情 報 保 護 學 會 誌
第 16 卷 第 3 號, 2006. 6

IKEv2 설계 및 구현

엄 희 정*, 김 락 현**, 염 흥 열***

요 약

인터넷을 이용한 생활 패턴의 변화와 정보의 공유가 점차 이동 망과 무선망을 통해 확장되어, 이제는 유·무선의 경계가 없는 통신환경에서 살게 되었다. 이와 같은 접속환경의 변화와 확장은 더욱 네트워크 계층에서의 보안을 필요하게 만들었고, 이를 제공하는 IPSec은 네트워크 계층에서 IP 데이터 그램에 대한 인증, 무결성, 기밀성을 제공해 주는 표준 프로토콜로써 AH(Authentication Header), ESP(Encapsulation Security Payload), IKE(Internet Key Exchange)로 구성되며, 향후 통신의 발달과 함께 연구의 중요성이 부각되고 있다. 이 중에서 IKE는 IPSec에서 사용하는 보안연계(SA)를 자동으로 설정하는 기능을 가지고 있으며, 상호 인증 및 키 교환을 하는 하이브리드 프로토콜이다. IKEv1은 ISAKMP, DOI, SKEME등으로 구성되어 있다. 그러나 IKEv1은 상호호환성이 부족하기 때문에 구현이 용이하지 않으며, 서비스 거부 공격에 취약한 구조로 되어있다. IKEv2는 이러한 단점을 보안하기 위해 출현하였다. 본 논문에서는 IKEv2 설계 시 요구사항에 대하여 알아보고, 그 설계를 기반으로 구현하였다.

Ⅰ. 서 론

인터넷을 기반으로 한 유·무선 통신망의 발달과 통신매체의 발전으로, 이를 이용한 각종 서비스와 응용이 발전하고 있다. 또한 사용자들이 인터넷에 접속하는 것이 점점 용이해지고 있다. 그에 따라 인터넷 접속 방법 또한 다양화 되고 있고, 전자상거래, 은행 보안 등 보안에 민감한 어플리케이션들이 최근 증가하고 있다. 이에 따라 Application 계층보안 보다 기존 레거시 시스템을 그대로 이용할 수 있는 보안의 필요성은 명확해졌고, 이러한 필요성을 해결하기 위해 IETF의 Working Group IPSEC은 현재 차세대 네트워크 계층 보안을 고려중에 있으며, IETF는 IPv6 구현 시 IPSec을 필수사항으로 채택하였다. IPSec은 네트워크 계층에서 IP(Internet Protocol) 데이터 그램에 대한 무결성, 기밀성, 접근제어를 제공해주는 표준 프로토콜이다. 이것은 사용자 인증서비스를 제공해주는 기능을 갖는 AH(Authentication Header)와 메시지 인증과 무결성 그리고 기밀성을 제공하는 ESP (Encapsulation Security Payload)를 포함하며, 인증과 기밀성을 제공하기 위해서 네트워크 노드는 비밀 정보 교환을 필요로 한다. 이와 같은 정보를 교환하고 자동으로 SA(Security Association) 설정 기능을 제공하기 위해 IPSec은 IKE(Internet Key Exchange)를 이용하고[1.2.4.6] , ISAKMP (Internet Security Association and Key Management Protocol)에 의해 정의된 프레임 워크를 사용하는 하이브리드 프로토콜이다 [6] . 그리고 IKE는 Oakley 키 결정 프로토콜과 SKME("A Versatile Secure Key Exchange Mechanism for Internet)의 영향을 받아서 부분적으로 수용하고 있다. 이런 이유로 초창기에 IKE 프로토콜은 ISAKMP/Oakley 프로토콜로 불렸다. Oakley 키 결정 프로토콜은 일련의 키 교환 '모드'를 정의 하고 있는데, 각 모드는 키의 PFS(Perfect Forward Secrecy), ID 보호 (Identity Protection) 또는 인증(Authentication) 서비스를 제공한다. SKEME는 융통성 있는 키 교환 기법을 설명하고 있는데, 익명성, 부인봉쇄 그

본 논문은 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구 결과로 수행되었음
(IITA-2005-(C1019-0502-0020))
* (주)인젠 선임연구원 (crowehj@korea.com)
** 순천향대학교 정보보호학과 (rhkim@sch.ac.kr)
*** 순천향대학교 정보보호학과 (hyyoum@sch.ac.kr)

리고 신속한 키 업데이트를 지원한다. IKE를 표준문서의 기준으로 살펴보면, 1998년에 표준으로 제정된 IKE(i.e. IKEv1)는 RFC 2407과 2408, 2409의 3개 문서로 구성되어있으며, 이들은 많은 부분들을 상호 참조(Cross reference)하고 있다. 이러한 여러 문서로 되어 있는 문서를 하나로 통합하고, IKEv1의 단점인 용이하지 않은 구현으로 상호호환성이 떨어지고 서비스 공격에 취약하다는 단점을 보안하기 위해 IKEv2가 출현하게 되었다[7]. 본 논문에서는 IKEv2를 설계 및 구현함으로써 IKEv2가 가지고 있는 장점을 분석하고, 더 나아가 구현 시 실질적으로 어떤 부분이 더 필요한지를 고찰한다. Ⅰ장 서론에서는 IKEv2에 대한 소개를 하고, Ⅱ장에서는 IKEv2 프로토콜의 인증과정에 대해서 기술하며, 이를 기반으로 Ⅲ장에서는 IKEv2 설계에 관하여 기술한다. Ⅳ장에서는 테스트 및 구현결과에 대해 다루고 마지막으로 Ⅴ장에서는 IKEv2의 장점 및 향후 연구계획에 대해서 기술한다.

Ⅱ. IKEv2의 인증과 키교환 과정 예

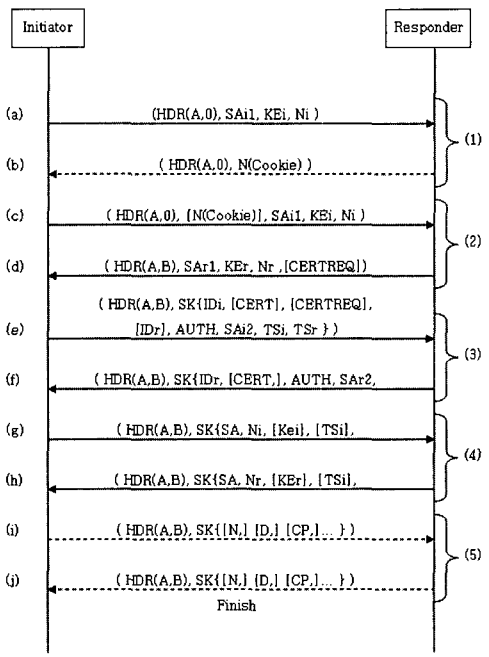
IKEv2는 인증과 세션키를 공유하기 위한 하나의 프로토콜이다. 그리고 IKEv2는 네트워크 노드들 사이에서 IPSec SA를 자동으로 설정해주는 기능을 제공한다. 그리고 IKEv2에서 네트워크 노드를 사이의 인증은 AUTH Payload을 계산함으로써 이루어진다. 본 장에서는 IKEv2의 인증과 키교환 과정에 대한 절차에 대해 소개한다. 표 1은 IKEv2의 파라미터와 의미를 나타낸다.

그림 1은 IKEv2의 메시지의 흐름을 나타내고, 이 흐름은 I(Initiator)와 R(Responder)사이에서 발생한다. 일반적으로 Initiator는 클라이언트 사용자를 말하며, Responder는 인증자이다.

IKEv2는 일반적으로 3개의 라운드 트랩으로 구성되고 서비스 거부 공격 방지 라운드 트랩을 사용하지 않으면 2개의 라운드 트랩으로 구성되어 진다. 첫 번째로 I는 HDR과 I에 의해 제공되는 암호학적 슈트를 포함하는 IKE_SA(SAi1), Diffie-Hellman[10] 값을 포함하는 KEi, 그리고 I가 생성한 Nonce를 포함하는 IKE_SA_INIT(2) Request(c) 메시지를 R에게 전송한다. 다음에 우리는 선택적으로 서비스 거부 공격 방지를 위한 라운드 트랩(b,c)을 이용할 수 있다. 메시지 교환 후 I가 제공한 암호학적 슈트 중에 선택한 것을 포함하는 SAr1과 Diffie-Hellman 키 교환을 완성하기 위한 KEr과 R이 생성한 Nonce(Nr)를 포함하는 메시지 (d)를 R이 I에게 전

〔표 1〕 IKE 약어 및 의미

표 기	의 미
I	Initiator(IKE 요청자)
R	Responder(IKE 응답자)
HDR	IKE 헤더
AUTH	사용자 인증 목적을 위해 사용되는 데이타
SAi	IPSec에서 보안연계 (Initiator가 제시)
SAr	IPSec에서 보안연계 (Responder가 선택)
KEi	DH 값(Initiator)
KEr	DH 값(Responder)
Ni	Replay Attack방지용 랜덤값 (Initiator)
Nr	Replay Attack방지용 랜덤값 (Responder)
IDi	Initiator 식별자
IDr	Responder 식별자
N(Cookie)	Responder가 생성 및 검증, stateless cookie
HDR(A,0)	IKE Header이고 I는 A라는 SPI를 사용, R은 부여받은 SPI가 없음
SKEYSEED	세션키를 만들 때 이용되는 값



〔그림 1〕 IKEv2 인증 및 키교환 과정

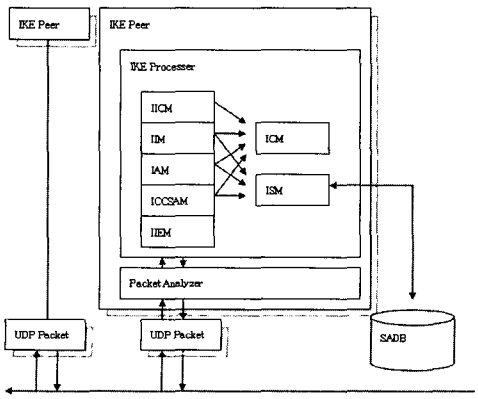
송한다. 이 단계에서 I와 R은 Ni, Nr 그리고 Diffie-Hellman 공유정보를 이용하여 PRF (Pseudo Random Function)사용해서 계산된 SKEYSEED를 생성하게 된다.(PRF는 I와 R이 협상한 SA에 포함되어 있다) IKE_SA을 위한 Keying Material은 SKEYSEED로부터 도출된다. IKEv2 메시지 중에 헤더를 제외한 나머지는 암호화되고 메시지 무결성이 지원되며 SK{...}으로 표기된다. I는 IKE_ AUTH(3) Request 메시지(e)를 R에게 전송한다. I는 자신의 식별자를 검증받기 위한 IDi와 메시지(e,f)의 무결성을 보호하기 위한 AUTH Payload를 전송한다. 그리고 인증서(CERT)와 인증기관 리스트, 예를 들어 CAs의 이름 그리고 공개키 (CERTREQ) [11] , IP Address내에서 다중 ID로 서비스하고 있는 I가 R중에 통신하고자 하는 R을 선택하는 기능을 가진 선택 가능한 IDr Payload를 송신한다. R은 메시지 (f)를 전송한다.

R은 그 다음에 IDr을 검증하고, 선택 가능하며 하나 또는 그 이상의 CERT 그리고 자신임을 증명하는 AUTH Payload를 포함해서 전송한다. 메시지 (g,h)는 IKE_CREATE_CHILD_SA(4) 이고 메시지(i,j)는 Information(5) 메시지이다. 이 메시지는 선택 가능하고, 언제든지 I 또는 R에 의해 전송될 수 있다. CREATE_CHILD_SA 메시지는 양쪽 노드들이 이미 기존에 SA를 가지고 있거나 만료 또는 서로 다른 보안 정책을 가지고 있을 때, 새로운 SA를 다시 생성하기 위해 사용되고, 이것은 IKE_AUTH메시지에서 인증 부분이 없어진 것과 유사한 메시지이다. I는 메시지(g)를 새로운 SA와 새로운 키 교환을 위한 KEi을 포함해서 R에게 전송한다. R은 메시지 (f)를 I가 제공한 SA중에 선택한 SA와 DH값을 포함하는 KEr을 I에게 전송한다. Information 메시지 (h,i)는 선택 가능한 메시지이고 오류 통지 또는 노드들 사이의 환경설정을 하기 위해 사용되어 진다.

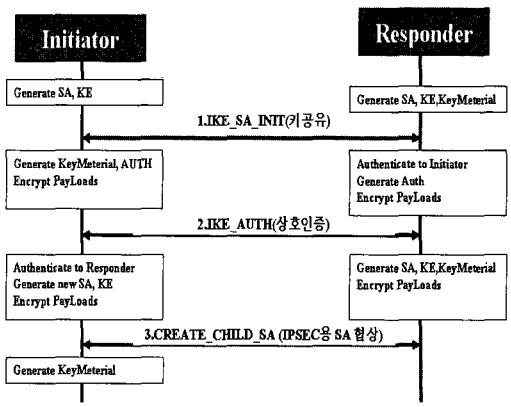
Ⅲ. IKEv2 설계

이 장에서는 우리가 구현한 IKEv2 구조에 대해 설명한다. 우리가 구현한 IKEv2구조는 크게 IKE 처리부와 패킷 분석기로 나눌 수 있고 이 구조는 IKE Peer인 Initiator와 Responder 모두 동일한 구조를 갖는다. IKE 처리부는 그림 2와 같이 7개의 모듈로 나눌 수 있고, 처리부의 역할은 IKE 시작에서 종료 시까지 수행하는 IKE_INIT, IKE_AUTH, CREATE_CHILD_SA 메시지를 처리한다. IKE메시지 분석기는 IKE메시지를 분석해서 IKE처리부에게 전송하고 IKE Peer에게 IKE Payload를 생성하여 전달하는 역할을 담당한다. 아래 그림 2는 IKEv2의 구성요소에 대한 모듈별 관계를 설명하고, 그림 3은 IKEv2의 시나리오를 나타낸 것이다.

IICM(IKE Init Cookie Module)은 서비스 공격 방지를 위해 IKE Peer가 Cookie의 생성 및 검증 역할을 한다. IAM(IKE Auth Module)은 IKE에서 사용할 세션키 및 인증값 생성 그리고 검증 역할을 한다. ICCSAM(IKE Create Child SA Module)은 Child SA 메시지 교환 시 세션키 및 SA 생성 및 선택 역할을 한다. IIEM(IKE Information Exchange Module)은 에러 정보를 처리한다. ICM(IKE Crypto Module)은 IKE에서 사용하는 암호모듈이며 암호화 및 무결성 값을 생성하는 역할을



〔그림 2〕 IKEv2 Peer 구성 요소 간 연동구조



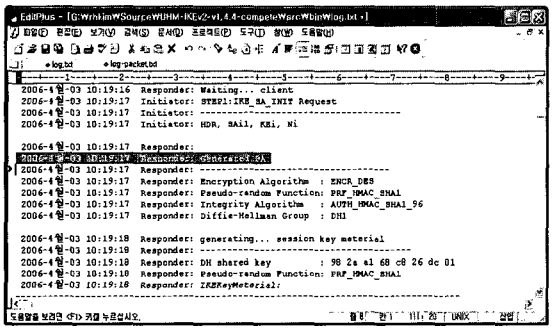
〔그림 3〕 IKEv2 Protocol 시나리오

한다. ISM(IKE Storage Module)은 SA와 Key Material을 저장하는 SADB를 연결하는 Interface이다. 예를 들어, UDP 포트 500을 통해 수신된 IKE Payload는 패킷 분석기를 통해 분석되고 분석된 Payload가 Cookie 메시지가 포함되어 있으면 IICM에서 이 값을 검증하고, 그 결과를 IIM 모듈을 통해 생성한다. 이때 사용할 SA를 ISM 모듈을 통해 SADB에서 가져오게 된다. 다음에 IIM모듈에서는 IKE_INIT Response 메시지 구성 후 패킷분석기로 보내고, 패킷분석기에서는 Payload를 생성한 후 Initiator에게 전송한다. 다음으로 IKE_ AUTH 메시지가 수신되면 IAM 모듈에서는 ICM 모듈을 통해 세션키 생성 및 인증을 하고, 다시 IAM 모듈에서 IKE_AUTH Response 메시지를 구성한 후 패킷분석기를 통해 Initiator에게 전송한다. IKE_CREATE_Child_SA 메시지가 수신되면 ICM을 통해 새로운 세션키를 생성하고, ICCSAM 모듈에서 ISA를 통해 사용할 SA을 선택하여 Initiator에게 전송한다.

Ⅳ. IKEv2 구현 환경 및 결과

본 논문에서는 IKE 프로토콜을 테스트하기 위해 유선환경에서 사용자 PC를 이용하여 네트워크를 구성하였다. 개발도구는 Java 1.4.2 sdk, 테스트 환경은 Windows XP Professional에서 Initiator와 Responder를 구현하고 테스트 하였다. Initiator와 Responder를 사용자 PC에 두고, 서로 키 교환과 상호인증과정을 테스트 하였다. IKEv2 상호인증과 키 교환 각 단계에 대한 결과는 그림 4~9와 같다.

그림 4와 5는 IKE INIT Request와 Response 단계이다. IKE INIT 단계는 IKE_SA의 협상 및 설정을 위해 Diffie-Hellman 키 교환을 수행하고 nonce 값과 필요한 파라미터(parameter)들을 교환한다. IKE_SA_INIT이 완료된 이후의 모든 메시지들은 설정된 IKE_SA에 의해 암호학적인 보호를 받는

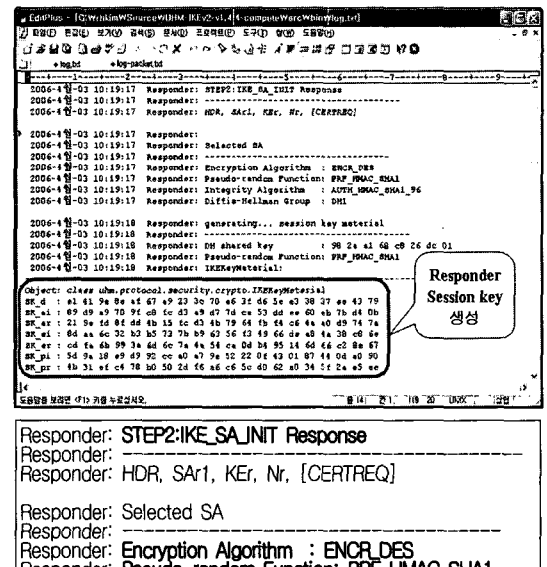


Responder: Waiting... client
Initiator: **STEP1:IKE_SA_INIT Request**
Initiator: ----------
Initiator: HDR, SAi1, KEi, Ni

Responder: Generated SA
Responder: ----------
Responder: **Encryption Algorithm : ENCR_DES**
Responder: **Pseudo-random Function: PRF_HMAC_SHA1**
Responder: **Integrity Algorithm : AUTH_HMAC_SHA1_96**
Responder: **Diffie-Hellman Group : DH1**

Responder: generating... session key material
Responder: ----------
Responder: **DH shared key : 98 2a a1 68 c8 26 dc 01**
Responder: Pseudo-random Function: PRF_HMAC_SHA1

(그림 4) IKE_INIT Request



Responder: **STEP2:IKE_SA_INIT Response**
Responder: ----------
Responder: HDR, SAr1, KEr, Nr, [CERTREQ]

Responder: Selected SA
Responder: ----------
Responder: **Encryption Algorithm : ENCR_DES**
Responder: **Pseudo-random Function: PRF_HMAC_SHA1**
Responder: **Integrity Algorithm : AUTH_HMAC_SHA1_96**
Responder: **Diffie-Hellman Group : DH1**

Responder: generating... session key material
Responder: ----------
Responder: DH shared key : **98 2a a1 68 c8 26 dc 01**
Responder: Pseudo-random Function: PRF_HMAC_SHA1
Responder: IKEKeyMeterial:

Object: class uhm.protocol.security.crypto.IKEKeyMeterial
SK_d : e1 41 9e 8e af 67 a9 23 3e 70 e6 3f d6 5e e3 38 37 ee 43 79
SK_ai : 89 d9 a9 70 9f c8 fc d3 a9 d7 7d ca 53 dd ee 60 eb 7b d4 0b
SK_ar : 21 9e fd 8f dd 4b 15 fc d3 4b 79 64 fb f4 c6 4a a0 d9 74 7a
SK_ei : ed aa 6c 32 b3 b5 73 7b b9 63 56 f3 49 66 de a8 4a 38 c8 6e
SK_er : cd fa 6b 99 3a 6d 6c 7a 4a 54 ca 0d b4 95 14 6d 66 c2 8e 67
SK_pi : 5d 9a 18 e9 d9 92 cc a0 a7 9e 52 22 0f 43 01 87 44 0d a0 90
SK_pr : 4b 31 ef c4 78 b0 50 2d f6 a6 c6 5c d0 62 a0 34 5f 2a e5 ee

(그림 5) IKE_INIT Response

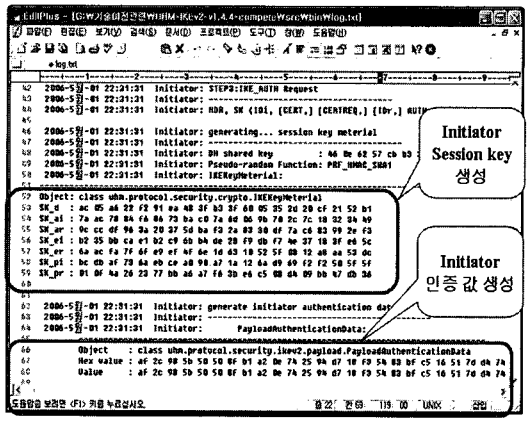
다. 그림 4의 IKE INIT Request에서 Initiator는 Responder에게 SPIs, IKE의 버전, 그리고 다양한 종류의 플래그등을 포함하는 HDR, Initiator가 IKE_SA를 위해 지원할 수 있는 암호 알고리즘 등을 명시한 SAi1, Diffie-Hellman 값인 KEi 그리고 nonce를 송신한다. IKE INIT Request를 수신한 Responder는 IKE INIT Responder는 암호알고리즘으로 DES를, 의사난수함수로 HMAC-SHA1을, 무결성 알고리즘으로 AUTH_HMAC_SHA1_96를 그리고 Diffie-Hellman 그룹으로 DH1을 선택하고, 이를 이용하여 세션키 요소값을 생성한다. 이 단계가 수행되고 나면 Initiator와 Responder는 IKE_SA에 사용될 키 생성에 필요한 세션키 요소값을 공유하게 된다. 세션키 요소값을 통해 생성되는 키와 각각의 용도는 다음과 같다.

- SK_e : 이후에 전송되는 메시지들의 암호화에 사용.
- SK_a : 이후에 전송되는 메시지들의 인증(무결성 검증)에 사용.
- SK_d : CHILD_SA를 위한 다른 키 값 생성에 사용.

이 경우에 SK_a와 SK_e는 송수신에 각각 서로 다른 키 값이 사용된다.

따라서 IKE INIT이후에 전송되는 모든 메시지들은 SK_e와 SK_a에 의해 보호를 받는다. 즉, SK_e에 의해 헤더를 제외한 모든 페이로드가 암호화되며 SK_a를 통해 헤더를 포함한 모든 페이로드가 인증 함수에 입력되어 불법적인 도청이나 메시지 위·변조로부터 보호를 받게 된다.

그림 6과 7은 IKE AUTH Request와 Response 단계이다. IKE AUTH 단계는 상호 인증을 위해 식별정보와 인증 정보를 교환하고 (first) CHILD_SA 설정에 필요한 파라미터들을 협상한다. 먼저 Initiator는 자신의 식별 정보, 그에 대응하는 비밀 정보를 이용하여 Initiator임을 증명하기 위한 페이로드, AUTH 페이로드 검증에 사용될 공개키를 포함하고 있는 자신의 인증서를 포함하는 페이로드, 자신이 통신하고자 하는 Responder의 식별자 그리고 CHILD SA 협상에 필요한 페이로드인 SAi2, TSi, TSr을 송신한다. 그러면 Responder는 수신한 정보를 이용하여 Initiator를 인증하고 Responder



Initiator: **STEP3:IKE_AUTH Request**
Initiator: ----------
Initiator: HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr

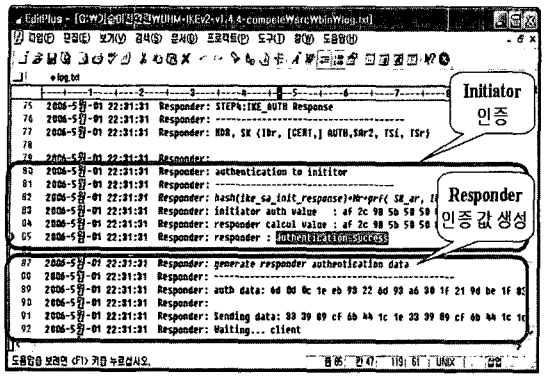
Initiator: generating... session key material
Initiator: ----------
Initiator: DH shared key : **98 2a a1 69 c8 b3 32 26**
Initiator: Pseudo-random Function: PRF_HMAC_SHA1
Initiator: IKEKeyMaterial:

Object: class uhm.protocol.security.crypto.IKEKeyMaterial
SK_d : ac 05 a6 22 f2 91 ea 48 3f b3 3f 60 05 35 2d 20 cf 21 52 b1
SK_ai : 7a ac 78 84 f6 86 73 ba c0 7a 6d 06 9b 70 2c 7c 18 32 34 49
SK_ar : 9c cc df 96 3a 20 37 5d ba f3 2a 83 30 df 7a c6 83 99 2e f3
SK_ei : b2 35 bb ca e1 b2 c9 6b b4 de 28 f9 db f7 4e 37 18 3f e6 5c
SK_er : 6a ac fa 7f 6f e9 ef 4f 6e 1d 63 10 52 5f 08 12 a8 aa 53 dc
SK_pi : bc db af 73 6a eb ce a0 90 a7 1a 12 6a d9 69 f2 f2 50 5f 5f
SK_pr : 01 0f 4a 26 23 77 bb a6 a7 f6 3b e6 c5 08 d4 09 bb 47 db 36

Initiator: **generate initiator authentication data**
Initiator: ----------
Initiator: PayloadAuthenticationData:
Object : class uhm.protocol.security.ikev2.payload.PayloadAuthenticationData
Hex value : af 2c 98 5b 50 50 8f b1 a2 0e 74 25 94 d7 18 f3 54 83 bf c5 16 51 7d d4 74 75 b4 01 0d b0 ec f8 02 8d 8b 02 4c 90 a5 cc 94 b3 77 d3 1e ce 5c 0d bd 1f b4 22 69 de c7 e9
Value : af 2c 98 5b 50 50 8f b1 a2 0e 74 25 94 d7 18 f3 54 83 bf c5 16 51 7d d4 74 75 b4 01 0d b0 ec f8 02 8d 8b 02 4c 90 a5 cc 94 b3 77 d3 1e ce 5c 0d bd 1f b4 22 69 de c7 e9

〔그림 6〕 IKE_AUTH Resquest

의 인증 값을 생성하여 IDr, AUTH 그리고 CHILDSA 협상에 필요한 SAr2, TSi, TSr를 Initiator에게 전송한다. 이 단계에서 Initiator와 Responder는 상대방을 인증하기 위한 AUTH 페이로드(Digital Signature or MAC)를 검증하고 ID 페이로드의 내용이 AUTH 페이로드의 생성에 사용된 키와 일치하는지를 확인해야 한다.



Responder: **STEP4:IKE_AUTH Response**
Responder: ----------
Responder: HDR, SK {IDr, [CERT,] AUTH,SAr2, TSi, TSr}

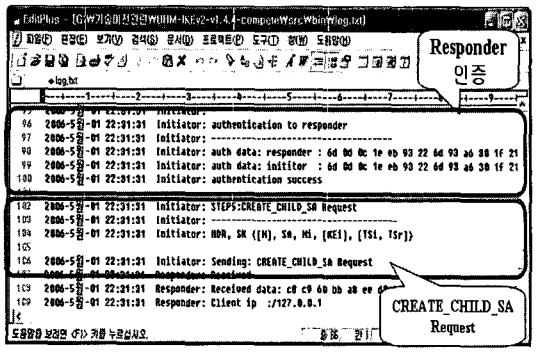
Responder: **authentication to inititor**
Responder: ----------
Responder: hash(ike_sa_init_response)+Nr+prf(SK_ar, IDr)
Responder: initiator auth value : af 2c 98 5b 50 50 8f b1 a2 0e 74 25 94 d7 18 f3 54 83 bf c5 16 51 7d d4 74 75 b4 01 0d b0 ec f8 02 8d 8b 02 4c 90 a5 cc 94 b3 77 d3 1e ce 5c 0d bd 1f b4 22 69 de c7 e9
Responder: responder calcul value : af 2c 98 5b 50 50 8f b1 a2 0e 74 25 94 d7 18 f3 54 83 bf c5 16 51 7d d4 74 75 b4 01 0d b0 ec f8 02 8d 8b 02 4c 90 a5 cc 94 b3 77 d3 1e ce 5c 0d bd 1f b4 22 69 de c7 e9
Responder: responder : **authentication success**

Responder: **generate responder authentication data**
Responder: ----------
Responder: auth data: 6d 0d 0c 1e eb 93 22 6d 93 a6 30 1f 21 9d be 1f 83 20 6f a1 d3 de 2b e1 eb 18 5a 3b 50 0c dd 36 dc fb 2f 01 9b f9 42 db aa 78 39 78 5e 8a 96 26 e1 17 59 aa 4b f9 f5 33

Responder: Sending data: 33 39 89 cf 6b 44 1c 1e 33 39 89 cf 6b 44 1c 1e 2e 20 23 04 33 39 89 cf 00 00 01 04 24 00 00 e8 1a 5d d1 a6 35 df bf 9f ea a3 39 0b 3d 12 19 72 51 f4 0f ef 94 21 cf 7e 73 a5 aa 34 c2 74 a9 e2 a1 d8 dd 51 be e8 3a 48 41 49 65 ea 89 b5 ce 58 3f 45 cd 14 3a c9 f5 7f 50 57 fd 77 5c fd 29 48 ca 0b 0d ba de e1 38 b9 91 46 16 c7 41 2d 66 8f 12 07 99 f9 71 ff cf 58 0a 16 05 cf 7b 13 24 3a df 1b 82 a8 2c c1 8c c0 3e 6c d7 9b f4 cb 19 25 f3 11 55 f5 ab e7 ce ad 56 07 2b 46 c2 64 cb 51 7b 89 7f b4 61 e1 78 ab 95 1e 59 58 17 d4 9c 3b da 2d c5 f9 cc 80 5a 90 35 5b 1d f5 93 23 56 e9 10 5e 69 32 c2 3b 29 00 1a 29 27 36 49 6c 06 9a c3 a4 4c 92 b9 12 8b ba 10 5e 69 32 c2 3b 29 00 1a 29 27 36 49 6c 06 9a 47 3a 7e 5c f0 7a 0b ca c0 3e 4c f1 b1 2a d3 57 f3 50 29 ef b4 c3 bf 4e 8c 90 43 b2
Responder: Waiting... client

〔그림 7〕 IKE_AUTH Response

그림 8과 9는 CREATE CHILD SA Request와 Response 단계이다. 이 단계는 IKEv1의 Phase 2에 해당하며 initial exchange가 완료된 후에는 Initiator와 Responder 누구나 CREATE CHILD SA의 initiator가 될 수 있다. 또한,



Initiator: authentication to responder
Initiator: ----------
Initiator: auth data: responder : 6d 0d 0c 1e eb 93 22 6d 93 a6 30 1f 21 9d be 1f 83 20 6f a1 d3 de 2b e1 eb 18 5a 3b 50 0c dd 36 dc fb 2f 01 9b f9 42 db aa 78 39 78 5e 8a 96 26 e1 17 59 aa 4b f9 f5 33
Initiator: auth data: inititor : 6d 0d 0c 1e eb 93 22 6d 93 a6 30 1f 21 9d be 1f 83 20 6f a1 d3 de 2b e1 eb 18 5a 3b 50 0c dd 36 dc fb 2f 01 9b f9 42 db aa 78 39 78 5e 8a 96 26 e1 17 59 aa 4b f9 f5 33
Initiator: **authentication success**

Initiator: **STEP5:CREATE_CHILD_SA Request**
Initiator: ----------
Initiator: HDR, SK {[N], SA, Ni, [KEi], [TSi, TSr]}

Initiator: Sending: **CREATE_CHILD_SA Request**
Responder: Received
Responder: Received data: c8 c9 60 bb a8 ee d0 98 c8 c9 60 bb a8 ee d0 98 2e 20 24 10 c8 c9 60 bb 00 00 00 9c 21 00 00 80 d3 9f 0e fb a3 ba 3e da b2 b3 fe 56 04 dc 29 b2 10 dc b5 c8 b5 ad 48 18 f9 dd 83 68 19 10 dd bf 17 81 7c 1b 82 8e a4 fc c7 b6 07 7d 85 52 88 ab 9d ec e9 5f bc d5 b3 f7 0b 1a c2 00 d0 2b 3d 60 ac 9b 56 3d 66 a1 5c 0c d0 18 1b 96 46 13 40 21 94 99 73 80 42 63 e1 57 71 50 ce 3a 7d 97 89 d2 9c ae dc cb e5 34 e7 df 8f 38 c5 b6 4a 1f 73 8f 33 3f 8d ee 79 8e b2 ca b6 5b 4c 22
Responder: Client ip :/127.0.0.1

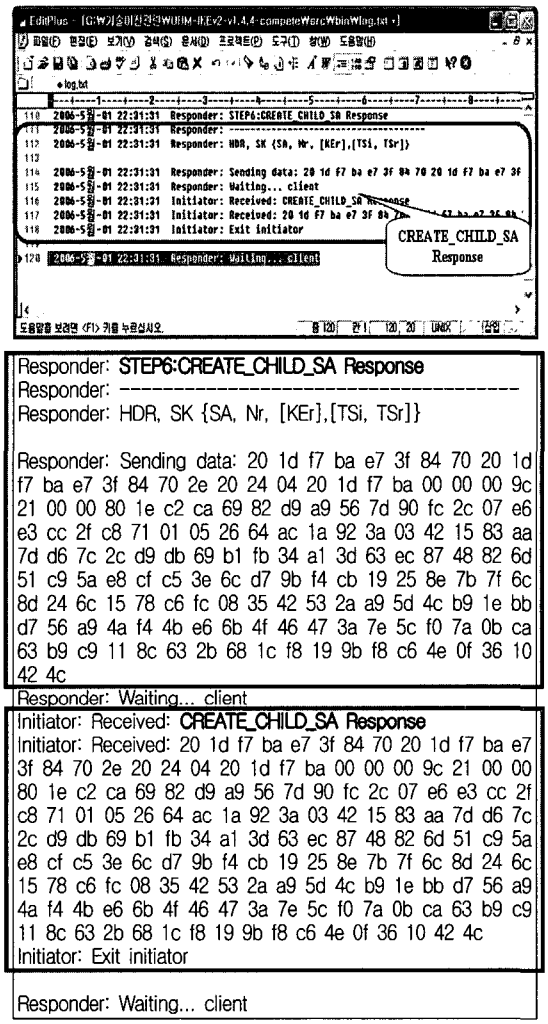
〔그림 8〕 CREATE_CHILD_SA Request

IKE_SA_INIT에서 설정된 키(SK_e, SK_a)에 의해 보호를 받는다.

CREATE CHILD SA는 선택적으로 KE 페이로드를 통해 추가적인 Diffie-Hellman 값을 교환할 수 있으며, 이때 CHILD_SA에 대한 PFS(Perfect Forward Secrecy)가 보장된다.

CHILD SA에 대한 키 값은 SK_d와 nonces(Ni, Nr), 그리고 KEi와 KEr에 의해 설정된 Diffie-Hellman 값의 함수로 표시된다.

만일, CHILD SA가 initial exchange에서 생성되었다면 KE 페이로드와 nonce 페이로드는 전송되어서는 안 되며 필요한 nonce값은 phase 1의 nonce 값을 그대로 사용한다.



〔그림 9〕 CREATE_CHILD_SA Response

V. 결 론

본 논문에서는 IKEv2 키 교환 프로토콜에 대한 설계와 구현을 하였다. 내용으로 IKEv2 구현을 위한 구성요소에 대한 요구사항을 바탕으로 실제적인 구현방안을 제시하였고, 구현된 IKEv2는 다른 시스템에 이식하는데 용이하도록 자바 모듈로 설계되었다. IKEv2는 IPSec에서 사용할 SA를 자동으로 설정하고 상호인증 및 키 교환이 가능하며 서비스 거부 공격에 방지 할 수 있는 특징을 가진다. IKEv2는 IKEv1에서 여러 문서로 되어있는 문서가 하나로 통합되고 8개의 메시지는 4개로 축소되었으며, 서비스 거부 공격에 방지할 수 있는 기능이 추가 되었다. 추후에는 구현된 IKEv2 프로토콜을 IPSec으로 확장하고, SADB에 관련된 부분에 대한 보안을 추가하는 것을 목표로 한다.

참 고 문 헌

〔1〕 S. Kent "Security Architecture for the Internet Protocol", RFC 2401, November, 1998.

〔2〕 S. Kent and R. Atkinson, "IP Authentication Header,", RFC 2402, November 1998.

〔3〕 S. Kent and R. Atkinson, "IP Encapsulating Security Payload(ESP),", RFC-2406 November 1998.

〔4〕 D. Paper, "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November, 1998.

〔5〕 D. Maughan, "Internet Security Association and Key Management Protocol", RFC 2408, November, 1998.

〔6〕 D. Harking and D. Carrel, "The Internet Key Exchange", RFC 2409, November 1998.

〔7〕 Charlie Kaufman "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17.txt. September2004.

〔8〕 H. Orman "The Oakley Key Determination Protocol", RFC 2412, November 1998.

〔9〕 Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.

〔10〕 Diffie, W., and Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22, n. 6, June 1977.

〔11〕 R. Housley, W. Polk, W. Ford, and D. Solo. "Internet X.509 public key infrastructure certificate and certicate revocation list (CRL) profile." RFC 3280, Internet Engineering Task Force, April 2002.

〈著 者 紹 介〉

엄 희 정 (Hee-Jung Eom)
학생회원

2000년 2월 : 수원대학교 전자계산학과 졸업
2004년 2월 : 순천향대학교 정보보호학과 석사 졸업
2004년 9월~현재 : 순천향대학교 정보보호학과 박사과정
〈관심분야〉 암호 프로토콜, 통신공학, 정보보호

김 락 현 (Rack-Hyun Kim)
학생회원

1997년 2월 : 순천향대학교 전자공학과 졸업
1999년 8월 : 순천향대학교 일반대학원 전기·전자공학과 석사 졸업
2000년~현재 : 순천향대학교, 청운대학교, 홍성기능대학교, 아산정보기능대학 외래강사
2001년 2월~현재 : 순천향대학교 일반대학원 정보보호학과 박사과정
〈관심분야〉 암호 이론, 공개키 기반구조, 네트워크 보안, 보안 프로토콜, 이동통신보안

염 흥 열 (Heung Youl Youm)
종신회원

1981년 2월 : 한양대학교 전자공학과 졸업(학사)
1983년 2월 : 한양대학교 대학원 전자공학과 졸업(석사)
1990년 2월 : 한양대학교 대학원 전자공학과 졸업(박사)
1982년 12월~1990년 9월 : 한국전자통신연구소 선임연구원
1990년 9월~현재 : 순천향대학교 공과대학 정보보호학과 교수
1997년 3월~2000년 3월 : 순천향대학교 산업기술연구소 소장
2000년 4월~2006년 2월 학교 산학연컨소시엄센터 소장
1997년 3월~현재 : 한국정보보호학회 총무이사, 학술이사, 교육이사, 현 총무이사
2004년 1월~현재 : 한국인터넷정보학회 이사, 논문지 편집위원
2004년 1월~현재 : OSIA 이사
2003년 9월~2004년 3월 : ITU-T SG17/Q10, Associate Rapporteur
2004년 3월~현재 : ITU-T SG17/Q9 Rapporteur
〈관심분야〉 네트워크보안, 전자상거래보안, 공개키 기반구조, 부호이론, 이동통신보안