IPsec 시스템의 효율성을 위한 SPDB와 SADB의 연계방안에 관한 연구

이형규*, 나재훈*, 남택용*, 손승원*
*한국전자통신연구원

Study on the Interaction of SPDB and SADB for IPsec System

Hyung-kyu lee, jae-hoon na*, taek-yong nam*, seung-won shon*,*
**ETRI*

요 약

본 논문에서는 우리가 개발한 IPsec 시스템의 효율성을 위해 인터넷 키교환 서버를 중심으로 모듈간 연동 구조와 이 에 따른 수행 절차를 다룬다. 개발한 IPsec 시스템은 IP기반 단대단 보안을 위해 IPsec 엔진을 중심으로 키관리, SADB, SPDB 모듈이 통합된 구조로 되어 있다. 따라서 각 모듈간의 효율적인 연동구조는 시스템의 전체 효율에 매우 큰 영향을 줄 수 있다. 특히, 우리의 IPsec 시스템에서 IPsec 엔진은 커널과의 통합방식으로 구현되었기 때문에 SPDB와 SADB의 구현 위치에 따른 조회의 효율성을 위해 여러 고려 사항들이 필요하다. 우리는 이러한 문제를 풀기위해 IKE 서버에 의해 생성된 SPI를 사용한다. 최종적으로 우리는 SPDB 엔트리와 SADB 엔트리 조회의 최적화 방법에 기인한 모듈간 연동구조를 제안한다.

I. 서론

IETF 보안 분야에서 IPsec WG는 인터넷 정보보호에 관한 기본 구조를 연구하고 있는 그룹으로서, AH(Authentication Header)와 ESP(Encapsulation Security Payload)의 두가지 확장헤더와 IKE(Internet Key Exchange)를 정의하였다.[1,2,3,4,5]. 현재 AH, ESP와 IKE는 대부분의 플랫폼에서 프로토타입으로 구현되었으나 아직 완전하다고는 볼 수 없는 형편이다.

IPsec의 특징은 다음 몇 가지로 요약될 수 있다. 정보보호 서비스가 IP 계층에서 제공됨으로서 기존의 응용 소프트웨어에 대한 변경을 요하지 않아, 일반 인터넷 사용자에게는 투명한 상태로 처리된다. 응용계층 및 트랜스포트 계층의 모든 프로토콜에 공통된 정보보호 서비스를 제공할 수 있기 때문에, 한 호스트 내에서는 일관된 방식의 정보보호 서비스 설정이 가능하다. 현재 IPsec이 가장 활발하게 적용되고 있는 VPN(Virtual Private Network) 산업분야에서는, IPsec을 엔터프라이즈 네트워크에 적용시 확장성 및 호환성을 해결할 수 있는 유일한 정보보호 프로토콜로 여기고 있다. 이 에 본 논문에서는 IPsec의 효율적인 서비스를 위해 IKE(Internet Key Exchange : 인터넷 키교환)

서버를 중심으로 IPsec엔진과 SPDB(보안정책 데이터 베이스) 및 SADB(보안연계 데이터베이스)의 연동구조 및 절차에 대해 논의한다. 특히, IPsec 엔진과 IKE가 각각 시스템내의 서로 다른 계층에 존재하므로 SPDB와 SADB의 계층별 위치 선정과 각 엔트리들의 연계성은 시스템의 효율적인 동작을 위해 매우 중요한 역할을 수행한다. 이 에 우리는 효율성 및 SPDB와 SADB 연계성을 위해 필요한 파라미터 및 각 모듈간 수행절차를 제시한다.

II. 본론

1. *IPsec* 구성요소

이 절에서는 대부분의 IPsec 구현이 가지는 구성요소를 중심으로 IPsec 서비스를 위한 각 기능을 설명한다.[1,9].

□ **IPsec 기본 프로토콜** : 본 구성요소는 ESP와 AH에 대한 구현이다. IPsec 기본 프로토콜은 패킷에 인증이나 기밀성과 같은 보안기능을 주기 위해 SPDB 및 SADB와 통신하여 헤더를 처리하며, 프레그멘테이션이나 PMTU와 같은 네트워크 계층의 이슈들을 조정한다.

□ **SPDB(Security Policy Database : 보안정책데이터베이스)** : SPDB는 패킷에 줄 수 있는 보안을 결정하는 중요한 구성요소이다. SPDB는 패킷이 외부로 나갈 때(Outbound packet)나 내부로 들어올 때(Inbound packet)에 참조된다. 즉, IPsec 기본 프로토콜은 전송패킷 (Outbound packet)에 대해 패킷에 어떤 보안정책을 적용할지를 결정하기 위해 SPDB(보안정책데이터베이스)를 참조하고 수신패킷(Inbound packet)에 대해서는 패킷에 적용된 보안처리가 해당정책에 설정된 보안엔트리와 일치하는지를 결정하기 위해 참조한다.

□ **SADB** (Security Association Database : 보안연계데이터베이스) : SADB는 송신패킷과 수신패킷을 처리하기 위해 사용중인 SA(보안연계)의 목록을 유지, 관리한다. SA(보안연계)는 수동적으로 또는 IKE 서버를 경유해서 SADB에 저장하게 되며 암호 알고리즘, 키의 수명, 해쉬함수 등 보안처리를 위해 요구되는 일련의 정보이다.

□ **IKE(Internet Key Exchange : 인터넷키교환)** : 인터넷 키교환은 사용자 수준의 프로세스이다. 이것은 키교환과 SA 협상을 위한 프로토콜이다.[4,5,6] 우리의 IPsec 시스템에서 IKE의 구현은 IKEB이다. IKEB는 SPCB나 다른 IKEB의 협상요구에 의해 동작하게 된다.

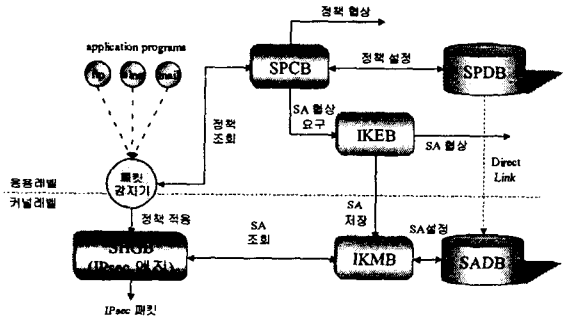
□ **SPDB와 SADB 관리 모듈** : SPDB와 SADB를 관리하기 위한 응용들로써 각각 SPCB와 IKMB가 해당 역할을 수행하고 있다.

2. *IPsec* 구성요소간 연동 구조

본 논문에서 제안하는 IPsec 시스템은 사용자 투명성과 인터넷에 범용적으로 적용될 수 있는 자동화된 구조로 설계되었다. 즉, IPsec 시스템은 사전에 설정된 정책과 SA를 요구하지 않는다. 즉, 애플리케이션의 동작이 이루어지면 각 모듈이 연동하여 IPsec을 적용하는 구조로 이루어진다. 참고로 오픈 소스로 제공되고 있는 freeswan의 경우, IPsec 통신을 위해서는 통신을 위한 임의의 모든 노드들에 대한 정책이 존재하고, 해당정책에 따른 SA협상이 수행된 후에야 가능하다.[11].

일반적으로 IPsec을 위한 정책은 패킷에 대해 IPsec을 적용할지, 적용하지 않을지, 폐기할지에 대한 결정을 수행한다. 따라서, 패킷에 대해 IPsec이 적용되어야 한다면 패킷 감지기가 SHGB로의 패킷을 감시하면서 SPCB를 통해 통신하려는 패킷에 대한 SPDB를 조회한 후 정책을 결정하고 만약 정책이 존재하지 않으면 해당 정책을 다른 도메인에 있는 SPCB와 협상하게 된다.[7,8]. 정책간의 협

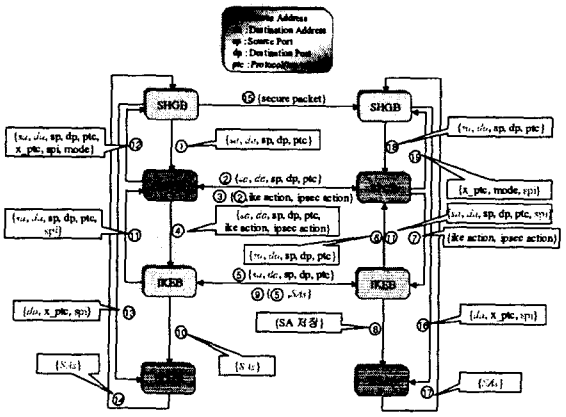
상은 IETF IPSP 작업그룹에서 현재 표준화를 위해 연구되고 있는 주제로써 정책 협상을 위한 프로토콜 등에 대한 드래프트 버전들이 나와 있다. 제안하는 IPsec 시스템은 아직 정책 협상 기능을 지원하지는 않지만 그러한 기능도 염두에 두고 정책이 설정할 수 있도록 설계하였다. 정책이 협상되면 해당 정책에 따른 SA협상을 IKEB에게 요구하며 IKEB가 협상한 SA는 IKMB를 통해 SADB에 저장한다. 만약, 정책이 존재하는 경우에는 협상된 SA가 이미 존재하기 때문에 SHGB가 SADB를 조회할 수 있도록 송신지 주소와 수신지 주소 및 SPI를 SHGB에게 전달하게 된다. 아래의 그림 1은 앞에서 설명한 IPsec 시스템의 연동구조를 개념적으로 설명하고 있다.



[그림 1] IPsec 시스템의 각 구성요소간 연동구조

IKEB는 IPsec SA(Security Association)와 공유키의 생성을 위한 키교환 프로토콜의 구현이다. IPsec 시스템에서 IKEB는 IPsec 시스템 내부에 설치되어 키교환 기능을 수행하게 된다. 이 때 원활한 키교환과 효율적인 IPsec구현을 위해 다른 기능블록들과의 연동구조를 가진다.

IKE는 우리가 이미 잘알고 있는 포트(well-known port 500)와 UDP 프로토콜을 사용하여 모든 구현들이 IKE 패킷을 인식하도록 해야 한다.[2]. 그렇지 않으면 보안을 요구하는 어떤 패킷도 IPsec시스템을 나갈 수 없다. 이것은 IPsec 프로토콜 엔진이 패킷에 대해 보안처리(Apply), 폐기(Discard) 및 통과(Bypass) 정책을 적용하기 때문이다.[1,9,10]. IPsec 시스템이 어떻게 보안정책을 적용하여 IPsec 패킷으로 다른 시스템과 통신하는가에 대한 수행 절차와 주요 전송 파라미터는 아래 [그림 2]와 같다.



[그림 2] 연동 절차에 따른 송수신 파라미터

사전에 설정된 정책과 SA가 없다는 가정하에 프로토콜의 송신측과 수신측이 어떻게 IPsec 통신을 하는가를 나타내고 있다. 물론 telnet이나 ftp와 같은 애플리케이션의 동작에 의해 아래의 과정이 시작된다. 특히, IPsec 패킷의 처리를 위한 SHGB와 SPCB의 연동과정과 SA협상을 위한 SPCB와 IKEB의 연동과정은 프로토콜 시작자와 응답자에 따라 차이가 있음을 알 수 있다. 위의 [그림 2]를 보다 자세히 설명하면 SHGB가 패킷을 내보내기 위해서 먼저, 패킷에 보안 처리를 적용해야 할지에 대해 SPCB에게 문의하게 된다. 이 때 사전에 설정된 정책이 없다고 가정하면, SPCB는 정책을 설정하기 위해 다른 도메인의 SPCB와 정책을 협상하고 이 때 해당 패킷에 IPsec이 적용되어야 한다면 IKEB를 통해 SA를 협상하도록 한다. 협상된 SA는 IKMB를 통해 SADB에 저장하게 되고 SHGB가 이 SADB를 조회할 수 있도록 하기 위해 IKEB는 SPI값을 SPCB를 통해 SHGB에게 전해준

다. 이렇게 되면 IPsec을 적용하기 위해 SHGB는 SPI와 sa, da 등의 값으로 IKMB에게 해당 SA에 대한 조회 요구를 보내고 IKMB는 해당 SA값을 SADB로부터 읽어서 SHGB에게 리턴하게 된다. 이 때 SPDB나 SADB를 조회하기 위해 사용되는 값들을 식별자(selector)로 부른다. 또한 IPsec 패킷은 송신측과 수신측에서의 처리방법이 서로 다르다.[2]. 일반적으로 IPsec의 송신측에서 SA를 조회하는 방법은 패킷의 식별자를 사용하여 SPDB를 조회하고 SPDB가 포인팅하는 SADB의 엔트리들을 송신 패킷에 적용하는 것이다. 반면, 수신측에서는 SPI, 목적지 IP주소 및 프로토콜을 사용하여 SADB를 먼저 조회한다. 여기에서 프로토콜은 패킷에 적용된 보안처리가 AH인가 혹은 ESP인가를 가리키는 것이며 SPI는 랜덤수로써 SA를 유일하게 식별할 수 있게 한다. 이러한 SPI의 생성은 IKE 협상시 이루어지고 SA 데이터베이스에 저장되었다가 패킷에 실려 통신 상대방으로 전달된다. 수신 패킷에 대한 IPsec처리가 수행되고 나면 후에 SPDB의 엔트리들과 SADB 엔트리가 서로 매칭하는지에 대한 검사를 수행하게 된다. 한편, 우리의 IPsec 시스템의 구현은 커널에 통합되는 방식으로 구현되었기 때문에 응용레벨에 존재하는 SPDB가 커널에 존재하는 SADB를 포인팅하기 위한 논리적인 구조를 요구한다고 하였다. 이것을 위해 우리는 SPI(Security Parameter Index : 보안파라미터지수)를 사용하며 결과적으로 송/수신지 모두 송신지 및 목적지 IP주소와 SPI를 사용하여 SADB의 엔트리들이 조회된다. 좀 더 자세히 설명하면 IKEB에 의해 생성된 SPI는 SPCB를 통해 SPDB에 저장되고 이것을 받은 SHGB는 송신지 및 수신지 IP주소와 함께 해당 SADB를 조회하는데 사용한다. 이러한 구조는 SA 조회에 대한 최적화를 위한 것으로 IPsec 엔진이 보안처리를 위해 SPDB를 조회할 때, 조회된 정책에 유일하게 매핑되는 SA를 가리키게 하기 위한 구조를 제공하기 위해서이다.

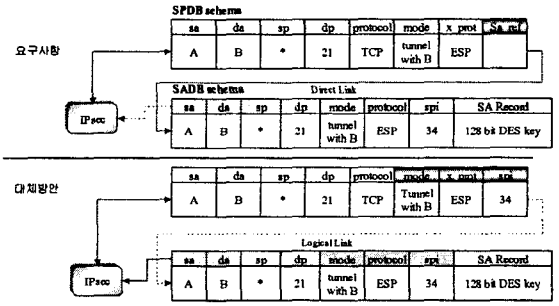
3. 효율적인 *SPDB와 SADB의 연계방안*

앞 절에서 설명한 연동구조를 지원하고 효율적인 SA 조회 경로를 제공하기 위해 SPDB와 SADB의 설계는 매우 중요하다. 다음은 SPDB와 SADB의 설계를 결정하는 중요 요인들이다.[1,9].

- SPDB와 SADB에서 예상 엔트리의 수
- 요구되는 할당 메모리의 비용 대 큰 테이블의 유지 비용 및 사용되지 않는 메모리의 비율
- SADB 또는 SPDB 엔트리로의 포인터를 캐쉬하기 위해 시스템이 제공하는 어떤 유형의 최적화

상기 요인 중 세 번째 요인이 IPsec 엔진이 IPsec정책을 조회하고 IPsec 패킷을 생성하기 위한 효율적인 구조에 매우 중요한 역할을 제공한다. 따라서, 우리가 개발한 커널 통합 방식의 경우 SADB의 조회 효율성을 위해 SADB가 커널에 존재하므로 응용레벨에 있는 SPDB와 커널에 있는 SADB의 매칭을 위한 어떠한 연동 절차가 필요하게 된다.

아래 [그림 3]은 상기한 SPDB와 SADB의 연계방법을 나타내고 있다.



[그림 3] SPDB와 SADB의 효율적인 연계방안

[그림 3]은 IPsec 엔진이 패킷 식별자를 사용하

여 SPDB 엔트리를 조회하고 이것이 곧 SADB 엔트리의 조회로 이어지도록 Sa_ref라는 SADB로의 포인터를 사용하도록 하고 있다. 또한 SPDB와 SADB의 매칭을 위한 구현은 SPDB와 SADB의 엔트리에 대한 생성과 폐기 및 갱신시에 따른 동기화도 필수적으로 고려되어야 한다. 한편, SPDB와 SADB의 요구사항과 동등한 효과를 위한 논리적 최적화 방안을 위해 우리는 다음 [그림 3]의 하단부에 있는 대체방안을 제안한다. 왜냐하면 우리의 IPsec 시스템에서 SPDB와 SADB의 구현 위치가 각각 응용과 커널이기 때문에 포인터를 캐쉬할 수 있는 일반적인 방법이 없기 때문이다. 따라서, [그림 3]에서 나타난 대체 방안은 SPDB와 SADB를 각각 조회하지만 논리적 링크와 동기를 유지할 수 있도록 하기 위해 SPI를 사용하고 있다. 이와 같은 설계에 의해 우리가 제안하는 IPsec 시스템은 효율적으로 동작할 수 있다.

III. 결론

본 논문에서는 효율적인 IPsec 구현을 위해 각 구성요소에 대한 요구사항을 바탕으로 실제적인 구현 방안을 제시하였다. 특히, 실제적인 보안서비스를 위하여 보안 파라미터들의 조회경로를 최적화 하도록 IPsec 시스템의 각 구성요소간 연동구조를 제안하고 있다. 이러한 구조는 특히 SPDB와 SADB의 최적화 연계방안과 관련하여 시스템의 효율성을 높이도록 설계되었다.

참고문헌

[1] S. Kent and R. Atkinson "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. Harkins and D. Carrel, The Internet Key Exchange (IKE) , RFC 2409, November 1998.

[5] D. Piper, "The Internet IP Security Domain of Interpretation for ISAKMP," *RFC 2407, November 1998.*

[6] D. Maughan, M. Schertler, M. Schneider and J. Turner "Internet Security Association and Key Management Protocol," *RFC 2408, November 1998.*

[7] Jamie Jason, Lee Rafalow and Eric Vyncke "IPsec Configuration Policy Model" *draft-ietf-ipsp-config-policy-model-05.txt*, February 2002.

[8] Man Li, Avri Doria, Jamie Jason, Cliff Wang and Markus Stenberg "IPsec Policy Information Base" *draft-ietf-ipsp-ipsecpib-04.txt*, February 2002.

[9] Naganand Doraswamy and Dan Harkins, "IPsec The New Security Standard for the Internet, Intranets, and Virtual Private Networks," *Prentice Hall PTR, Upper Saddle River, NJ 07458.*

[10] Dave Kosiur, "Building and Managing Virtual Private Networks," *Wiley Computer Publishing*, Published by John Wiley & Sons, Inc.

[11] http://www.freeswan.org