타낸다.

NSID 바이트 뒤에 있는 Next Header 필드를 이용하여 다음에 어떤 프로토콜을 사용할 것인가를 나타낸다. 이 필드는 항상 ESP 또는 AH를 나타낸다. 그러나, 이 필드에는 keying material을 필요로 하는 어떤 프로토콜을 표시할 수도 있다.

Counter n 필드는 32비트 크기로서, coarse-grain playback 방지 혹은 키 재사용 방지의 목적으로 이용된다. 1바이트 필드 K_{ij} 알고리즘은 두개의 서로 다른 알고리즘인 키 암호용 알고리즘과 해쉬 알고리즘을 확인하는 용도이다. 키 암호 알고리즘의 키로서는 DH 공유 비밀 값의 하위 비트를 사용한다. K_{ij} Alg은 암호 알고리즘과 해쉬 알고리즘을 특정하여야 한다. 키 암호 알고리즘은 K_p를 암호하기 위하여 CBC 모드가 사용되어야 한다. CBC 모드 암호를 위한 초기치는 항상 zero가 되어야 한다.

키 암호 알고리즘의 입력은 키 암호 알고리즘의 블록 사이즈의 정수배로 랜덤하게 채워져서 주어진다. K_p의 길이는 암호 알고리즘이나 MAC 알고리즘의 정보에 따라 주어진다. 복호된 출력의 키 길이 만큼의 하위 비트가 K_p로 사용된다.

해쉬 알고리즘은 일반적으로 일방향 의사 랜덤 함수로서 마스터키의 n번째 버전을 발생하기 위해 사용하며 K_p를 분리하여 암호와 인증키로 구분한다. 키 분리 알고리즘은 일반적으로 마스터키를 생성하기 위해 사용되는 암호학적 해쉬 함수로서, K_p를 분리하여 암호키와 인증키를 만든다.

Cryp Alg필드와 MAC Alg필드는 암호와 인증을 위한 내부 프로토콜에서 사용되는 알고리즘을 특정한다. 이들 알고리즘은 사용하는 프로토콜에 부합되며 정의하고 있는 변환에 적합하다. 일반적으로, MAC Alg는 MAC 코드를 연산하는 방법을 자세히 기술하며, Crypt Alg은 패킷을 암호하기 위해 사용하는 알고리즘을 표시한다. 그러나 이것은 절대적인 것이 아니며 Crypt Alg에 암호와 MAC 계산을 동시에 수용할 수 있는 방안을 정의할 수도 있다.

Comp Alg 필드는 암호나 인증을 하기 전에 패킷을 압축하기 위해 사용하는 알고리즘을 나타낸다. Non-zero Comp Alg 필드는 암호와 인증 이전에 평문에 압축이 행해졌음을 나타낸다. Comp Alg 필드의 값은 사용된 알고리즘을 나타낸다. "K_p Encrypted in K_{ijn}" 필드는 암호된 K_p가 K_{ij} 알고리즘을 이용하여 키 K_{ijn}을 갖고

표 4 ESP 적용시 구조

IPv4 Header	SKIP Hdr	Auth Hdr	Inner Protocol(e.g.,IP,TCP,

표 5. ESP 적용시 SKIP 상세 구조

0			15	16			31
Clear IP header protocol = SKIP (typically 20 bytes)							
Ver	Rsvd	Source NSID		Dest NSID		Next Header=ESP	
Counter n							
K_{ij} Alg		Crypt Alg		Reserved		Comp Alg	
K_p encrypted in K_{ijn} (typically 8 to 16 bytes)							
SKIP_SPI							
Opaque transform data (variable length)							

표 6. AH, ESP를 포함한 SKIP 구조

0			15	16			31
Clear IP header protocol = SKIP (typically 20 bytes)							
Ver	Rsvd	Source NSID		Dest NSID		Next Header=AH	
Counter n							
K_{ij} Alg		Crypt Alg		MAC Alg		Comp Alg	
K_p encrypted in K_{ijn} (typically 8 to 16 bytes)							
Source Master Key-ID							
Destination Master Key-ID							
Next=ESP	Length	Reserved					
SKIP_SPI							
Variable length AH MAC, computed using A_K_p							
SKIP_SPI							
ESP transform data (e.g., IV), payload encrypted using E K_p							

암호 되었음을 표시한다. 발신측 MKID 필드는 발신측의 MKID를 포함하고 있다. 이 필드는 발신측 NSID가 non-zero일 경우에만 나타난다. 목적지 MKID 필드는 원하는 SKIP 수신자가 MKID 값을 갖고 있음을 나타낸다. 이 필드도 착신측 NSID가 non-zero일 경우에만 나타난다.

2) AH 프로토콜 적용(IPv4의 경우)

AH 프로토콜은 IP 데이터그램에 대한 인증을 제공하기 위해 사용된다. SKIP 헤더는 AH 헤더보다 앞에 있으며 표 2와 같이 IP 헤더의 뒤에 있다. AH와 결합된 SKIP의 자세한 프로토콜은 표 3과 같다.

AH 헤더에 있는 SKIP_SPI는 알고리즘 정보와 keying material이 다음에 오는 SKIP헤더에 포함된다는 것을 표시한다. 인증 데이터와 계산된 MAC의 위치는 특정한 변환으로 정의된다. RFC 1828을 참조하면 인증을 위한 변환의 예를 볼 수 있다.

3) ESP 프로토콜 적용(IPv4의 경우)

암호를 위해 SKIP을 사용하는 예는 ESP 프로토콜과 SKIP을 결합한 경우이다. ESP 프로토콜은 IP 데이터그램 전체를 암호하거나 IP 데이터그램의 Payload만을 암호하며, 그 각각은 ESP가 터널모드로 동작하느냐 전송모드로 동작하느냐에 달려 있다. SKIP 헤더는 IP 헤더 다음에 오며 ESP 헤더의 앞에 온다.

ESP와 결합된 SKIP의 자세한 프로토콜은 표 5와 같다. ESP 헤더에 있는 SPI 즉 SKIP_SPI는 알고리즘 정보와 keying

material이 다음에 오는 SKIP 헤더에 포함된다는 것을 표시한다. 이 예약된 SPI는 상징적인 값 SKIP_SPI로 할당되어 있다고 가정한다. 소스와 목적지 NSID는 zero로 가정하여 MKID가 존재하지 않는 것으로 한다.

불투명한 변환 데이터는 DES-CBC와 같은 변환으로 정의된다. 이 데이터에는 일반적으로 암호 데이터와 IV와 같은 변환을 위한 특별한 데이터가 포함된다. K_{ij} Alg 필드는 K_p를 암호하기 위해 사용되는 암호 알고리즘을 나타낸다. K_p는 Payload를 암호하기 위해 사용되는 E_K_p값을 얻는데 이용된다.

4) AH, ESP 프로토콜 적용(IPv4의 경우)

SKIP은 AH와 ESP 모드가 동시에 결합되어 사용될 수 있다. SKIP 헤더에 있는 Next Protocol 필드는 AH이며 AH 헤더에 있는 Next Protocol 필드는 ESP이다..

A_K_p는 인증용으로 사용되며, E_K_p는 암호용으로 사용된다. 표 6은 AH, ESP를 포함한 SKIP을 예로 설명하고 있다. 또한, MKID의 사용도 제시하고 있다.

3. 키 분배 및 추출

1) 키 분배

인증된 공개키 분배 기반구조가 구축되기 이전 임시 대책으로 노드에서는 키 정보들을 수동으로 전달하면 된다. 이 경우에 마스터키 K_{ij}는 SKIP을 사용할 때 수동으로 설치되는 키 중의 하나가 되어야 한다.

수동방식의 키 재설정 과정이 느리고 불편하므로, 아직 두 레벨 keying 구조, 즉 수동 설치 마스터키 K_{ij}를 사용하여 패킷 암호 키 K_p를 암호하는 방식을 사용하고 있다. 이 방법은 마스터키를 상대적으로 오랜 기간동안 유지하는 장점이 있으며 마스터키의 노출을 피할 수 있게 된다. 이 방법은 특히 정보의 수명은 짧으나 대용량 데이터를 쉽게 암호하기 원하는 분야인 고속 네트워크 링크에 적합하다.

마스터키와 트래픽 암호 키를 분리하기 때문에 SKIP기법은 마스터키를 수동으로 전송하는 경우에라도 트래픽 암호 키를 자동으로 변경하는 것이 가능하다.

2) 키 추출

일반적으로 패킷은 암호와 인증 두 가지 기능을 수행할 수 있다. 암호와 인증을 수행할 때 중요한 사항은 키를 분리하는 것이다. SKIP 구현을 정상적으로 하려면 SKIP을 통해 암호와 인증 키를 발생해야 한다.

패킷 헤더로부터 복호하여 얻은 K_p는 패킷을 직접 암호 및 복호하거나 인증하기 위해 이용하지 않으며, E_K_p와 A_K_p 키를 생성하기 위해 이용된다. E_K_p와 A_K_p는 다음의 방법으로 패킷에서 복호된 K_p로부터 생성된다.

$$E\_K_p = h(K_p|\text{Crypt Alg}|02h)|\ h(K_p|\text{Crypt Alg}|00h)$$

$$A\_K_p = h(K_p|\text{MAC Alg}|03h)|\ h(K_p|\text{MAC Alg}|01h)$$

여기서 h( )는 일방향 해쉬 함수이다. 해쉬 알고리즘으로 특정되어진 이 함수는 K_{ij} 알고리즘의 일부분이다. 암호 알고리즘과 MAC 알고리즘은 SKIP 헤더에 1바이트 필드로 주어진다.

이러한 방법으로 구성하는 데에는 E_K_p나 A_K_p 중 하나를 알고 있어도 h( ) 함수의 특성에 따라 다른 키에 대한 정보를 유추하지는 못한다는 특성을 갖고 있기 때문이다. 이것은 weak 암호 키가 strong 인증 키와 함께 사용될 수 있다는 것을 의미한다. E_K_p가 노출되었다 하여도 A_K_p는 전혀 노출되지 않으며 반대 경우도 성립한다.

실제로 사용되는 키 비트의 수는 알고리즘에 따라 다르다. 알고리즘에서 256비트 이하를 필요로 하면 일방향 함수의 출력에 대한 하위 비트의 키 사이즈만큼이 실제의 키로서 사용된다. 해쉬 함수로 MD5 알고리즘이 사용된 경우, 암호 키가 128비트 이하가 필요하다면, MD5 (K_p|Crypt Alg|00h)만을 이용하여도 E_K_p의 낮은 차수 128비트를 제공할 수 있으므로 이 함수만으로 계산하면 된다. 마찬가지로 128비트 또는 그 이하의 인증키 A_K_p를 필요로 하면 MD5(K_p|MAC Alg|01h) 함수

만을 이용하여 계산하면 된다.

MD5를 사용할 때 위에서 정의한 함수는 전체 256비트를 제공하여 SKIP과 함께 사용되는 암호와 인증에 필요한 충분한 크기의 키를 제공한다.

시스템을 구현할 때는 SKIP 헤더에서 복호하여 얻는 실제 K_p의 길이를 정하기 위하여 Crypt Alg 필드와 MAC Alg 필드에서 정의한 최대 키 사이즈를 사용한다. 패킷 헤더에 있는 암호된 K_p의 길이는 K_p의 길이와 암호 알고리즘으로부터 얻어지며, 이 알고리즘은 K_{ij} Alg에 표시되어 있다. 예를 들어, K_p의 길이가 120비트가 되고 키 암호 알고리즘이 64비트 블록 암호이면, SKIP 헤더에 있는 암호된 K_p의 길이는 128비트가 되는데 상위에 8비트의 랜덤한 값을 삽입한다. 삽입하는 랜덤한 값은 E_K_p와 A_K_p함수를 계산하기 위해서는 zero로 인식되어야 한다. 주어진 예에서 E_K_p 와 A_K_p 함수에 대한 K_p 입력은 128비트로서 상위 8비트는 zero로 채워져 있다.

4. SKIP을 사용한 보안 서비스

1) 인증 서비스

SKIP의 구현에서 어떻게 SKIP을 사용하여 키를 찾아내고 패킷 인증을 수행하는지를 알아본다. 프라이버시가 없는 인증만을 이루기 위하여, SKIP을 구현한 제품에서는 키 A_K_p를 이용하여 MAC을 계산하여 패킷에 첨부하여 헤더 필드를 만든다.

SKIP 헤더에 있는 MAC Alg 필드는 특수한 인증 변환을 찾기 위해 사용되어야 한다. 키 A_K_p는 예를 들어 MD5를 사용하여 MAC을 계산하기 위한 키로 사용된다. MAC은 캡슐화된 프로토콜이 어떻게 정의되던지 캡슐화된 프로토콜내에 삽입되어야 한다. K_{ij} Alg 필드는 K_p를 암호하기 위해 사용하는 암호 알고리즘을 나타낸다.

예약되어 있지 않은 모든 SKIP 헤더 정보는 데이터를 인증 할 목적으로 IP 인증헤더 계산 시 포함되어 있어야 한다. SKIP 헤더의 예약된 필드는 IP 인증헤더의 인증 데이터 계산에 사용될 목적으로 zero로 채워져야 한다.

2) 암호 서비스

SKIP이 암호만을 위하여 키 관련 장치로 사용된다면 Crypt Alg필드는 패킷 암호 알고리즘을 나타낸다. E_K_p는 Crypt Alg의 키로서 사용된다. Crypt Alg은 표준 변환으로 수행된다. 이 변환에는 IV나 message indicator(MI)와 같은 정보를 포함한다.

3) 암호 알고리즘 사용의 독립성

비록 위에서 설명한 사항들이 고전적인 DH방식을 암호학적으로 구성하며 표현했지만, 이 프로토콜은 어떤 공개키 협정 시스템에서도 일반화하여 적용할 수 있다. 여기서 공개키 협정 시스템이란 다른 쪽의 공개 값과 자신의 비밀 값을 결합하여 상호 공유 비밀 값을 계산해 내는 과정이다.

암호학적 구성의 예로서, 공개키 협정시스템과 동일한 특성을 제공할 수 있는 소수 필드에서 지수함수를 이용하는 방법 이외에도 유한필드의 타원곡선 방식에서 지수함수 방법으로 구성할 수도 있다.

위에서 거론한 키 관리 프로토콜은, 공개키 협정방식에서 공개 값과 비밀 값을 사용하는 방식으로 일반화될 수 있다.

그러므로 사용할 공개키 협정 알고리즘은 공개키 보증서와 동등한 메커니즘(예를 들어, 안전한 DNS)을 사용하여 서로 알고리즘 식별자를 교환하면 된다.

5. 안전성 분석

1) 마스터키 변경 알고리즘

마스터키는 n으로 표시하는 카운터의 함수로 만들어 임의의 수를 생성하기 위해 이용될 수 있다. 카운터 값 n은 데이터 트래픽의 재생을 방지할 뿐만 아니라, 트래픽 인증키의 재사용을 방지할 수 있다. 특정한 트래픽 인증 키가 어떤 이유로든 노출되어 재 사용하여도 간접적인 마스터 키 K_{ij}를 변경하므로 동일한 마스터키를 재 사용하지 않아 인증 키의 재사용도 막을 수 있게 된다.

카운터 이용 시에 카운터가 변경되면, 패킷은 카운터 값의 도움을 받아 암호되며, 수신된 카운터 값이 로컬 n과 1보다 더 큰 차이를 갖게 되면 그 패킷은 버린다. n번

째 마스터키는 다음 함수로 계산된다.

$K_{ij}' = $ Low order 256 bits of K_{ij}

$K_{ijn} = h(K_{ij}'|n|01h)|h(K_{ij}'|n|00h)$

K_{ij}를 설치하기 위하여 공개키 협정방식(public key agreement)이나 수동키 협정방식(manual key agreement)을 사용할 때 K_{ij}는 256비트가 되어야 한다. K_{ij}가 $g^{ij} \bmod p$로부터 추출되어 하위 비트부터 256비트가 선택되고 K_{ijn} 연산을 위한 입력으로 사용된다. K_{ij}을 수동으로 설치할 때도 그 길이는 256비트이어야 한다.

MD5를 사용하여 동작과정을 표기하면 다음과 같다.

MD5(K_{ij}[MSB first]|n[32 bits]|00) --> Low 128 bits of K_{ijn}

MD5(K_{ij}[MSB first]|n[32 bits]|01) --> High 128 bits of K_{ijn}

2) 키 생성 및 변경

○ 키 생성

소프트웨어로 SKIP을 구현할 때 가장 고려하여야 할 사항은 랜덤한 난수를 발생하는 것이다. 약점이 있는 예측 가능한 난수의 소스는, SKIP의 보안성에 치명적인 요소가 된다. 특히, 이러한 예측 불가능한 난수 발생을 위해 피해야 할 사항은 현재의 프로세스 ID나 호스트 ID, ethernet 번지, 현재 날짜 또는 이들을 적당히 결합하여 암호 키를 생성하기 위한 입력으로 사용하는 것이다. 이 값들은 예측 불가능한 것은 아니다. 적어도 키 생성단계에서 가능한 한 많은 랜덤 비트가 포함되어야 한다. 이처럼 공통으로 사용되는 소스는 공동으로 사용되는 암호 알고리즘에 충분히 랜덤한 값을 제공하지 못한다.

○ 키 변경

암호와 인증에 대한 분석을 어렵게 하는 좋은 방법은 패킷을 암호하고 인증에 사용하는 키를 주기적으로 변경하는 것이다. 그러나 키가 변경되는 빈도는 보안정책에 따라 결정되어야 하며 몇 가지 파라미터가 권고되어진다.

트래픽 암호와 인증키는 10M 바이트마다 한번씩 또는 2분마다 한번씩 변경되는 것을 권고한다. 트래픽 암호와 인증키는 K_{ijn}이 변경될 때마다 변경되어야 한다. 또한, 주어진 노드에 다수의 서로 다른 K_{ijn}이 존재하면 서로 다른 K_{ijn}사이에서는 K_p를 공유시키지 말아야 한다. K_p는 각 수신자마다 또는 같은 노드에 있는 책임자 별로 랜덤하게 발생되어야 한다.

3) 공격에 대한 안전성 검토

SKIP에서는 일반적인 암호 공격법을 어떻게 피하고 있는가를 살펴본다.

○ intruder in-the middle attacks

인증되지 않은 DH는 intruder in-the-middle 공격을 받을 여지가 있다. 이를 막으려면 인증된 DH 기법을 사용하여 대상자의 비밀 서명키로 서명동작을 하여야 한다.

SKIP은 intruder in-the-middle 공격을 받을 여지가 없다. 이는 DH 공개 파라미터들이 long-term이며 인증되기 때문이다. DH에 대한 intruder in-the-middle 공격을 하는 침입자는 대상자가 공개 DH 키를 결정할 수 없을 것이라고 가정한다. 인증된 DH 공개 값은 두 대상자 사이에서 메시지의 변경을 요구하지 않으며 또는 자리 수가 큰 지수연산의 오버헤드를 초래하지 않으므로 이러한 가능성을 줄여준다.

○ known key attacks

패킷 인증을 위해 사용하는 대역 내 트래픽키 K_p가 언젠가 노출된다면, 마스터 변경 알고리즘에 의해 위조된 트래픽을 보내기 위해 노출된 키의 재사용을 방지할 수 있다. 이는 특정한 트래픽키 K_p가 노출되었다 하여도, 이를 통하여 현재 사용되는 간접적인 K_{ijn}에 대한 어떤 정보도 공격자에게 제공하지 않으므로 공격자는 K_{ijn}를 이용하여 암호된 K_p를 복호할 수 없게 된다. K_{ijn}를 이용하여 K_p에 대한 정보를 구할 수 없다면 공격자는 과거에 노출된 키 K_p를 어떤 목적으로도 재 사용할 수 없게 된다.

또한, 주어진 K_{ijn}로 암호된 패킷 암호용 키와 인증키(K_p)의 모든 값이 노출되었다 하여도 이를 통하여 K_{ijn}를 알 수 없으므로, 공격자는 앞으로 사용될 다른 K_p에 대한 정보를 얻는데 도움이 되지 않는다. K_p

를 알거나 선택하여 K_{ijn}를 알아내기 위해 사용하는 것은 K_{ijn}에 대한 known plain-text attack 또는 chosen plain-text attack과 같아서 비록 아무리 많은 K_p를 알거나 선택하더라도 키 암호 알고리즘이 노출되지 않는 한 불가능하다. 키 암호알고리즘이 안전하면 SKIP은 known key attack 또는 chosen key attack에 대하여 안전하다.

◌ clogging defense

공격자는 SKIP을 구현한 노드에, 예를 들어 DH 연산을 위하여 받아들일 수 없을 정도로 큰 수를 수행하도록 강요하여 서비스 거부현상이 일어나도록 할 수 있다.

서비스 거부에 의한 공격을 막기 위해서 사용 패턴을 조사하여 대처하든지, 마스터키를 미리 계산하고 결과를 저장하여 두는 방법이 권고되고 있다. 동작을 중지시키는 공격이 사용되는 경우에도 IP 노드는 아직은 다른 장비들과 통신을 할 수 있어 마스터키가 미리 계산된 경우가 있는지 확인하고 새로운 마스터키를 계산하는 동작을 중지한다. 그러므로 마스터키가 일단 계산된 적이 있으면 IP 노드에서 keying이나 키 재설정을 위해 연산에 대한 과도한 부하가 없게 되어 서비스 거부 공격에도 정상동작을 유지할 수 있다.

시스템 관리에 속한 키들은 항상 이러한 공격에 대비하여 미리 계산하여 저장하는 장소에 보관해 두어야 한다. 그러므로 clogging attack의 경우에도 관리자는 더 중요한 사항들을 이 저장 영역에 안전하게 저장하는 것이 가능하게 된다.

4) 전방보호

long-term keying material의 노출로 인하여 과거에 암호 되었던 트래픽의 노출위험은 특별한 마스터키의 사용 기간을 최소화시켜 줄일 수 있다. 앞으로 기본적 SKIP 프로토콜을 확장할 때는 단명(短命)의 인증된 DH 공개 값을 갖거나, 또는 단명의 DH 요소를 사용하여 마스터키를 계산하는 방법을 이용하여 전방보호 기능을 처리할 수 있을 것이다.

IV. 완벽한 전방보호를 위한 SKIP의 확장

1. 완벽한 전방보호 기능

현재 기술적인 측면에서 멀티캐스트의 가장 큰 문제점은 그룹 키를 효율적으로 관리하는 문제이다.[5] 멀티캐스트의 통신에서 모든 통신내용이 그룹 키에 의하여 암호화된다는 가정하에서 그룹키 관리의 가장 어려운 부분은 멤버들이 매우 빈번하게 가입 및 탈퇴를 할 수 있다는 것이다. 이러한 멤버들의 가입 및 탈퇴에서 요구되는 정보보호는 새로운 멤버가 가입할 때 이러한 새로운 멤버는 가입하기 이전에 발생한 통신 내용을 알아서는 안되며, 기존의 멤버가 탈퇴할 때 그룹을 떠난 이후에 그룹 내에서 발생되는 통신내용을 알아서는 안 된다. 이러한 2가지 요구사항은 통합하여 전방보호라 불리고 있으나 엄밀히 구분하여 전자를 후방보호(backward secrecy), 후자를 전방보호(forward secrecy)라고 한다.

이 장에서는 수명이 매우 짧은 DH 키교환 방식을 SKIP 키분배 프로토콜과 함께 사용하여 완벽한 전방보호(perfect forward secrecy : PFS)를 필요로 하는 분야에서 PFS 기능을 어떻게 제공할 수 있는지를 보여준다.

단명의 보증서는 단명의 마스터키를 계산하기 위해 이용된다. 이 값은 SKIP 프로토콜에서 사용되었던 K_{ijn}에 대치되어 사용된다. 새로운 타입의 마스터키 ID(MKID)가 여기서 정의되어 단명의 마스터키 사용을 표시한다. 단명의 보증서를 교환하는 환경에서는 PFS 이외에도 익명성 서비스의 지원이 가능하다.

세션성 모드와 비세션성 모드의 동작을 둘 다 선택적으로 사용하도록 하는 것은 한 쪽 모드의 동작을 지원하거나 아니면 다른 모드의 동작을 제공하는 프로토콜보다는 SKIP 프로토콜 환경에서 더 큰 융통성을 제공하도록 한다.

여기서 n은 SKIP 헤더에 있는 카운터

표 8. 단명의 보증서 구조

0			15	16			31
Ver	Rsvd	Protocol		Port		Cert MAC alg	
Validity Interval(seconds)							
DH Public Value Length				DH Public Value(g^x or g^y)			
~ ~							
Generator Length				Generator(g)			
~ ~							
Modulus Length				Modulus(p)			
~ ~							
Ephemeral Master Key-ID EMKID_I_J							
Ephemeral Master Key-ID EMKID_J_I							
Cert Enc. Alg		Cert Type		Encrypted Cert. Length			
Encrypted Long Lived Certificate							
~ ~							
Ephemeral Certificate MAC							
~ ~							

값이다. 이 마스터 키 연산은 기본적인 SKIP 프로토콜에서 정의한 마스터키 연산과 매우 유사하며 단지 마스터키 해쉬 값 계산에서 단명의 DH 공유 비밀 값으로 g^{xy}를 포함하는 것만 다르다.

EMKID_I_J와 EMKID_J_I의 값은 I에서 J로 또는 J에서 I로 전송하는 SKIP 패킷에서 사용되는 단명의 MKID가 된다. I에서는 J에서 I로 전송하는 패킷에 포함되어 있는 단명의 MKID값을 읽어내며 J에서는 I에서 J로 전송한 패킷에 포함되어 있는 단명의 MKID값을 읽어낸다. 어느 경우에 있어서도 양쪽의 단명의 MKID는 동일한 EK_{ijn}을 계산한다. EK_{ijn}은 SKIP 헤더의 필드에 있는 패킷 키 K_p를 암호하기 위해 사용된다.

g^{xj}를 사용하는 책임자 보증서의 암호는 선택사항이며, 이것이 단명 정보 교환에 포함되어 있는 상대에 대한 익명성을 제공하게 된다. 익명성이 요구되지 않거나 필요없을 경우에는 g^{xj}를 사용한 암호는 제외시킨다.

2. 단명의 보증서 포멧

단명의 보증서(ephemeral certificate)에는 본질적으로 랜덤하게 발생된 DH공개 값을 단명의 정보로 포함하고 있으며, 기본적인 SKIP 프로토콜에서 사용되는 유효기간이 긴 DH 값을 사용하여 보증하게 된다. 이 보증서는 기본 SKIP 프로토콜에서 MAC 값을 계산하는 키로서 K_{ij}를 사용하여 보증한다.

단명의 보증서 교환 정보를 포함하고 있는 사용자들은 위에서 살펴본 바와 같이 유효기간이 긴 DH 공유 비밀 값을 가지고 단명의 마스터키를 계산해 낸다. 그리고 나서 이 단명의 마스터키는 SKIP 헤더를 통해 전달되는 트래픽 키 K_p를 암호하기 위해 사용된다.

단명의 DH 공개 값에 추가하여 단명의 보증서에는 교환 시발자에 대한 식별자와 보증된 DH 공개 값을 포함하고 있다. 이 식별자는 익명성을 지원하기 위하여 암호될 수 있다. 표 7은 단명의 보증서 포맷이다.

버전(Ver) 필드는 보증서 포맷의 버전을 나타내는 값 1이 되어야 한다.

프로토콜과 포트번호 1은 함께 보증서 교환 값이 의도하는 응답자를 나타낸다. 응답자의 한 예는 telnet이나 ftp가 되며, 어느 경우에서도 프로토콜 필드는 TCP가 되어야 하며 포트번호는 telnet이나 ftp 데몬을 모니터하는 포트가 된다. 만일 프로토콜 번호가 zero라면, 노드는 단명의 보

증서가 교환되는 사용자가 됨을 나타낸다. 보증서가 J에서 I로 전송되는 동안에 이 필드는 zero로 채워져 있어야 한다.

"Cert MAC Alg"은 보증서 내용을 갖고 MAC을 계산하는데 이용되는 알고리즘을 나타낸다. MAC연산의 범주는 전체 보증서가 되며 MAC 계산을 위하여 zero로 채워진 MAC필드를 포함한다.

"validity interval"은 이 교환된 값을 통해 추출한 단명의 마스터키를 얼마의 길이로 사용할 것인가를 정해 준다. 이 값은 초 단위이다. 응답자는 발신자와 다른 값으로 이 필드를 선택할 수도 있으며, 그 경우 이 마스터키에 대한 유효 간격은 서로 교환되는 두개의 값 중에 적은 쪽이 된다. 유효기간이 끝나면 단명의 마스터키와 관련된 공유 비밀 정보들은 발신자와 응답자 양쪽에서 파괴된다. 새로운 정보 교환이 일어나며, 단명의 마스터키가 파기되기 전이나 후에 PFS 모드에서 보내질 필요가 있는 암호된 트래픽이 아직도 존재한다.

"DH public value length"는 DH 공개 값 필드의 길이를 정한다. "DH public value"에는 단명의 DH 공개 값을 포함하고 있으며, 상위의 8진수 값을 먼저 표시하는 방법으로 8진수로 표시한다. 마찬가지로 "generator length", "generator", "modulus length"와 "modulus"는 DH 연산에서 사용되는 발생기(g)와 모듈러(p)의 길이와 값을 나타낸다. 응답자는 발신자와 동일한 발생기와 모듈러의 값을 사용해야 한다.

EMKID_I_J는 I에서 J로 전송하는 단명의 MKID 패킷을 말한다. J에서 이 필드의 값을 추출하여 사용하므로 I에서 J로 전송되는 단명의 보증서는 zero로 채워져야 한다. 이 필드의 값은 I에서 J로 전송되는 단명의 보증서에 나타나 있다.

EMKID_J_I는 J에서 I로 전송하는 단명의 MKID 패킷을 말한다. I에서 이 필드의 값을 추출하여 사용하므로 J는 앞에서 설명한 I에서 전송한 패킷을 J가 수신하는 단명의 보증서의 경우와 마찬가지로 이 필드를 동일한 값으로 채워야 한다. EMKID_I_J와 EMKID_J_I는 SKIP 헤더에 있는 목적지 MKID처럼 사용되어야 한다. 단명의 마스터키 모드에서 사용될 때에는 소스 MKID는 비워 있어야 하며 SKIP 헤더에 있는 소스 NSID 값을 zero로 채워 이 사실을 나타낸다. 단명의 목적지 MKID와 목적지 IP 번지를 결합하여 사용하면 단명의 마스터키가 유일하게 구분되어진다.

"Cert Enc. Alg"는 I와 J의 유효기간이 긴 보증서를 암호하기 위하여 사용하는 알고리즘을 나타낸다. 이는 기본 SKIP 프로토콜에서 사용되는 DH 보증서와 동일하다. 이러한 보증서의 타입은 "Cert Type" 필드를 사용하여 나타낸다. 이 값이 단명의 보증서 타입으로 간주되어서는 안 된다.

이 보증서는 암호 키로서 g^{xj}의 하위 비트로부터 키 사이즈만큼을 사용하여 암호한다. 암호 알고리즘이 메시지마다 변수(예 : IV)를 필요로 하면 이 값은 g^{xy}의 상위 비트로부터 변수가 요구하는 길이의 비트를 추출하면 된다. I와 J만이 g^{xj}를 적절히 계산할 수 있으므로 I와 J의 보증서를 암호하는 것은 익명성이 요구될 때에 사용자의 익명성을 지원하게 된다. 이러한 익명성에 대한 보호는 능동적이거나 수동적인 형태의 공격자에게도 안전할 수 있다.

만일 "Cert Enc. Alg" 필드가 zero라면 유효기간이 긴 보증서는 지워진다. 이 경우 "Encrypted Long Lived Certificate" 필드에는 유효기간이 긴 DH 보증서 값이 지워져서 포함되어 있게 된다.

J가 암호된 유효기간이 긴 보증서를 수신하면 이 유효기간이 긴 DH 보증서를 복호하기 위하여 g^{xj}를 먼저 계산한다. 이 유효기간이 긴 보증서를 검증하고 확보하게 되면 J는 g^{ij}를 계산하고 K_{ij}를 얻어 단명의 보증서에 있는 "Certificate MAC" 필드를 검증하는데 사용한다. 만일 MAC 필드가 틀리면 이 단명의 보증서를 버려야 한다. MAC 필드 값이 옳으면 J는 EK_{ijn}을 위에서 설명한 방법으로 계산하여 자신의 단명의 보증서에 응답한다.

I가 단명의 보증서를 수신하면 EMKID_J_I 값을 사용하여 요구에 대한

정당한 응답 신호라는 것을 표시한다. EMKID_I_J 필드가 zero가 아니라는 것은 J에 의해 발신되었던 새로운 보증서에 대한 것과는 달리 단명의 보증서에 대한 응답신호가 I에 의해 나왔음을 나타낸다.

V. SKIP을 이용한 멀티캐스트 서비스

1. 그룹모드

SKIP은 멀티캐스트 트래픽에 사용될 수 있도록 확장되어 개발되고 있다. SKIP이 사용될 수 있는 두 가지 다른 모드가 있는데, 하나는 임시 멀티캐스트 그룹에 적합한 모형이고 다른 하나는 고정 멀티캐스트 그룹이나 브로드캐스트 그룹에 적합한 모형이다.[6]

임시 멀티캐스트 그룹에 사용되는 모드는 그룹 소유자가 존재하여 키 분배를 담당한다. 그룹 멤버들은 group interchange key(GIK)를 확보하기 위하여 그룹 소유자에게 키 전송을 요구한다. GIK는 키 암호용 키로서, AH와 ESP에 사용되는 키를 암호하는 기능을 갖고 있다. 발신자마다 패킷 암호 및 인증용 키를 선택하여 GIK로 암호하여 SKIP 헤더에 실어 보낸다.

멀티캐스트 그룹이 고정되어 있고 그룹 소유자가 없을 경우에는, GIK는 모든 그룹 멤버에게 수동으로 전달되어야 한다. 이 경우, GIK가 키 암호용 키로서 사용되지 않으나 GIK나 시각정보에 대한 MD5 해쉬용으로 사용된다.

SKIP에서는 GIK를 얻기 위해 안전한 유니캐스트 프로토콜을 정의하고 있다. 예견되는 그룹 멤버들이 실제로 그룹에 가입하고자 할 때에는 이 프로토콜을 사용하게 된다.

멀티캐스트 키 분배는 두 가지 환경에서 고려해 볼 수 있다. 하나는 임시 멀티캐스트 그룹이며 이를 이용한 응용분야로 비디오, 오디오를 이용한 화상회의를 생각할 수 있으며, 이런 환경에서는 매우 동적으로 멀티캐스트 번지를 할당하며 일정기간 사용하고 난 후에는 그 번지를 버린다.

두 번째 경우는 고정 멀티캐스트 그룹 환경이며, 잘 알려진 멀티캐스트 번지를 프로토콜을 신청한 사람들이 사용한다. 이와 같은 예는 라우팅 프로토콜이 네트워크에서 라우팅 정보를 전파하기 위하여 잘 알려진 멀티캐스트 번지를 사용하는 데서 찾을 수 있다.

브로드캐스트 IP는 두 번째 환경의 특수한 경우로 볼 수 있으며, 여기서는 잘 알려진 IP번지가 서브 네트워크에서 브로드캐스트용 번지로 이용된다. 이 프로토콜의 목표는 유니캐스트용 SKIP을 멀티캐스트용 SKIP으로 완성하는데 있다. 멀티캐스트 SKIP의 필드를 해당된 유니캐스트 SKIP의 필드에 일치시키는 시도가 추진되고 있다.

2. 임시 멀티캐스트 그룹

유니캐스트 IP용 SKIP에 단순히 삽입하는 방법으로 IP 멀티캐스트 응용프로그램에 확장된 키 관리기법을 제공한다. 임시 사용자 그룹을 안전하게 하기 위해서는 키 관리에 대한 인식을 갖고 멀티캐스트 그룹 가입 절차를 만들 필요가 있다.

더욱이, 멀티캐스트 keying material을 분배하려면 그룹 소유자 개념이 존재할 필요가 있다. 그룹 소유자를 확인하는 방법과 신청한 노드와 그룹 소유자가 통신하는 방법은 응용계층에서 담당한다. 그러나, 이러한 일들이 안전하게 처리되어야 하며, 그렇지 않으면 근원적인 키 관리가 파괴될 수 있다. 두 가지의 이점이 있는데, 하나는 멀티캐스트 그룹의 각 멤버들은 필요할 때면 언제든지 키를 변경하며 키 설정을 위하여 상호 통신하는 오버헤드가 없다. 패킷 암호키를 변경할 때 추가 통신에 대한 부담이 없으므로 대규모 그룹의 경우도 수초에 한번 정도씩 자주 키를 변경할 수 있다.

두 번째는, 모든 패킷 암호키는 랜덤하게 발생되며 서로 다르기 때문에 멀티캐스트에 스트림 암호기법이 사용되어도 문제가 발생되지 않는다. 스트림 암호기법이 사용될 때 암호된 트래픽의 각 소스에서는 서로 다른 키 스트림을 사용하는 것이므로 동일한 키 스트림 재사용의 문제가 생기지

않는다. 모든 멀티캐스트 그룹에 속한 멤버들이 동일한 패킷 암호키를 사용했다면 키 스트림의 재 사용문제로 인해 어떤 스트림 암호기법도 멀티캐스트 IP에는 사용될 수 없다.

3. 방송그룹 및 고정 멀티캐스트 그룹

브로드캐스트와 고정 멀티캐스트 그룹용의 키 분배 기법은 K_g가 고정되어 만료 일자가 없다는 것과 그룹에서 허가된 멤버 사이에서만 수동으로 키를 분배할 수 있다는 것을 제외하면 임시 멀티캐스트 그룹 방식과 유사하다.

고정 그룹에 사용되는 마스터키를 변경하는 방법은 매우 단순하고 자동적이며 확장이 가능하도록 허용하여, 암호 분석을 더욱 어렵게 만든다. 마스터키를 시간 단위로 변경하는 방식에서 카운터가 오버플로우 되는 시간이 길기 때문에 문제가 되지 않는다. 암호 기능을 갖는 멀티캐스트나 브로드캐스트 그룹을 초기화하기 위하여 수동으로 키를 분배하더라도 마스터키와 트래픽 보호키는 위에서 설명한 절차를 따라 자동으로 변경된다.

4. 신뢰도를 고려한 멀티캐스트 프로토콜

SKIP이 고정 멀티캐스트 그룹 환경에서 어떻게 사용될 수 있으며 신뢰성을 고려한 프로토콜인 RIPv2[RFC 1388]에 대해서 SKIP이 사용된다. RIP 프로토콜은 인증 타입을 위한 필드를 정의하였고 인증된 RIP 정보로 어떻게 MAC을 계산하고 분배하는가 하는 방법을 기술한다.

첫 번째 RIP 엔트리로서 인증 타입은 symbolic name SKIP으로 할당된 값을 갖는다. Authentication 필드는 MAC 값을 포함한다. 인증 범위는 AH에서의 범위와 동일하다.

RIP는 브로드캐스트용 프로토콜이기 때문에 유니캐스트의 키 관련 의미와는 다르다. 그러나 다른 프로토콜들도 유니캐스트, 임시 멀티캐스트 그룹, 고정 멀티캐스트 그리고 브로드캐스트 그룹 통신환경 중 어디에서나 이와 유사하게 SKIP과 함께 사용될 수 있다.

VI. 결 론

IP와 같은 비연결성 데이터그램 프로토콜과 이를 대치하기 위해 연구되는 IPv6에 특히 잘 정합되는 SKIP을 설명하였다. 이 기법에서 사용되는 프로토콜과 연산에 대한 오버헤드는 매우 적다. In-band signaled key는 공유키 암호방식의 블록 사이즈에 대한 키 길이 오버헤드만 존재한다. 또한 패킷 암호키를 발생하고 변경하는 것은 공유키 암호 동작에만 존재한다. 그래도 이 기법은 인증된 공개키 기반시설에 대한 확장성과 강인성을 갖고 있다. 더욱이 중간이나 최종 단말에서 문제가 생겼을 때 복구를 위해 복잡하게 고려할 필요가 없다.

이 논문에서는 보안 메커니즘을 네트워크 계층에 두어야 하는 이론적 근거를 제공한 후 SKIP이 어떻게 인터넷을 통하여 급속히 전개되어 사용되었는지를 설명하였다. 보안정책이나 보안통신을 할 수 있는 운영체계에 따라 인터넷의 상당 부분이 본질적으로 안전하게 되었다. 이는 두 통신 상대 사이에서 보안을 향상시키기도 하고 저하시키기도 하는 것을 허용하게 되었다. SKIP이 대부분의 기능을 제공하고 있다.

비록 단명의 보증서 교환 기법이 기존의 DH 기법의 구조를 사용하고 있지만 다른 기법 즉, DH의 변형 또는 타원곡선(EC : elliptic curve) 방식 등 키 교환 방식을 사용하여 생성될 수 있다. 이러한 변형된 DH 기법을 사용하기 위해서는 새로운 단명의 보증서 타입이 정의되고 적절한 파라미터를 포함하여야 한다. SKIP은 그룹모드, 임시 그룹, 방송 및 고정그룹, 신뢰도를 고려한 프로토콜 등 다양한 분야에 활용이 되며, 활용 분야는 점차적으로 확장될 것이다.

참고문헌

[1]Ashar Aziz, Tom Markson, Hemma Prafullchandra, "Simple Key-Management For Internet Protocols (SKIP)",

draft-ietf-ipsec-skip-07.txt, August 1996.
[2]T. Maufer, C. Semeria, "Introduction to IP Multicast Routing", draft-ietf-mboned-intro-multicast-00.txt, March 1997.
[3]S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
[4]김대호, 박응기, 김영수, "멀티캐스팅 네트웍에서의 정보보호를 위한 키 관리기술", KIISC REVIEW, 제9권 제3호, 1999.9.
[5]Ashar Aziz, "SKIP Extensions for Perfect Forward Secrecy (PFS)", URL:http://www.skip.org/spec/EPFS.html
[6]Ashar Aziz, Tom Markson, Hemma Prafullchandra, "SKIP Extensions for IP Multicast", URL:http://skip.incog.com/spec/EIPM.html

著者紹介

김대호(Dae-Ho Kim) 종신회원

1997년 2월:한양대학교 전자공학과 학사
1984년 8월:한양대학교 전자공학과 석사
1993년 Univ. of Maryland 방문연구원
1995년 전기통신기술사, 정보통신기술사
1977년 3월 ~ 현재 한국전자통신연구원 책임연구원
〈관심분야〉 전송분야, 통신 및 컴퓨터 보안

박응기(Eung-Ki Park) 정회원

1986년 2월:중앙대학교 전자계산학과 학사
1988년 2월:중앙대학교 전자계산학과 석사
1996년 3월 ~ 현재 아주대학교 컴퓨터공학과 박사과정
1988년 2월 ~ 현재 한국전자통신연구원 선임연구원
〈관심분야〉네트워크 정보보호, 컴퓨터정보보호

김영수(Young-Soo Kim) 정회원

1986년 2월:한남대학교 전자계산공학과 학사
1990년 2월:한남대학교 수학과 석사
1996년 3월 ~ 현재 한남대학교 컴퓨터공학과 박사과정
1986년 1월 ~ 현재 한국전자통신연구원 선임연구원
〈관심분야〉네트워크 정보보호, 컴퓨터정보보호