단순화된 IKE 프로토콜의 구현 및 테스트

정선화* · 손형선 · 박석천
*경원대학교 소프트웨어 학부

Design and Implementation of Simplified IKE Protocol

Sun-Hwa Jung* · Hyung-Seon Son · Seck-Cheon Park
*Division of Software, Kyungwon University
E-mail : scpark@kyungwon.ac.kr

요 약

전 세계적으로 안전한 데이터의 송수신에 대한 많은 연구가 지속되어 왔다. IPsec은 IP 패킷에 대해 기밀성과 무결성 그리고 인증과 같은 보안 서비스를 제공하는 국제 표준 프로토콜이며 IPsec의 키 관리 메커니즘이 IKE이다. 기존 IKE는 시스템간의 상호 운용성이 떨어질 뿐만 아니라 시스템의 성능을 저하시키며 서비스 거부 공격(DoS)에 대해 매우 취약한 구조를 가지고 있다. 본 논문에서는 이러한 문제점을 해결하기 위해서 기존 IKE 프로토콜에서 사용되지 않는 모드를 삭제하고 인증 방법을 전자서명과 공유된 비밀키에 기반한 방식으로 단일화시켰다. 또한 DoS 공격을 예방할 수 있도록 단순화된 IKE 프로토콜을 구현하고 테스트하였다.

키워드

IKE, IPsec, Dos, Security

Ⅰ. 서 론

전 세계적으로 안전한 데이터의 송수신에 대한 많은 관심과 연구가 지속되어 왔다. IPsec(Internet Protocol security)은 IP(Internet Protocol) 패킷(Packet)에 대하여 기밀성과 무결성 그리고 인증과 같은 보안 서비스를 제공하는 국제 표준 프로토콜이다. IPsec은 AH(Authentication Header)와 ESP(Encapsulating Security Payload), 그리고 IKE(Internet Key Exchange)라는 세부 프로토콜들로 구성되어 있으며 강력한 암호학적 알고리즘과 프로토콜을 이용하여 안전한 보안 서비스를 제공해주고 있다. IPsec의 AH와 ESP를 통해 제공되는 무결성과 기밀성은 송수신자가 같은 키를 공유한 후 공유된 키를 이용해 대칭키 암호 알고리즘, 또는 HMAC(Hash Message Authentication Code)와 같은 함수를 통해 제공된다. 이때 송수신자가 같은 키를 공유할 수 있도록 해주는 메커니즘이 IKE이다. 그러나 IPsec은 전체적인 시스템의 복잡성(complexity)은 구현된 시스템의 상호 호환(interoperability)을 어렵게 할 뿐만 아니라 서비스 거부 공격(DoS : Denial of Service)에 취약하다는 문제점을 비롯해 보다 향상된 안전성의 보장이 중요한 해결과제로 지적되고 있다.

Ⅱ. IKE 프로토콜

IKE는 IPsec에서 사용할 SA나 ISAKMP에서 사용하기 위한 인증된 키 재료들을 얻기 위하여 ISAKMP와 연계하여 Oakley의 일부와 SKEME(Security Key Exchange Mechanism)의 일부를 이용한 합성 프로토콜이다. ISAKMP는 인증 및 키 교환을 위한 프레임워크로서 1단계와 2단계를 제공하고, Oakley는 각각의 대상이 자신의 속도에 맞게 프로토콜의 상태를 진행해 나갈 수 있는 자유로운 형식의 키 교환 모드를 정의하며, SKEME는 키 공유 및 re-keying 기법을 제공한다.

IKE 프로토콜은 메인(main) 모드와 어그레시브(aggressive) 모드로 이루어진다. 이 두 모드는 DH 키 교환에 의해서 키 교환을 위한 두 IKE 개체간의 SA를 생성하기 위한 것이다. IKE는 IKE SA를 생성

하는 1단계 키 교환과 실제 IPsec 프로토콜이 사용할 SA를 정의하는 2단계 키 교환으로 이루어진다.

III. 단순화된 IKE 프로토콜 설계

본 장에서는 IP 기반의 보안 서비스를 제공하기 위한 IPsec의 키 관리 프로토콜인 IKE의 복잡성과 안전성을 개선한 단순화된 IKE 프로토콜을 설계하였으며 설계한 단순화된 IKE 프로토콜에서 사용될 메시지는 기존의 IKE 프로토콜의 헤더 및 페이로드를 기반으로 최소한의 수정을 통하여 정의하였다. 모든 IKE 메시지는 HDR로 표시되는 IKE 헤더가 붙는다. 헤더 뒤에는 하나 이상의 페이로드들이 올 수 있으며 이들은 이전 페이로드의 "Next Payload" 필드에 의해 식별된다. 인증 페이로드(AUTH)는 사용자 인증에 필요한 인증 정보를 전달하기 위해 사용된다. 이는 기존의 4가지 종류의 인증 방식에서 2가지로 단일화된 전자서명과 사전 공유 키를 이용한 인증 방식을 지원하기 위해 설계하였다. 인증 데이터는 인증 방법에 따라 그 형태가 달라진다.

3.1. 단순화된 IKE 프로토콜 동작절차

IKE를 위해서는 실제 데이터를 보호하기 위한 AH나 ESP 같은 프로토콜에서 사용할 알고리즘과 키 값(IPsec_SA) 뿐만 아니라 IKE 메시지 자체를 보호하기 위한 알고리즘과 키 값(IKE_SA)도 설정되어야 한다. 설계한 IKE는 기존의 IKE에서처럼 2단계 방식을 사용하고 있고 1단계는 4개의 메시지(2개의 request/response 쌍)로 이루어졌으며 IKE_SA를 설정한다.

1단계는 IKE_SA_INIT과 IKE_AUTH라는 2개의 라운드로 구성되어있으며 각 라운드는 request와 response의 메시지 쌍을 가지기 때문에 총 4개의 메시지를 주고받도록 설계하였다. 이는 기존의 IKE의 불필요한 과도한 6번의 메시지 교환을 개선하여 설계한 것이다. 다음에 수행되는 1단계의 두 번째 쌍(IKE_AUTH_request /response)은 서로를 확인하기 위한 정보와 인증(메시지/사용자 인증)에 사용될 정보를 교환한다. 두 번째 라운드의 메시지들은 첫 번째 라운드에서 설정된 키 값에 의해 보호받는다. 1단계 통신의 메시지 교환은 그림 1과 같다.

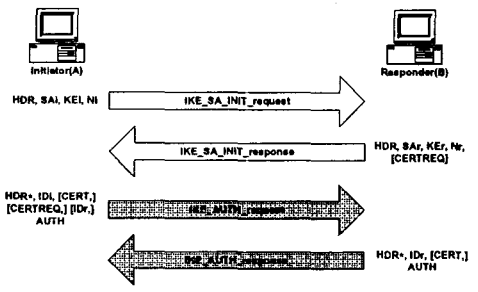

그림 4. 1단계 통신의 메시지 교환

IKE 프로토콜은 위조된 IP 주소들을 이용해 무수히 많은 연결을 시도하여 응답자의 메모리와 CPU를 소모시키는 형태의 DoS 공격이 있을 수 있다. 이 경우 응답자는 IKE_SA_INIT_request의 응답으로 IKE_SA_INIT_response 메시지를 전송하고, KEi와 KEr을 이용해 많은 연산이 필요한 DH 값을 계산해야 할 뿐만 아니라 송신자의 IKE_SA_INIT_request와 관련된 상태 정보(IP, SPI, nonce 등)들을 저장해야한다. 이러한 방식은 응답자의 리소스를 점유함으로써 응답자가 정상적인 연결 요청에 응답 못하게 하는 결과를 초래할 수도 있다. 이와 같은 공격은 연결을 시도하는 당사자가 주장하는 IP가 유효한(실제로 IKE 통신을 원하는) 주소인지가 확인되기 전까지 어떠한 상태 저장이나 CPU 소모(공개키의 지수 연산 등)를 하지 않게 함으로써 예방할 수 있으며 본 논문에서는 "stateless cookie"를 이용해 이를 설계하였다. DoS 공격에 대응한 1단계 통신의 메시지 교환은 그림 2와 같다.

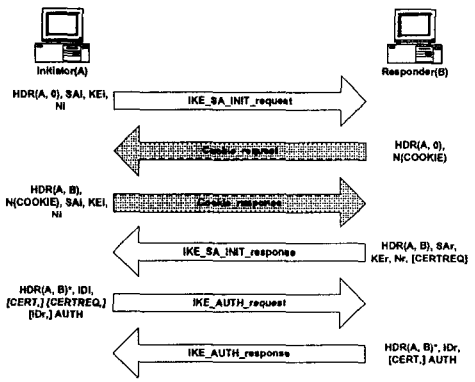

그림 5. DoS 공격에 대응한 1단계 통신

3.2. 알고리즘

본 절에서는 단순화된 IKE 프로토콜의 동작 알고리즘을 설계하였다. 단순화된 IKE 프로토콜은 1단계 통신에서 기존의 6번의 메시지 교환을 4번의 교환만으로 가능하게 설계하였다. IKE 통신을 원하는 노드에서 1번(IKE_SA_INIT_request)과 3번(IKE_AUTH_request) 메시지를 전송하고,상대방 노드는 이에 대한 응답으로 2번(IKE_SA_INIT_response)과 4번(IKE_AUTH_ response) 메시지를 전송하게된다. 각 단계에 따라 메시지가 만들어지고 암호화가 필요한 메시지라면 Crypt 함수를 호출하여 메시지를 암호화한다. 이렇게 생성된 메시지는 Sendto 함수를 통하여 상대편 서버로 전송된다. 전송 후 타이머를 설정하여 타이머가 완료될 때까지 응답이 오지 않으면 동일 메시지를 재전송한다. 단순화된 IKE 프로토콜의 송신부와 수신부의 동작 알고리즘은 그림 3과 같이 설계하였다. 알고리즘에서 보듯이 IKE 1단계일 경우 총 4번의 메시지를 송수신하

고 IKE 2단계일 경우 총 3번의 메시지를 송수신하고 종료하게 된다.

```
**********************************************************
************ 실제로 메시지를 주고받는 함수 ********************
**********************************************************

Recvfrom(msg);                                    //메시지 읽기
Sendto(msg);                                      //메시지 보내기
Alarm(time);                                      //타이머 설정
Message_analysis(msg);                            //메시지 분석
Make_message(num);                                //메시지 생성
Crypt(msg);                                       //암복호화
Copy_Keydata(msg);                        //키 재료를 메모리에 복사
int m_num;                                        //메시지 번호

int R = SEND;
if(Phase == 1){                                   //IKE 1단계
  Start = 0; End = 4; }
else{                                             //IKE 2단계
  Start = 4;  End = 7; }
for(m_num=Start ; m_num<End ; m_num++){           //단계별 교환
                                                  //횟수동안 루프
  if(R = RECEIVE) {                               //수신부
    Recvfrom(IKE_MSG)
    Alarm(0);
    if (encrypted){
      Message_analysis(Crypt(IKE_MSG)); }
    else{
      Message_analysis(IKE_MSG); }
    Copy_keydata(Message_analysis(IKE_MSG));
    R = SEND;
  }
  else{                                           //송신부
    Make_message(m_num);
    if(encrypt){
      Crypt(IKE_MSG); }
    Sendto(IKE_MSG);
    Alarm(wait_time);
    R = RECEIVE;
  }
}
```

그림 3. 송수신부 동작 알고리즘

IV. 단순화된 IKE 프로토콜의 구현 및 테스트

4.1. 단순화된 IKE 프로토콜의 구현

본 논문에서 구현한 단순화된 IKE 프로토콜의 구현 환경은 표 1과 같으며 운영체제로 Linux 커널 2.4.18을 이용하였고 언어는 gcc, gdb 및 python을 사용하였다.

표 1. 구현환경

소프트웨어	운영체제	Linux 커널 2.2.18
	언어	gcc, gdb, python
하드웨어	VPN 게이트웨이	펜티엄4 2.0G 512 RAM

기존의 IKE 프로토콜의 복잡성과 안정성을 고려하여 설계한 단순화된 IKE 프로토콜을 구현한 시스템 구성도는 그림 4와 같다. IPsec ESP 패킷을 만들기 위해서는 어떤 암호화 알고리즘을 사용할 것인지, 어떤 키를 이용하여서 암호화를 할 것인지, 어떤 암호화 모드를 이용할 것인지를 미리 정의해야 할 필요가 있다. 이와 같은 정의 내용(SA)은 IKE 프로토콜을 이용하여 두 VPN 게이트웨이 사이에 통신을 통해 협상, 교환되고 SA는 SPD(Security Policy Database)라는 보안 규약에 관련된 정보를 갖는 데이터베이스와 SAD(Security Association Database)라는 보안 연계에 관련된 정보를 갖는 데이터베이스로 구성되어있다. 따라서 구현한 시스템은 III장에서 설계한 단순화된 IKE 프로토콜을 이용하여 두 VPN 게이트웨이간에 SA를 협상, 교환하고 그 SA를 바탕으로 IP 계층에서 패킷을 ESP를 통하여 암복호화하여 통신한다.

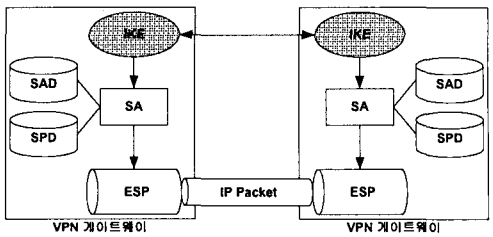

그림 4. 시스템 구성도

본 논문에서 구현한 단순화된 IKE 프로토콜은 UDP 소켓 500번 포트를 통해 키 교환을 위한 데이터들을 주고 받으며, 교환 과정은 III장에서 설계한 단순화된 IKE 프로토콜을 이용하여 1단계에서 4번, 2단계에서 3번의 통신이 이루어지고 2단계의 경우 SA Lifetime이 경과하는 경우 재수행이 되어 새로운 키재료를 생성하도록 구현하였다.

4.2. 단순화된 IKE 프로토콜의 테스트 및 검토

본 논문에서 구현한 단순화된 IKE 프로토콜의 테스트를 위하여 전체 시스템에 대한 테스트 모델을 구성하고 패킷 캡쳐 프로그램을 이용하여 IKE가 정상적인 암호화 통신을 수행하는지 테스트하고 그 결과를 검토하였다.

가. 단순화된 IKE 프로토콜의 테스트

본 논문에서 단순화된 IKE 프로토콜을 테스트하기 위한 전체 시스템 구성도는 그림 5와 같다.

그림 5. 전체 시스템 구성도

두 VPN 게이트웨이간에 단순화된 IKE 프로토콜을 통한 암호화된 통신이 정상적으로 동작하는지 확인하기 위하여 라우터를 사이에 둔 두 대의 VPN 게이트웨이로 전체 시스템을 구성하였다. 그리고 각 VPN 게이트웨이에는 구현한 단순화된 IKE 프로토콜을 설치하고 이를 통하여 두 VPN 게이트웨이간에 SA를 협상, 교환하고 그 SA를 바탕으로 IP 계층에서 패킷을 ESP를 통하여 암호화하여 통신한다. 두 VPN 게이트웨이간의 패킷을 캡쳐하기 위하여 ethereal이라는 모니터링 프로그램을 사용하였다. 이 모니터링 프로그램을 사용하여 먼저 IPsec 통신을 적용하지 않은 경우와 단순화된 IKE 프로토콜을 통하여 IPsec을 적용한 경우를 비교분석하였다. IPsec 통신을 적용하기 전의 ping 테스트의 경우에는 전송되는 데이터의 상세한 내용까지 모두 알 수 있었으며 IPsec 통신을 적용한 경우에는 모든 전송 데이터가 ESP 프로토콜에 의하여 암호화되어 전송되는 패킷의 어떠한 데이터나 정보 등도 알 수 없음을 시험을 통하여 확인하였다.

나. 구현 프로토콜의 검토

본 논문에서 설계 및 구현한 프로토콜의 테스트를 위하여 테스트 모델을 구성하고 각 게이트웨이에 구현한 IKE 프로토콜을 이용한 VPN을 설치하였다. 그 후 패킷 캡쳐 프로그램을 통하여 III장에서 설계한 단순화된 IKE 프로토콜의 동작절차에 SA를 협상한 뒤 생성된 SA를 이용하여 IPsec 통신을 수행하는 것을 확인할 수 있었다. 또한 기존 IKE 프로토콜과 비교를 통하여 구현한 프로토콜이 암호화된 통신을 정상적으로 수행함을 확인하였다.

V. 결 론

본 논문에서는 기존 IKE의 문제점을 해결할 수 있는 새로운 IKE 프로토콜을 설계하고 구현하였다. 이를 위해 먼저 기존 IKE 프로토콜의 2가지 모드(메인, 어그레시브)와 4가지 인증 방법(전자서명, 공개키 기반, 수정된 공개키 기반, 공유된 비밀키) 중 사용되지 않는 어그레시브 모드와 공개키 기반, 수정된 공개키 기반의 인증 방법을 제거하고 전자서명과 공유된 비밀키에 기반한 방식으로 단일화시켜 기존 IKE 프로토콜을 단순화시켰다. 또한 기존 프로토콜이 DoS 공격에 취약한 보안상의 문제점을 해결하기 위하여 DoS 공격시 "stateless cookie"를 이용하여 연결을 시도하는 당사자가 주장하는 IP가 유효한(실제로 IKE 통신을 원하는) 주소인지 확인되기 전까지 어떠한 상태 저장이나 CPU 소모(공개키의 지수 연산 등)를 하지 않게 함으로써 DoS 공격을 예방할 수 있는 메커니즘을 추가하였다. 그리고 기존 IKE가 가지고 있던 대부분의 특징 및 속성들(identity hiding, PFS, two phases 등)을 그대로 유지하면서 최소한의 수정만으로 단순화된 IKE 프로토콜을 설계하였다. 설계한 단순화된 IKE 프로토콜을 구현하기 위해 Linux 환경에서 gcc와 gdb 언어를 사용하였으며 설계한 프로토콜의 동작성 테스트를 위해 테스트 모델을 구성하여 본 논문에서 설계 및 구현한 단순화된 IKE 프로토콜의 테스트하였다. 테스트를 수행한 결과 기존 IKE 프로토콜의 복잡성을 개선한 단순화된 IKE 프로토콜이 정상적으로 동작함을 확인하였다.

참고문헌

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

[2] D. Piper, "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, Nov. 1998.

[3] D. Maughan, "Internet Security Association and Key Management Protocol", RFC 2408, Nov. 1998.

[4] D. Harking and D. Carrel, "The Internet Key Exchange", RFC2409, Nov. 1998.

[5] P. Hoffman, "Features of Proposed Successors to IKE", IETF WG, May 31. 2002.

[6] 강권학, "ISAKMP 프로토콜", 안철수연구소, 2002.

[7] 이정훈 외, "VPN을 위한 자동 키 관리", 비트 프로젝트 67호.