**IP Security**

1. **개 요**

IPSec( Internet Protocol Security )는 통신 세션의 각 IP패킷을 암호화하고 인증하는 안전한 인터넷 프로토콜( IP ) 통신을 위한 프로토콜 스위트이다. 이 보안은 통신 세션의 개별 IP 패킷을 인증하고 암호화함으로써 처리된다. IPsec은 세션의 시작에서 에이전트들 사이에서 상호 인증을 확립하거나 세션을 맺는 중에 사용될 암호화 키의 협상을 위한 프로토콜을 포함한다.

인터넷 프로토콜의 약점은 전송되는 데이터가 암호화되지 않아 데이터그램의 인증과 프라이버시 보호 기능을 제공하는 범용적인 방법이 없다는 점이다. IP 데이터그램은 알려지지 않은 네트워크 사이에서 두 장비간에 라우팅이 되어야 하기 때문에, 그 안에 있는 정보를 가로채고 수정하는 공격에 취약하다. 이러한 보안 문제는 25년 전 인터넷이 작고 상대적으로 “private”한 환경이었기 때문이다. 인터넷에 몇 대의 컴퓨터만 있을 때는 문제가 없었고, 인터넷이 성장할 때에도 연구원 및 기타 네트워킹 전문가를 연결하는 데만 사용되었다. 성장하는 인터넷 네트워크에 추가된 새 사이트의 신원은 적어도 누군가 알고 있었다는 얘기다. 하지만 점차 인터넷은 거대하고 모두에게 “public”한 환경이되면서 전체적인 네트워크의 보안 유지가 불가능하게 되었다.

이러한 IP의 단점을 보완하기 위해 개발된 것이 IPSec( IP Security )라는 프로토콜 모음이다. 즉, IPSec은 하나의 프로토콜이 아니라 IP 네트워크를 위한 보안 솔루션을 제공하는 서비스와 프로토콜의 모음이라는 것이다. IPSec은 원래 IPv4의 주소 고갈 문제를 해결하기 위해 개발된 IPv6의 IP 단점인 보안문제를 추가하기 위해 개발되었고, 이것이 현재의 IPv4에도 적용되었다. 참고로 IPv4에서는 보안이 필요한 경우에만 선택적으로 사용하지만, IPv6에서는 기본 스펙에 포함된다. 또한 네트워크 계층에서 동작하기 때문에 자체 보안기능이 없는 상위 계층 TCP/IP Application이나 프로토콜도 보호할 수 있다. IPSec이 실제로 적용된 시기는 얼마 안되었고, 주로 VPN을 구현하는데 사용한다.

1. **역 사**

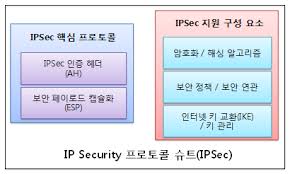
NSA는 1896년부터 1991년까지 SNDS( Secure Data Network Systems ) 프로그램에 따라 인터넷 보안 프로토콜 개발을 후원했다. 1988년에 네트워크 암호화 장치를 생산한 모토로라를 비롯한 여러 공급 업체가 모였습니다. NIST에서 1988년에 공개적으로 출간했으며 SP3( Security Protocol on Layer 3,네트워크 계층에 대한 보안 프로토콜을 의미한다. )는 결국 ISO 표준 네트워크 레이어 보안 프로토콜로 변경될 것입니다.

1992년부터 1995년까지 다양한 연구 그룹이 SDNS의 SP3를 개선했다. 1992년 미국 해군 연구소( NRL )는 SIP 암호화 알고리즘을 연구하고 구현하는 SIPP 프로젝트를 시작했다. 1993년 12월, 실험적인 소프트트웨어 IP 암호화 프로토콜( swIPe )가 콜럼비아 대학교에서 SunOS와 AT&T 벨 연구소의 John loannidis에서 개발되었다. 1993년 백악관에서 자금을 조달받은 신뢰할 수 있는 정보 시스템( TIS )의 Wei Xu는 swipe 연구를 수행하였고, IP 보안 프로토콜을 강화했으며 데이터 암호화 표준에 대한 디바이스 드라이버를 개발하였다. 1994년 12월까지 3DES 하드웨어 암호화가 통합된 TIS Gauntlet Firewall 제품은 T1 이상의 속도로 상용 IP 보안을 제공하며 미국 동부와 서부 해안 사이의 네트워크를 안전하게 유지했다. 이 기간동안 IETF( Internet Engineering Task Force ) IP Security Working Group은 IPsec이라고 하는 개방적이고 자유롭게 사용할 수 있는 보안 확장 집합으로 이러한 노력을 표준화했다. 1995년에는, 워킹그룹은 NRL이 RFC-1827을 통해 RFC-1825를 출판하면서 첫 번째 구현을 수행했다.

* IPSec 표준

|  |  |  |
| --- | --- | --- |
| RFC  번호 | 이름 | 설명 |
| 2401 | Security Architecture for the Internet Protocol | 주 IPsec 문서로 IPsec 기술의 구조와 일반 동작, 여러 구성 요소가 어떻게 쓰이는지를 보여준다. |
| 2402 | IP Authentication Header | 데이터 무결성과 원본 검증을 보장하는 데 쓰이는 IPsec 인증 세더(AH) 프로토콜을 정의한다. |
| 2403 | The Use of HMAC-MD5-96 within ESP and AH | AH와 보안 페이로드 캡슐화(ESP) 프로토콜에서 사용하는 암호화 알고리즘인 Message Digest 5(MD5), HMAC 변종을 설명한다. |
| 2404 | The Use of HMAC-SHA-1-96 within ESP and AH | AH와 ESP에서 사용하는 암호화 알고리즘인 Secure Hash Algorithm 1(SHA-1), HMAC 변종을 설명한다. |
| 2406 | IP Encapsulating Security Payload(ESP) | 기밀성을 위해 데이터를 암호화하는 IPsec ESP 프로토콜을 설명한다. |
| 2408 | Internet Security Association and Key Management Protocol(ISAKMP) | 키를 교환하고 보안 연관을 협상하는 방법을 정의한다. |
| 2409 | The Internet Key Exchange(IKE) | 두 장비간 안전한 통신을 위해 보안 연관을 협상하고 키를 교환하는 데 쓰이는 IKE 프로토콜을 설명한다. IKE는 ISAKMP와 OAKLEY에 기반을 두고 있다. |
| 2412 | The OAKLEY Key Determination Protocol | 키 교환을 위한 범용 프로토콜을 설명한다. |

1. **IPSec의 일반 동작, 구성요소, 프로토콜**



* 두 장비간 안전한 통신을 하기 위한 수행 작업

상대방이 이해할 수 있는 형태로 데이터를 보내는데 사용할 보안 프로토콜에 동의해야 하며, 데이터를 인코딩할 때 사용할 구체적인 암호화 알고리즘을 결정해야 한다. 또한 암호로 인코딩된 데이터를 푸는 데 쓰이는 키를 교환해야한다. 이러한 배경 작업을 오나료하면, 각 장비는 협상한 프로토콜, 방법, 키를 이용하여 데이터를 인코딩하고 네트워크로 송신할 수 있다.

* IPSec 핵심 프로토콜

핵심 프로토콜에는 IPSec 인증 헤더( Authentication Header )와 보안 페이로드 캡슐화( Encapsulating Security Payload )가 있다.

AH는 IPSec을 위한 인증 서버를 제공하며, 중간에 데이터를 가로채 변경하지 않았음을 검증하는 무결성을 확인한다. 또한 공격자가 메시지를 캡처한 뒤 재전송하는 공격인 Replay Attack에 대한 보호 기능을 제공한다.

AH는 데이터의 무결성을 보장하나 프라이버시를 보장하지는 않는다. 이것이 ESP가 하는 역할이며 ESP를 이용하여 데이터를 암호화하여 원하는 사람만 데이터를 볼 수 있게 해준다.

* IPSec 지원 구성 요소

IPSec은 AH와 ESP는 암호화 방법을 지정하지 않아 필요에 따라 다양한 암호화 알고리즘을 사용할 수 있는 유연성을 제공한다. 일반적으로 널리 쓰이는 알고리즘은 MD5와 SHA-1이다. 이러한 암호화/해싱 알고리즘들은 입력 데이터와 키를 이용하여 해시( Hash )라는 값을 계산하는 방법으로 동작하여 해싱 알고리즘이라 불린다.

IPSec을 사용하는 장비간에는 어떤 방법 및 정책으로 구현할 것인지 정해야한다. 이 때 IPSec은 정해진 정보를 서로 교환하는 방법을 제공한다.

암호화된 정보를 확인하기 위해서는 키가 필요하고, 이를 서로 공유하기 위해 인터넷 키 교환( Internet Key Exchange )를 제공한다.

1. **IPSec 인증 헤더( AH )**

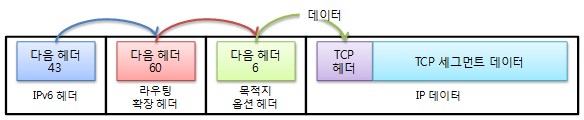
* 개요

AH( Authentication Header )는 데이터그램의 값에 근거하여 계산되는 헤더를 추가하여 데이터그램 컨텐츠의 전체 또는 일부분에 대한 인증을 제공한다. 계산에 쓰이는 데이터그램의 부분과 헤더의 위치는 IPSec 모드와 IP Version에 따라 달라진다. AH의 에러 탐지는 메시지 컨텐츠에 기반한 Checksum이나 CRC코드를 계산한다. 그리고 특수 해싱 알고리즘( HMAC-SHA1… )과 오직 출발지와 목적지 장비에게만 알려진 특수 키를 사용한다. 출발지에서 무결성 검사값( ICV, Integrity Check Value )를 다른 필드와 함께 특수 필드에 넣어 전송하고, 목적지에선 공유된 키를 사용하여 동일한 계산을 수행한다. 따라서 목적지 장비는 원본 데이터그램의 무결성을 확인한다. AH는 메시지의 무결성을 제공하기는 하지만 암호화는 하지 않는다. 암호화는 ESP에서 담당한다.

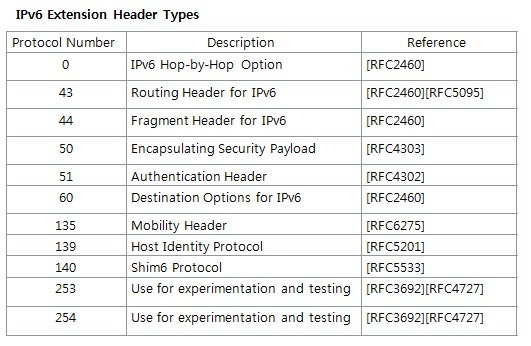
AH는 다음 수준의 프로토콜 데이터는 물론 가능한 많은 IP 헤더에 대한 인증을 제공한다. 그러나 일부 IP 헤더 필드는 전송 중에 변경될 수 있으며 패킷이 수신자에게 도착할 때 이 필드 값은 송신자가 예측할 수 없다. 이러한 필드의 값은 AH로 보호할 수 없으므로 AH가 IP 헤더에 제공하는 보호는 단편적이다. AH는 ESP와 결합하여 단독으로 적용하거나 중첩된 방식으로 적용할 수 있다. 보안 서비스는 통신 호스트 쌍 사이, 통신 보안 게이트웨이 쌍 사이, 통신 보안 게이트웨이 쌍 사이 또는 보안 게이트웨이와 호스트 사이에 제공될 수 있습니다. ESP는 동일한 재생 방지 및 유사한 무결성 서비스를 제공하는데 사용될 수 있으며 기밀 (암호화) 서비스도 제공한다. ESP와 AH가 제공하는 무결성의 가장 큰 차이점은 보험 적용 범위이다. 특히 ESP는 ESP로 해당 필드를 캡슐화하지 않는한 IP 헤더 필드를 보호하지 않는다.

* AH 데이터그램 위치와 연결

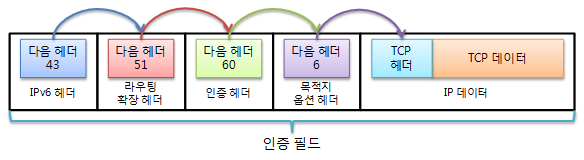
AH 데이터그램 위치와 연결은 IPv6과 IPv4로 나뉠 수 있다. IPv6에서 AH 위치와 연결은 다음 그림과 같다.



원본 IPv6 데이터그램 포맷 ( 라우팅 확장 헤더와 목적지별로 고유한 목적지 옵션 확장 헤더 포함 )

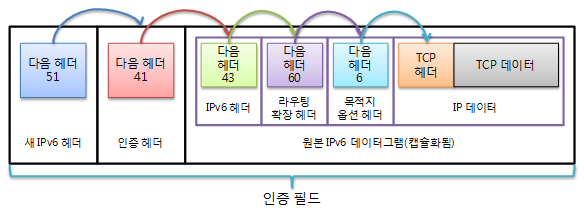


참고로 그림에서 의미하는 다음 헤더에서 지정할 수 있는 프로토콜 넘버는 위 표와 같다. 즉 위의 IPv6 헤더는 기본 헤더에 2개의 확장 헤더가 붙은 형태라고 이해하고 넘어가면 되겠다. ( 비고 : 다음 헤더가 6일 경우 이것은 TCP 헤더를 의미하는 번호이다. )



IPSec 전송 모드 – IPv6 AH 데이터그램 포맷

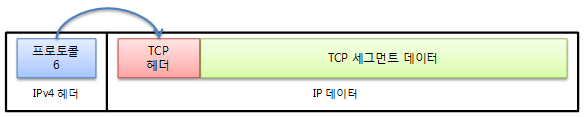
전송 모드에서 AH는 단순히 라우팅 확장 헤더와 목적지 옵션 헤더 사이에 위치하는 새로운 확장 헤더로 추가된다. 다음 헤더 필드는 각 헤더를 다음 헤더와 연결하는 데 사용된다.



IPSec 터널 모드 – IPv6 AH 데이터그램 포맷

터널 모드에서 원본 데이터그램은 AH헤더를 포함하는 새 IPv6 데이터그램으로 캡슐화된다. 다음 헤더 필드는 각 헤더를 다음 헤더와 연결하는 데 사용된다.

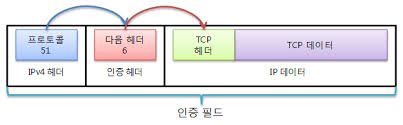
IPv4에서 AH 위치와 연결은 다음 그림과 같다.



원본 IPv4 데이터그램 포맷

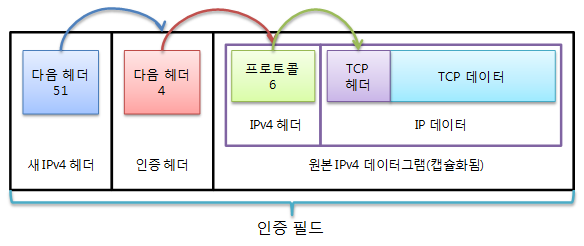
아래는 IPv4 헤더의 프로토콜 식별자에 대한 표이다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ICMP | 1 | IGMP | 2 | TCP | 6 |
| EGP | 8 | UDP | 17 | IPv6 | 41 |
| AH | 51 | SWIPE | 53 | OSPF | 89 |



IPSec 전송 모드 – IPv4 AH 데이터그램 포맷

전송 모드에서 AH 헤더는 IP 헤더와 IP 데이터 사이에 추가된다. IP 헤더의 프로토콜 필드는 AH를 가리키고 AH의 다음 헤더 필드는 IP 헤더의 이전 프로토콜값을 포함한다.

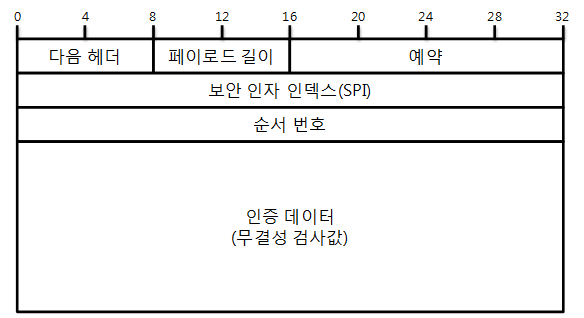


IPSec 터널 모드 – IPv4 AH 데이터그램 포맷

터널 모드에서 IPv4 데이터그램은 AH 헤더를 포함하는 새 IPv4 데이터그램으로 캡슐화된다.

* AH 포맷

AH 포맷 형식은 다음과 같다.



|  |  |  |
| --- | --- | --- |
| 필드 이름 | 크기(Byte) | 설명 |
| 다음 헤더 | 1 | AH 다음에 오는 헤더의 프로토콜 번호를 담고 있다. 헤더를 서로 연결하는데 사용된다. |
| 페이로드  길이 | 1 | 이름과는 달리 이 필드는 페이로드가 아니라 인증 헤더 자체의 길이를 측정한다. 이 길이는 32bit 단위로 측정되며 IPv6에서 헤더 길이를 계산하는 일반적인 방법과 유사하게, 2를 뺀 값으로 표현된다. |
| 예약됨 | 2 | 0으로 설정되고, 사용되지 않는다. |
| SPI | 4 | 32Bit 값으로, 목적지 주소와 보안 프로토콜 유형( AH )과 함께 이 데이터 그램에 쓰이는 보안 연관( SA )를 식별한다. |
| 순서 번호 | 4 | 두 장비간 SA를 구성될 떄 0으로 초기화되는 카운터 필드이다. 이 필드는 그 SA를 사용하여 데이터그램이 송신될 때마다 증가한다. 이 필드는 특정 SA 내에서의 각 데이터그램을 유일하게 식별하며 캡쳐한 데이터그램을 재전송하는 것을 방지함으로써 재생 공격으로부터 IPSec을 방어하는데 사용된다. |
| 인증  데이터 | 가변적 | AH 프로토콜이 수행하는 해싱 알고리즘의 계산 결과인 무결성 검사값( ICV )을 포함한다. 전송 도중에 변경되는 TTL 필드를 제외한 전체 IP 패킷에 해쉬 함수( HMAC-SHA-96, HMAC-MD5-96 등 )을 적용한 결과값을 의미한다. 이 때 이 값에는 변하지 않는 필드인 Version, Internet Hader Length, Total Length, Identification, Protocol, Source Address, Destination Address 등이 계산시 포함된다. |

AH 헤더에 IP 주소가 없는 것은 IPv4와 IPv6에서 동시에 사용되기 위해 꼭 필요하다.

* AH 처리

아웃바운드 AH에 대한 처리는 다음의 순서로 진행된다 :

* Security Association Lookup : 모든 IPSec 프로세싱에서 가장 먼저 이루어지는 단계로 트래픽이 흐르기 전에 SA가 양단간에 설정되는데, 트래픽이 들어왔을 떄 이 SA가 존재하는 지 찾는 과정이다.
* Sequence Number Generation : 처음 SA가 개설되면 Sequence number는 0으로 세팅되고 패킷 전송 후 1씩 증가한다. Cycle이 되지않도록 하는 것이 일반적이며, Overflow가 되면 기존에 쓰던 SA를 삭제하고 새로 SA를 개설하는 절차를 밟는다. 하지만 Cycle 감시를 Disable할 수 있으며 이 경우 SA가 새로 개설되지 않고, 계속 1씩 증가하는 Sequence number를 가지고 패킷이 전송된다.
* Integrity Check Value Calculation : 패킷이 들어오면 패킷의 immutable field를 이용하여 Authentication Data를 만든다. 그렇게 하여 실제 들어온 패킷의 Authentication Data와 계산된 값이 같은지 검사하는데 다르면 들어온 패킷을 버린다.
* Fragmentation : 일반적인 IP fragmentation과 같은 과정으로 AH에서 Fragmentation이 필요하면 AH Processing을 하고 난 후 단편화를 한다. 수신하는 쪽에서는 패킷이 들어왔을 때 모두 재조립을 한 후 AH Processing을 해야한다.

인바운드 AH에 대한 처리는 다음의 순서로 진행된다 :

* Reassembly : 앞서 Fragmentation 과정에서 설명하였듯이 이 과정은 AH Processing이 일어나기 전에 수행 되어야 한다. 그래야 계산된 Authentication 값과 수신된 값이 일치하게 된다.
* Security Association Lookup : 패킷이 재조립되면, 이 패킷을 처리하기 위해 어떤 알고리즘이 필요하고 Key 값은 어떻게 되고, Sequence Number는 체크를 해야하는지 와 같은 것들을 알아야 하는데, 이러한 정보를 가지고 있는 것이 SA ( Security Association )이다. 그래서 SA 을 Lookup 하는 과정을 거쳐 SA가 찾아지면 다음 단계로 넘어가지만, 그렇지 않을 경우에는 패킷은 버려지게 된다.
* Sequence Number Verification : 수신단은 송신단과 마찬가지로 SA가 설립이 되면 Sequence number을 0로 셋팅하고 들어오는 패킷의 Sequence number을 감시합니다. 이 때 중복된 Sequence Number을 가진 패킷이 들어오면 버리게 되는데 윈도우 사이즈를 이용하여 그 안에 들어 왔는지 아닌지를 체크 하여 결정한다. Anti-replay 기능을 위하여 사용된다.
* Integrity Check Value Verification : 실제 Authentication Data값을 비교하는 부분으로 SA lookup 과정에서 찾아진 정보를 이용하여 들어온 패킷의 Authentication Data 값과 계산된 값을 비겨힌디. 이 값이 틀리면 이 패킷은 버려지게 된다.

1. **IPSec 보안 페이로드 캡슐화( ESP )**

* 개요

ESP의 주요 역할은 IP 데이터그램을 암호화하여 프라이버시를 보장하는 것이다. 암호화 알고리즘은 데이터그램의 데이터를 키와 결합하여 암호화된 형식으로 변환한다. 그런 다음 곧 다시 보게될 특수 형식을 사용하여 다시 패키징되고 대상으로 전송한다. 그러면 대상이 동일한 알고리즘을 사용하여 암호를 해독한다. ESP는 또한 AH에서 사용되는 것과 같은 자체 인증 체계를 지원하거나 AH와 함께 사용할 수 있다.

ESP는 통신 당사자가 교환하는 데이터를 암호화하고 해독하는데 사용하는 대칭키를 사용한다. 발신자와 수신자는 보안 통신이 이루어지기 전에 키에 동의해야 한다. VPN은 DES, 3DES, AES 또는 AES-CBC 및 AES-CTR을 사용하여 암호화한다.

또한 ESP는 인증 기능을 제공하기 위해 HMAC-MD5, HMAC-SHA, HMAC-SHA-256 / HMAC-SHA-384, HMAC-SHA-512 및 / AES-XCBC-MAC 알고리즘을 사용한다. 각 알고리즘은 고정 길이 출력 데이터를 생성하기 위해 가변 길이 입력 데이터와 비밀 -키를 사용한다. 두 메시지의 해시가 일치하면 메시지가 동일한 것으로 확인할 수 있다.

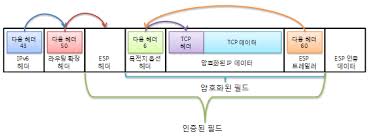
* ESP 필드

ESP에는 AH에서 사용된 것과 동일한 여러 필드가 있지만 매우 다른 방식으로 필드를 패키징한다. 헤더만 있는 대신 필드를 세 가지 구성요소로 나눈다.

* ESP 헤더 : SPI와 Sequence Number라는 두 필드를 포함하며, 암호화된 데이터 앞에 온다.
* ESP 트레일러 : 암호화된 데이터 뒤에 위치하며, 패딩과 패딩 길이 필드를 이용하여 암호화된 데이터를 32비트로 맞춘다. ESP 트레일러는 ESP의 다음 헤더 필드도 포함한다.
* ESP 인증 데이터 : AH 프로토콜과 유사한 방식으로 계산되는 ICV를 포함한다. ESP의 선택적 인증 기능이 적용될 때 사용된다.
* ESP 인증 필드 계산과 위치

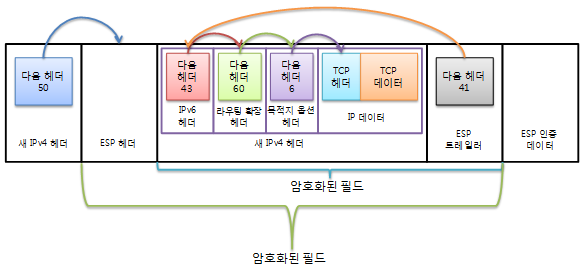
선택적 ESP 인증 기능이 쓰일 경우, 전체 ESP 데이터그램에 대한 계산이 이루어진다. 계산 대상은 ESP 헤더, 페이로드, 트레일러를 포함한다. 마찬가지고 IPv6 버전과 IPv4 버전이 있다.

IPv6에서의 ESP 인증 필드 계산과 위치는 다음과 같다: ( 원본 IPv6은 앞의 그림을 참조하길 바란다. )



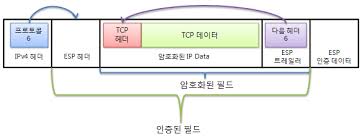
IPSec 전송모드 – IPv6 ESP 데이터그램 포맷

ESP가 전송모드에서 적용될 경우, ESP 헤더 필드는 AH와 마찬가지로 기존 데이터그램 앞에 붙고, ESP 트레일러와 ESP 인증 데이터 필드는 뒤에 붙는다. IP 헤더를 인증하거나 암호화하지 않으므로 데이터그램이 전송되는 동안 잠재적인 공격자에게 주소 지정 정보가 노출될 수 있다. 전송 모드는 터널 모드보다 처리 오버헤드가 적지만 많은 보안을 제공하지는 않는다. 대부분의 경우 호스트는 전송 모드에서 ESP를 사용한다.

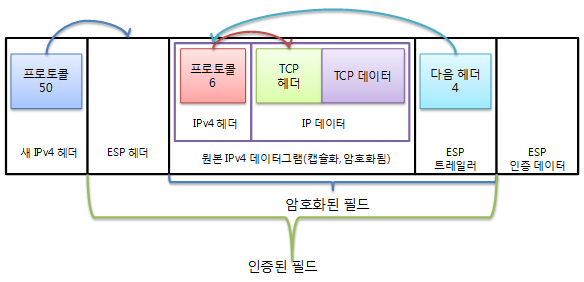


IPSec 터널 모드 – IPv6 ESP 데이터그램 포맷

터널 모드에서 ESP 헤더와 트레일러 필드는 전체 캡슐화된 IPv6 데이터그램을 감싼다. 즉, 새로운 IP 헤더를 생성하고 이를 데이터그램의 가장 바깥쪽 IP헤더로 사용하고 그 다음에 ESP 헤더와 원래 데이터그램을 사용한다. ESP 트레일러 및 선택적 인증 데이터가 페이로드에 추가된다. 암호화와 인증을 모두 사용하면 ESP는 새로운 ESP 패킷의 페이로드 데이터이므로 원래의 데이터 그램을 완전히 보호합니다. 그러나 ESP는 새 IP 헤더를 보호하지 않는다. 게이트웨이는 터널모드에서 ESP를 사용해야 한다.

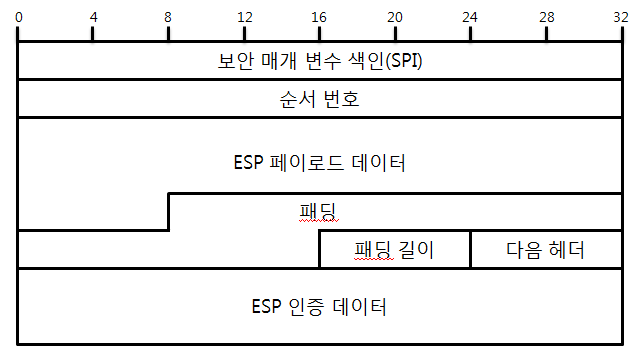


IPSec 전송 모드 – IPv4 ESP 데이터그램 포맷



IPSec 터널 모드 – IPv4 ESP 데이터그램 포맷

* ESP 포맷



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 구간 | 필드  이름 | 크기  (Byte) | 설명 | 암호화  범위 | 인증  범위 |
| ESP  헤더 | SPI | 4 | 32Bit값으로, 목적지 주소와 보안 프로토콜 유형과 함께 이 데이터그램에 쓰이는 보안연관( SA )을 식별한다. |  |  |
| 순서번호 | 4 | 두 장비간 SA가 구성될 떄 0으로 초기화되는 카운터 필드이다. 이 필드는 그 SA를 사용하여 데이터그램이 송신될 때마다 증가한다. 이 필드는 재생 공격으로부터 IPSec을 방어하는데 쓰인다. |  |
| 페이로드 | 페이로드 | 가변적 | 암호화된 페이로드 데이터로 상위 계층 데이터 메시지 또는 캡슐화된 IP 데이터그램으로 구성된다. 이 필드는 일부 암호화 방법에서 필요로 하는 초기화 벡터 같은 지원 정보를 포함할 수 있다. |  |
| ESP  트레일러 | 패딩 | 가변적  (0에서 255) | 암호화 또는 정렬을 위해 추가적인 패딩 바이트가 포함된다. |
| 패딩 길이 | 1 | 위 패딩 필드의 바이트 수 |
| 다음 헤더 | 1 | 데이터그램에서 다음 헤더의 프로토콜 번호를 포함한다. 헤더를 연결하는 데 쓰인다. |  |
| ESP 인증 데이터 | | 가변적 | 선택적인 ESP 인증 알고리즘을 적용하여 계산한 ICV를 포함한다. |  |  |

패딩 필드는 암호화 알고리즘에서 필요로 할 경우 쓰인다. 또한 ESP 트레일러 필드가 32비트 경계에서 끝나도록 하는데 쓰이기도 한다.

* ESP 처리

기본적으로 SPDB( Security Policy Database ) lookup 이라던가 SADB( Security Association Database ) lookup 이라던가 하는 단계들이 존재하나 여기서는 어떤 패킷이 ESP로 처리된다고 결정되었다는 결정하에 실제 ESP가 어떤 동작을 통해 인바운드/ 아웃바운드를 처리하는 지 알아보도록 한다.

아웃바운드 처리는 다음의 순서로 진행한다 :

* Security Association Lookup : 이미 Policy가 결정되었다는 가정 하에 동작하기 때문에 IPSec Processing이 필요한 패킷은 어떤 SA (Security Association)와 연관이 있게 되고, 이 단계에서는 그 SA정보를 찾는 과정을 거치게 된다.
* Packet Encryption : 첫번째 단계에서 찾은 SA정보를 이용하여 실제 패킷을 암호화 하는 과정이다. 여기서는 전송 모드일 때와 터널 모드일 때로 나뉘는데, 전송 모드일 때는 상위 레이어에 대해서만 암호화를 진행하지만 터널 모드일 경우는 원래의 IP 헤더를 포함하여 암호화를 진행한다. 이 과정에서 암호 동기화 데이터가 필요하면 IV을 값을 ESP payload data의 앞 부분에 넣어주게 되고, 만약 ESP 인증도 선택이 되었다면 먼저 암호화를 한 후에 이 패킷을 가지고 인증 데이터를 만들어 넣어 주게 된다.
* Sequence Number Generation : SA가 설립될 때에 Sequence Number값은 0으로 세팅된다. 그리고 패킷이 하나씩 나갈 때 마다 1씩 증가합니다. 그래서 처음 ESP 패킷은 1이 되고, 만약 anti-replay옵션이 선택되었다면 이 값은 반복되지 않는다. 만약 값이 계속 늘어나서 반복되기 바로 전까지 가면 SA를 다시 설립해야 한다. 하지만 anti-replay옵션이 선택되지 않았다면 그냥 계속 증가시키면서 Max 값이 되면 다시 0 부터 시작하게 된다.
* Integrity Check Value Calculation : 만약 Authentication이 선택되었다면, 송신자는 Authentication data부분을 뺀 ESP 패킷(SPI, Sequence Number, Payload Data, Padding, Pad Length, Next Header)의 ICV값을 계산한다.
* Fragmentation : Fragmentation은 IPSec에서 ESP Processing이 끝난 후에 일어난다. 전송 모드의 경우 단편화된 IP에 대하여 ESP Processing이 되는 것이 아니라 전체 IP Data에 대하여 ESP Processing이 진행된 후 그리고 나서 단편화가 진행된다. Tunnel mode의 경우에는 IP 패킷에 대하여 ESP Processing이 적용되는데, 이 때의 IP 패킷은 단편화된 IP 패킷일 수도 있다.

인바운드 처리는 다음의 순서를 진행한다 :

* Reassembly : 이 과정은 ESP Processing이 일어나기 전에 수행되는데 Reassembly가 끝난 후 이 것을 가지고 ESP Processing을 수행하게 된다.
* Security Association Lookup : 어떤 ESP 패킷을 복호화 하기 위해서는 패킷과 관련된 SA가 필요하다. 이 SA을 찾기 위하여 수신단은 ESP Header안에 있는 SPI 값을 이용하여 SA을 찾게 된다. 이렇게 해서 찾은 SA는 Sequence number을 Check 해야 하는지 Authentication을 Check해야 하는 지와 같은 여러가지 정보들을 담고 있다.
* Sequence Number Verification : 수신단은 송신단과 마찬가지로 SA가 설립이 되면 Sequence number을 0로 셋팅하고 들어오는 패킷의 Sequence number을 감시한다. 이 때 중복된 Sequence Number을 가진 패킷이 들어오면 버리게 되는데 윈도우 사이즈를 이용하여 그 안에 들어 왔는지 아닌지를 체크 하여 결정할 수 있다.
* Integrity Check Value Verification : Authentication이 선택되어 있으면 IPSec 은 복호화 전에 먼저 ICV값을 체크 하게 되는데, 여기서 값이 틀리면 패킷을 버리고 확인된 패킷만이 복호화 단계로 넘어가게 된다.
* Packet Decryption : 위의 모든 과정이 끝나게 되면 패킷을 복호화 한다. 전송 모드일 경우는 패킷 복호화후 원래의 IP header와 상위 Layer의 Data를 조합하여 IP 패킷을 재조립하고 터널 모드의 경우는 tunnel IP header 와 ESP Payload 안쪽에 있던 IP header + data 부분을 이용하여 IP 패킷을 재조립 하게 된다.

1. **IPSec 인터넷 키 교환( IKE, Internet Key Exchange )**

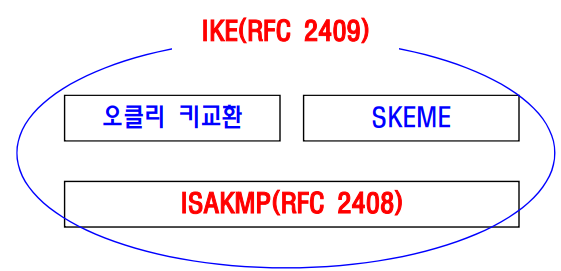
* 개요
* IKE

IKE의 목적은 장비가 안전한 통신을 위해 필요로 하는 정보를 교환하는 것이다. IKE는 IPSec 지원 장비가 SA를 교환하는 방식으로 동작하고, SA는 SAD에 추가되어서 AH와 ESP 프로토콜로 보호된 데이터그램을 교환하는데 사용된다. IKE는 3개의 다른 프로토콜 기능을 결합하기 때문에 혼성 프로토콜로 간주된다.

IKE 그 자체로는 AH 헤더 및 ESP 헤더와는 독립된 것이며, SA를 확립하는 데 필요한 데이터들을 안전하게 전달하기 위한 것이라고 볼 수 있다. 기반이 되는 키 교환, 인증 방식으로는 Diffie-Hellman 알고리즘, 공개키 방식, X.509 디지털 인증서가 있다. 키 전달시에는 UDP 포트번호 500을 이용한다.

IKE에는 버전이 2개가 존재하며 각각 IKEv1, IKEv2가 있다. IKEv1의 경우 1998년 RFC 2409 문서에 의해 확립되었으며 ISAKMP, OAKLEY, SKEME 등의 일련의 다른 프로토콜들을 참조하며, 주 모드와 공격 모드로 나뉘는 단계 1과 신속 모드로 구성된 단계 2가 있다. 이 버전에서의 인증 방법은 사전 공유키( Pre Shared Key, PSK ), 디지털 서명( Digital Signature, RSA-Sig ), Original Public Key Encryption, Revised Public Key Encryption 4가지가 있다.

IKEv2는 2014년 RFC 7296에 의해 확룁되었으며 IKEv1보다 더 가볍고 효율적이나 보안성은 강화되었다. 인증 방법은 PSK와 RSA-Sig, Extensible Authentication Protoco( EAP ) 3가지로 바뀌었다.



* ISAKMP

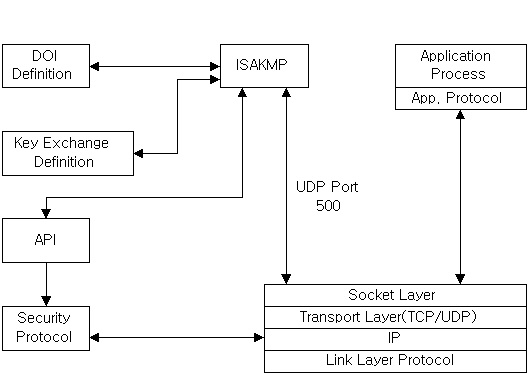
앞에서 AH 및 ESP 프로토콜에 대해서 알아보았다. AH 프로토콜은 송수신자가 같은 키를 공유하고 이 키를 이용해서 해시를 구해서 인증을 수행하며, ESP 프로토콜은 송수신자가 같은키를 공유하고 대칭형 암복호화 알고리즘을 사용하여 패킷을 암복호화한다.

여기서 송수신자가 같은 비밀키를 안전하게 나누어 가지는 방법이 문제가 될 수 있음을 알 수 있다. 원시적인 방법으로는 송수신자가 IPSec을 수행할 프로토콜( AH로 할 지 ESP 로 할 지 ), 알고리즘( DES? AES? SHA-1? MD5? ), 키( 암호화에 어떤 키를 쓰고 인증에 어떤 키를 쓸 것인가 ), 그리고 그 외 속성( 암복호화 및 인증에 쓰이는 세션 키의 유효기간, 키 길이, 터널/전송 모드 )의 정보를 직접 입력하는 방법이 있을 수 있다. 이런 파라미터들을 SA라고 한다. 그리고 송수신자의 관리자가 직접 손으로 SA 정보를 입력하는 방식을 수동키 입력 방식( manual keying )이라고 부른다.

하지만 암호화 프로토콜의 특성상 특정 세션을 암/복호화하는 세션키는 오래 사용하면 암호 분석학적 공격에 취약해 질 수 있으므로, 주기적으로 세션키를 교체해야 한다. 통상적으로 세션키는 1~8시간 또는 (232 ? 1 )개의 패킷을 전송하기 전에 변경되어야 하낟. 그런데 송수신자간에 주기적으로 이렇게 세션 키 및 SA를 입력하는 것은 매우 불편하다.

ISAKMP( Internet Security Association and Key Management Protocol )는 이러한 과정을 자동으로 암호화 키와 보안 연관 정보를 교환하기 위한 구조를 제공하며 보안 연관이 일련의 단계를 통해 협상되도록 함으로써 동작하며 상당히 범용적인 프로토콜로서 설계되었다. 즉, IPSec 프로토콜만을 대상으로 정의한 것은 아니며 IPSec 프로토콜에 대해서는 다음에 다룰 IKE 프로토콜에서 IPSec을 위해 확장한 부분이 포함되어서 사용된다. ISAKMP는 인증 및 키 교환을 위한 프레임워크를 제공하지만( 즉, SA를 설정하는 과정과 각 과정에서 필요한 패킷에 대한 정의 ) 실제 키 교환을 정의하지는 않는다. ISAKMP에서는 사용할 프로토콜과 알고리즘의 키에 동의하고 협상하는 것과 통신하는 상대의 신분을 보증하는 인증, 사용할 키가 합의로 결정된 후에 안전하게 교환될 수 있도록 관리하는 키 관리 기능을 제공한다.

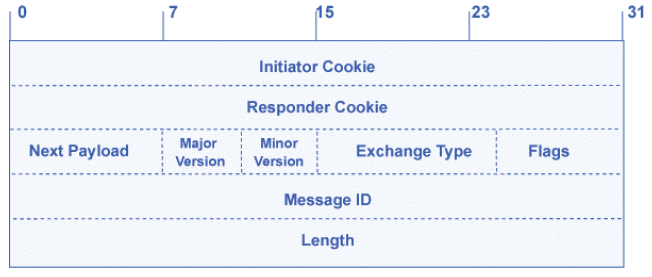
IPSec 프로토콜과 ISAKMP의 관계를 그림으로 표현하자면 다음과 같다 :



그림에서 보듯이 UDP 포트 500을 이용하여 통신한다. DOI는 Domain of Interperter의 약자로 통신할 때 사용되는 값들이 무슨 의미를 가지고 있는지를 정의해 놓은 것이다. ISAKMP는 DOI에서 정의된 대로 페이로드를 생성하고 해석한다. Key Exchange Definition은 DH같은 키 교환 방법을 정의해 놓은 것이고, API는 ISAKMP와 IPSec과의 인터페이스를 위한 것이다. Security Protocol은 앞에서 설명했듯이 ESP와 AH 2가지가 사용된다.

* ISAKMP 헤더

각 ISAKMP 메시지는 ISAKMP 헤더 + ISAKMP 페이로드들의 연속으로 구성된다. 먼저 ISAKMP 헤더의 구조는 다음과 같다 :



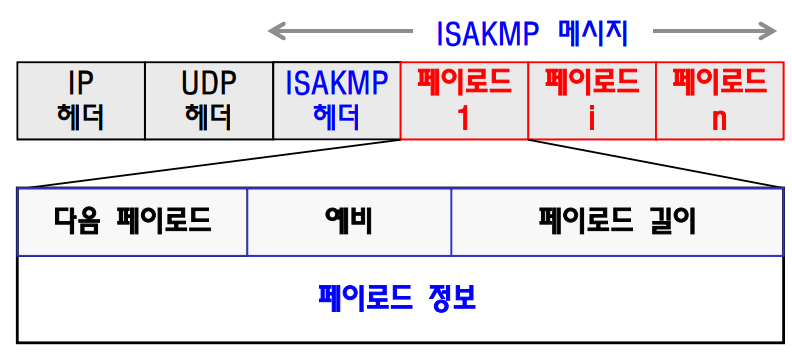
ISAKMP 헤더는 ISAKMP가 동작할 때 기본적으로 붙는 Payload이다. 페이로드는 프로토콜에서 실제 전달되는 의미있는 구조화된 데이터를 말하며, ISAKMP 프로토콜은 DOI에서 정의한 영역 안에서 여러 종류의 페이로드들을 정의하고 있다. 기본적으로 cookie 정보가 들어가는데, cookie 정보는 Random 함수를 이용하여 만드는 게 보통이다.

각각의 필드에 대한 설명은 다음과 같다 :

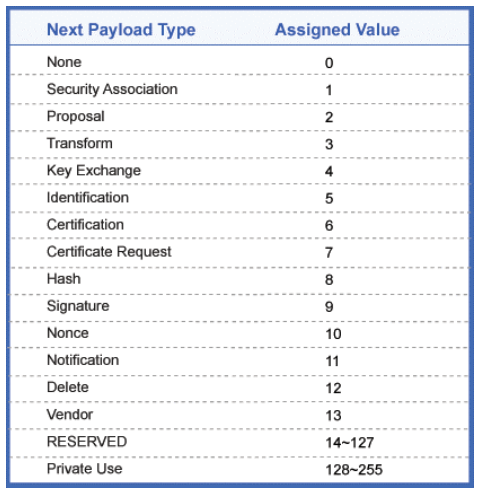
|  |  |  |
| --- | --- | --- |
| Initiator Cookie | 8 옥텟  =8Byte | 게시자에 의해 발생되는 고유 비트열. sa 확립, sa 통지, sa 삭제를 게시하는 실체의 쿠키이다. 상대방으로부터 수신한 쿠키 값이 이전에 수신한 쿠키값과 다른 경우에 수신한 메시지를 폐기하는 방식으로 DoS 공격에 대한 보호용으로 사용한다.  쿠키 생성 방식은 구현에 따라 다르며 표준규격에 규정된 것으로는 고유 비밀 정보를 사용하여 쿠키를 발생시켜야 하며, 쿠키로부터 역으로 그 비밀 정보를 알아낼 수 없어야 한다. |
| Responder Cookie | 8 옥텟 | 응답자에 의해 발생되는 고유 비트열. sa 확립, sa 통지, sa 삭에 반응하는 실체의 쿠키이다. 상대방으로부터 수신한 쿠키 값이 이전에 수신한 쿠키값과 다른 경우에 수신한 메시지를 폐기하는 방식으로 DoS 공격에 대한 보호용으로 사용한다. |
| Next Payload | 1 옥텟 | 다음에 오는 페이로드의 종류를 나타낸다. 아래 페이로드 유형에 대한 표 참조 |
| Major Version | 4 비트 | 사용하는 ISAKMP 프로토콜의 major version. 자신 보다 높은 주 버전이나 부 버전 값을 갖는 ISAKMP 메시지는 받아들이지 않는다. |
| Minor Version | 4 비트 | 사용하는 ISAKMP 프로토콜의 minor version |
| Exchange Type | 1 옥텟 | 단계 1과 단계 2의 협상에서 교환되는 페이로드의 유형과 순서를 규정한다. |
| Flags | 1 옥텟 | ISAKMP 메시지의 페이로드가 암호화 또는 인증되었음을 나타내는 플래그이다. |
| Message ID | 4 옥텟 | 단계 2의 협상 중 프로토콜의 상태를 식별하는 유일한 메시지 식별자. 게시자가 발생시킨 난수 값으로 키 생성에 사용된다. |
| Length | 4 옥텟 | 옥텟 단위의 메시지 전체 길이 |

* ISAKMP 페이로드

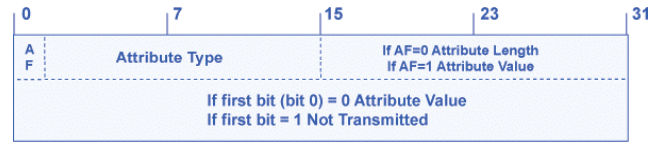
ISAKMP의 패킷 형식은 다음 그림과 같다. 앞에서 설명했듯이 헤더 + 페이로드들로 구성된다 :



ISAKMP 페이로드의 구성은 유형마다 다르지만 다음에 나타나는 페이로드의 유형을 표시하는 다음 페이로드( 다음 페이로드의 종류 ), 예비( RESERVED, 사용되지 않으며 반드시 0으로 세팅되어 있어야 한다 ), 현재 페이로드의 길이 항목( 헤더를 포함한 페이로드의 길이 )을 공통 헤더로 가지고 있고, 이후 페이로드 정보가 나온다. 여기서 페이로드 유형의 종류는 다음과 같다 :



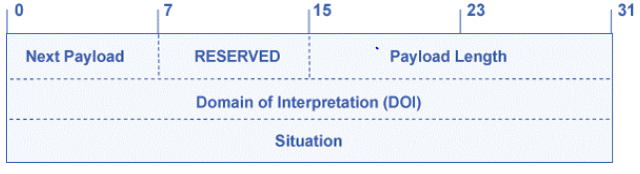
ISAKMP 페이로드 안에는 여러 가지 속성 값들을 기술할 필요가 있는데 이 경우 속성값들을 표현하는 일반적인 구조는 다음과 같다:



|  |  |  |
| --- | --- | --- |
| Attribute Type | 2 옥텟 | 표현하고자 하는 속성의 종류를 나타내며, DOI에서 정의되어 있음. |
| Attribute Length | 2 옥텟 | AF가 영일 때 실제 데이터 길이를, 1인 경우에 데이터 값을 나타낸다. |
| Attribute Value | 가변 | AF가 0인 경우에, 표시된 길이 만큼의 데이터 값을 가진다. |

* 보안연계 페이로드

보안연계 페이로드는 보안연계의 설정 및 협상이 발생하는 도메인을 표시한다. 인터넷에 적용되는 IP 보안 프로토콜 도메인은 한 예이며 ISAKMP는 다른 도메인에도 사용될 수 있다. SA 페이로드는 다음과 같이 구성된다 :

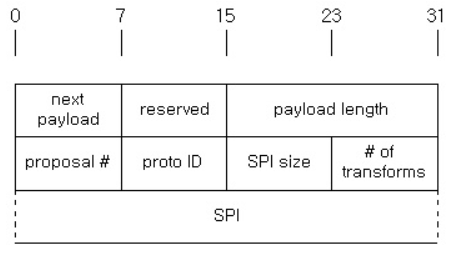


|  |  |  |
| --- | --- | --- |
| DOI | 4 옥텟 | 현재 협상되고 있는 SA의 DOI를 나타낸다. 1단계 SA일 경우에는 0이 사용되며, IPSec DOI에서 해석되어야 하는 경우에는 1의 값이 표시된다. |
| Situation | 가변 | DOI에 따라서 필요한 정보를 기술한다. 이것 역시 IPSec DOI for ISAKMP에 정의되어 있는데 현재 세가지로 정의된다.   |  |  | | --- | --- | | Situation | Value | | SIT\_IDENTITY\_ONLY | 0x01 | | SIT\_SECRECY | 0x02 | | SIT\_INTEGRITY | 0x04 | |

각각의 Situation에 대한 설명은 IPSec DOI for ISAKMP에서 확인할 수 있다. IPSec DOI를 구현할 떄 기본적으로 SIT\_IDENTITY\_ONLY를 지원하도록 정의되어 있다.

* 제안 페이로드

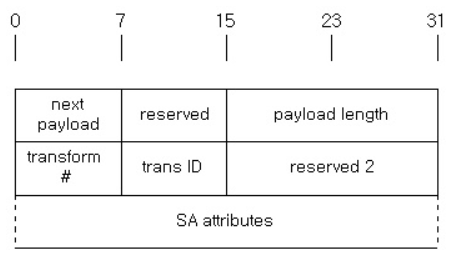
제안 페이로드는 보안연계에 대하여 선호하는 보안 프로토콜과, 해당 보안 프로토콜에 관련된 트랜스폼 페이로드의 개수를 상대방에게 제안하는 페이로드이다. 제안 페이로드는 다음과 같이 구성된다 :



|  |  |  |
| --- | --- | --- |
| 제안 번호 | 1 옥텟 | 제안 페이로드의 식별번호. 예를 들어 3DES ESP 암호와 HMAC-SHA-1 AH 인증이 결합된 보안을 희망하면 게시자는 동일한 제안 번호를 갖는 두 개의 제안 페이로드를 전송해야 한다.  HMAC-MD5 ESP 인증이나 HMAC-SHA-1 AH 인증 중에 하나를 요구한다면 게시자는 단조 증가하는 제안 번호를 갖는 두 개의 제안 페이로드를 전송한다. 제안 선호도가 클수록 제안 번호에 작은 값이 할당되어야 한다. |
| 프로토콜 ID | 1 옥텟 | 현재 협상에 대한 프로토콜의 식별자를 지정  ( 여기에는 ISAKMP, IPSec ESP, IPSec AH 등이 있다. ) |
| SPI 사이즈 | 1 옥텟 | 프로토콜 ID에 의해 정의되는 SPI의 길이 |
| # of 트랜스폼 | 1 옥텟 | 제안 페이로드에 대한 트랜스폼 페이로드의 개수를 지정한다 |
| SPI | 가변 | Security Parameter Index로 패킷의 수신 시에 목적지 ip 주소와 함께 보안연계 데이터베이스의 보안연계를 구분하는 인덱스이다. |

* 트랜스폼 페이로드

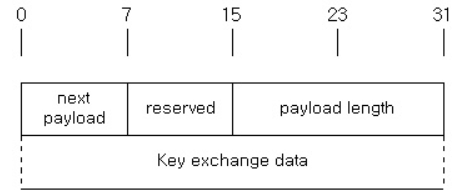
트랜스폼 페이로드는 제안 페이로드에 같이 들어가는데 주로 보안 알고리즘이나 인증 알고리즘 같은 SA의 속성을 정의하고 이것을 알려주는 역할을 한다. 페리로드 구조는 다음과 같다 :



|  |  |  |
| --- | --- | --- |
| 트랜스폼 번호 | 1 옥텟 | 트랜스폼 페이로드의 식별 번호. 제안된 IPSec에 대해 하나 이상의 트랜스폼이 있으면, 가장 선호되는 트랜스폼에 가장 낮은 트랜스폼 번호를 할당한다. |
| 트랜스폼 식별자 | 1 옥텟 | 현재 제안 내에서 프로토콜을 위한 식별자로 쓰인다. 즉, 제안의 프로토콜 ID가 1이고 트랜스폼 식별자가 1이라면 ISAKMP 프로토콜에서 IKE를 쓴다는 의미로 해석할 수 있다. 자세한 내용은 IPSec DOI for ISAKMP에서 확인할 수 있다.  AH 트랜스폼 식별자 : MD5, SHA, RIPEMD, …  ESP 트랜스폼 식별자 : DES, 3DES, AES, …  IPComp 트랜스폼 식별자 : DEFLATE, LZS, … |
| 예비 2 | 2 옥텟 |  |
| 보안연계 속성 | 4 옥텟 | 트랜스폼 식별자에 정의된 것에 대한 속성 값들을 정하는 용도로 사용된다. 전달될 수 있는 정보들은 다음과 같다 :  프로토콜 동작 모드 : 터널 or 전송 모드  보안연계의 수명 유형 : 경과시간 또는 보안연계가 적용된 총 바이트의 수  보안연계의 수명 기간 : 보안연계의 수명 주기를 결정하는 변수 |

* 키교환 페이로드

키교환 페이로드는 다양한 키 교환을 지원하기 위해 설계되었다. 예를 들어 Oakley나 Diffie-Hellman과 같은 키 교환 방식을 지원하기 위해, Key Exchange Data 필드에 세션키를 만들기 위해 필요한 데이터를 담아 보내준다. 그러면 이 데이터를 이용하여 세션키를 만들어 실제 암/복호화에서 이용하게 된다. 키교환 페이로드의 구조는 다음과 같다 :



* 기타 페이로드

기타 페이로드는 인증 페이로드 등의 위에서 언급한 페이로드들이 있다.

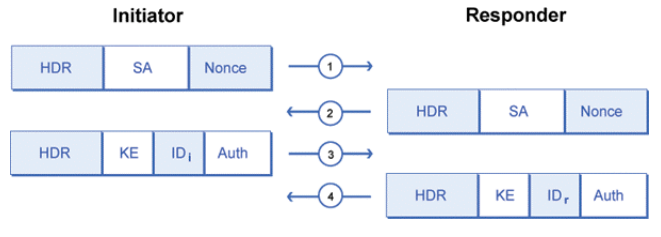
* 키 교환

앞에서 언급했지만 ISAKMP는 키 교환 기술에 대해서 정의하지 않는다. 키 교환 기술은 OAKLEY 및 SKEME( Secure Key Exchange Mechanism for Internet )프로토콜에서 정의한다. OAKLEY는 다양한 키 교환 “모드”를 정의하며 키 교환을 위한 구체적인 방법을 설명한다. OAKLEY의 주요 기능으로 man-in-the-middle 공격으로부터 보호하는 인증과 키에 대한 PFS( Perfect Forward Secrecy ) 및 통신 당사자의 신원을 보호하는 Identity protection이 있다. IKE 키 교환 절차 대부분은 OAKLEY에 기반을 두고 있다. 이 경우, ISAKMP / OAKLEY 프로토콜이라 하며, 이렇게 키 교환 프레임워크와 키 교환 기술을 묶어 IKE라고 부른다.

SKEME은 OAKLEY와는 다른 키 교환 방법을 설명하며 IKE는 SKEME의 일부 기능을 이용한다. SKEME은 공개 키 기반 키 교환, 키 배포 센터 및 수동 설치를 지원하고 안전하고 빠른 키 갱신 방법을 설명한다.

* ISAKMP 교환 방법

ISAKMP의 헤더와 페이로드를 이용하여 ISAKMP 프로토콜은 협상을 진행하게 된다. 이 때 이용하는 교환 방식은 Base Exchange Type, Identity Protection Exchange type, Authentication Only Exchange Type, Aggressive Exchange Type, Infomational Exchange Type 5가지가 있다.

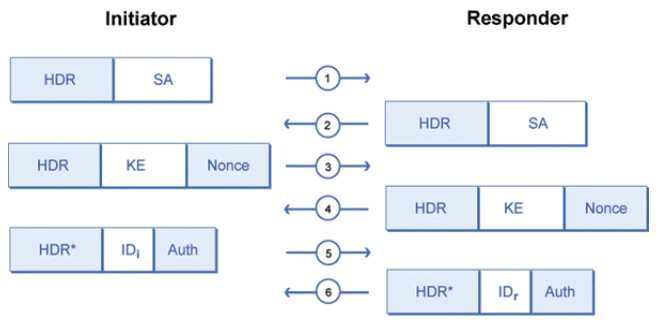


Base Exchange는 총 4단계의 절차를 걸쳐서 협상을 진행하게 된다( HDR : SA 헤더, SA : SA 페이로드, Nonce : Nonce 페이로드, 암호학적으로 건전한 연산을 수행할 수 있음을 입증하는데 사용되며 이 페이로드의 값은 키를 새생성하는 데 사용됨, KE : 키 교환 페이로드이며 키 생성을 위한 값을 교환하는 데 사용, ID : 자기 ID 정보를 가지며 IDi는 발신자의 ID 정보를 IDr은 수신자의 ID 정보를 담음, AUth : 인증 페이로드이며 메시지를 인증하고 현재 sa에서 키가 적절히 공유되었는지 확인하기 위해 사용 ).

먼저 1단계에서는 SA 페이로드 안에 보안 연계, 제안, 트랜스폼 페이로드를 포함하여 전송한다. 즉, 송신자 입장에서는 자신이 지원할 수 있는 알고리즘의 종류를 수신자가 알 수 있게 함으로써 수신측이 자신이 보내준 것 중에 하나를 선택할 수 있게 한다.

이렇게 해서 선택된 것이 암호화 알고리즘 같은 것들이 같은 형식의 포맷으로 위 그림의 2단계와 같이 송신 측으로 전달된다. 이 때 공통적으로 들어가는 것이 NONCE인데, 이것은 재생 공격을 방지하기 위해 랜덤값이 들어가게 된다.

3,4 단계에서는 1,2단계에서 결정된 SA를 이용할 때 쓰일 Key Material 값을 서로 주고 받게 되는데, 이 떄 자신의 ID와 Authentication을 위한 값을 함께 보내주게 된다. 이 방식의 특징은 키 교환 정보와 인증 정보를 한 메시지에 담아 보냄으로써 협상 단계를 줄였지만, Identity 정보가 공통키가 설립되기 전에 교환되기 때문에 Identity Protection은 지원하지 않게 된다.

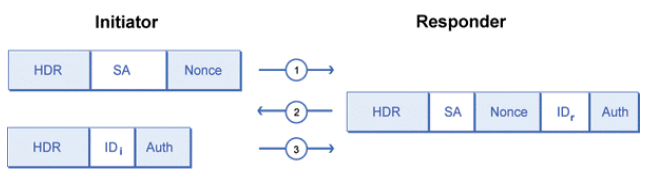


Identity Protection Exchange Type 방법은 키 교환 정보를 Identity, Authentication 정보와 분리시키기 위해 디자인되었다( 식별 정보와 인증 정보는 3,4 단계까지 협상하여 설정된 비밀키에 의해 보호를 받게 된다. )

협상 과정을 살펴 보면 그림과 같이 1,2단계에서는 SA 페이로드 안에 보안연계, 제안, 트랜스폼 정보를 담아 수신자에게 보내주면 수신자가 Base Exchange와 마찬가지로 여러 제안 중 하나를 선택하여 송신자에게 전송한다.

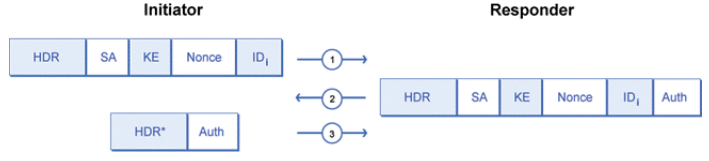
그리고 3,4 단계에서 송신자와 수신자는 키 교환 정보와 재생 공격 방지를 위한 Nonce 정보를 교환한다.

이렇게 해서 나누어 가진 비밀키 정보를 가지고 5,6 단계에서는 헤더를 제외한 ID와 인증 정보를 암호화 하여 서로 송수신하게 된다. 이렇게 하면 서로의 ID 정보와 인증 정보를 보호할 수 있다.



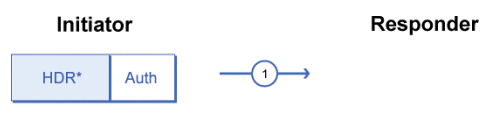
Authentication Only Exchange Type은 인증과 관련된 정보만을 전송하기 위해 설계되었다. 이 방벙의 이점으로는 암호화를 위한 Key 관련 계산을 안 하기 때문에 협상 속도가 빠르다는 장점이 있다. 그리고 이 방식을 다른 방법을 이용하여 안전하게 암호화할 수 있는데, ISAKMP의 단계 1에서 암호화를 위한 SA( 보안연계 )가 협상되었다면 이 때 협상된 ISAKMP SA를 이용하여 단계 2에서 Authentication Only Exchange를 암호화하는 방식으로 안전하게 다음 프로토콜을 위한 SA 교환을 시행할 수 있다.

이 방식의 교환 방법을 살펴 보면 송신자가 SA( 보안연계, 제안, 트랜스폼 )와 Nonce 정보를 보낸다. 그러면 수신자는 선택된 정보를 SA에 담고 자신의 ID 정보와 인증 정보를 같이 보내주게 된다. 이러면 송신자 측에서 인증을 확인할 수 있고, 확인 후에 송신자의 ID와 Nonce 정보를 수신자에게 보내주게 된다.



Aggressive Exchange는 SA 정보와 KE, 인증 정보가 한꺼번에 전송되도록 하기 위해 디자인되었다. Aggressive Exchange는 이렇게 한꺼번에 관련 파라미터들을 전송함으로써 협상 단계를 줄이는 효과를 가진다.

협상 과정을 보면 앞의 내용과 큰 차이는 없다. 송신자는 자신이 쓸 수 있는 알고리즘이나 정책들을 SA( 보안 연계, 제안, 트랜스폼 )에 담아 보내고 수신자는 선택하여 보내준다. 이 때 수신자는 자신의 ID과 인증 정보를 보내주고 이것을 받은 송신자는 인증 정보를 암호화하여 수신자에게 보내주게 된다. 만약 이것이 에러가 발생하게 되면 수신자가 Informational Exchange Type을 이용하여 Notify 페이로드를 송신자에게 전달할 수 있다.



Informational Exchange Type은 어떤 에러나 정보를 알려주기 위한 단방향 통신을 지원하기 위해 설계되었다. 만약 KE 교환 전에 Informational Exchange가 발생한다면 암호화되지 않고 전송되지만, SA( 보안 연계 )가 설립되어 있다면 그 SA를 이용하여 암호화되어 전송된다. 보통 에러를 알리거나 SA를 삭제할 때 많이 이용된다.

* IKE 동작과 Treck의 구현 예

ISAKMP 단계 1( main mode 또는 aggressive mode )에서는 두 장비가 이후에 정보를 어떻게 안전하게 교환할지에 동의하는 준비 단계이다. 두 장비는 협상을 통해 ISAKMP SA를 생성하며 ISAKMP SA가 협상에 쓰이는 속성들은 DES와 같은 암호화 알고리즘, 해시 알고리즘, 인증 방법, Diffe-Hellman 그룹, 수명 유형, 수명 기간, 의사 랜덤 함수, 키 길이가 있다. 일반적으로 단계 1에는 주 모드 및 공격 모드 2가지가 존재하며, 주 모드는 DH 교환을 처리하고 유효화하는 3개의 교환으로 구성되며( Identity Protected Exchange 기법 ) 공격 모드는 단일 교환 내에서 교환( Aggressive Exchange 기법 )을 수행한다.

Treck IPSec에서 1단계 협상 속성은 암호화 알고리즘, 해시 알고리즘, 인증 방법, 디피-헬먼 그룹, 수명 유형, 수명 기간, 의사 랜덤 함수, 키 길이가 있다. 단계 1 프로토콜 식별자는 TM\_DOI\_PROTO\_ISAKMP이고, 변환 식별자는 TM\_DOI\_KEY\_IKE이다.

암호화 알고리즘은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_IKE\_DES\_CBC | 1 |  |
| TM\_IKE\_IDEA\_CBC | 2 |  |
| TM\_IKE\_BLOWFISH\_CBC | 3 |  |
| TM\_IKE\_RC5\_R16\_B64\_CBC | 4 |  |
| TM\_IKE\_3DES\_CBC | 5 |  |
| TM\_IKE\_CAST\_CBC | 6 |  |
| TM\_IKE\_AES\_CBC | 7 |  |
| TM\_IKE\_TWOFISH\_CBC | 13 |  |

해시 알고리즘은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_IKE\_MD5 | 1 | MD5 해시 알고리즘 |
| TM\_IKE\_SHA1 | 2 | SHA1 해시 알고리즘 |
| TM\_IKE\_SHA256 | 4 | SHA256 해시 알고리즘 |
| TM\_IKE\_SHA384 | 5 | SHA384 해시 알고리즘 |
| TM\_IKE\_SHA512 | 6 | SHA512 해시 알고리즘 |
| TM\_IKE\_REPEMD | 65001 | RIPEMD 해시 알고리즘 |

인증 방법( Authentication methods )은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_IKE\_PRESHARED\_KEY | 1 | 사전 공유 키 |
| TM\_IKE\_DSS\_SIG | 2 | DSA 서명 |
| TM\_IKE\_RSA\_SIG | 3 | RSA 서명 |

Diffie-Hellman 그룹은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_DHGROUP\_1 | 1 | 768 MODP 그룹 |
| TM\_DHGROUP\_2 | 2 | 1024 MODP 그룹 |
| TM\_DHGROUP\_5 | 5 | 1536 MODP 그룹 |
| TM\_DHGROUP\_14 | 14 | 2048 MODP 그룹 |

Treck에서 사전 정의된 Diffie-Hellman MODP 그룹 1,2,5 및 14를 지원함. Treck IKE는 미리 정의된 MODP 그룹만 지원함. 그룹 3, 4 및 사용자 정의 그룹은 지원되지 않음.

수명 유형( Lifetime Type )은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_IKE\_LIFETYPE\_SECONDS | 1 | 생활 시간 (초) |
| TM\_IKE\_LIFETYPE\_KBYTES | 2 | 킬로그램 단위의 수명 |

수명 기간( Lifetime Duration )의 경우, Treck IKE는 SA 수명이 2 옥텟( AF 비트 세트의 경우 0 ~ 18.2 시간의 수명 ) 또는 4 옥텟( 0 ~ 136년, AF 비트가 설정되지 않음 ) 중 하나여야함을 요구함. 다른 데이터 길이는 지원되지 않음.

의사 랜덤 함수의 경우 협상된 해시 알고리즘의 HMAC버전을 PRF로 사용함. 다른 PRF는 지원되지 않음.

키 길이에 대해서 설명하자면, 일부 암호화 알고리즘이 다른 크기의 키를 허용할 수 있다. RFC에서 권장한 대로 8비트의 배수가 아닌 키 길이를 지원하지 않음. 키 길이가 고정된 알고리즘의 경우 키 길이 속성은 허용되지 않음. 종류는 다음과 같다 :

|  |  |  |
| --- | --- | --- |
| TM\_IKE\_DES\_CBC | 64 비트 |  |
| TM\_IKE\_BLOWFISH\_CBC | 40 ~ 448 비트 |  |
| TM\_IKE\_RC5\_R16\_B64\_CBC | 40 ~ 2040 비트 |  |
| TM\_IKE\_3DES\_CBC | 192 비트 |  |
| TM\_IKE\_CAST\_CBC | 40 ~ 128 비트 |  |
| TM\_IKE\_AES\_CBC | 128, 192 또는 256 비트 |  |
| TM\_IKE\_TWOFISH\_CBC | 128, 192 또는 256 비트 |  |

이 협상 이후는 IKE 협상 과정을 보호하게 된다. main mode는 6번의 메시지를 교환하며 aggressive mode는 3번의 메시지 교환으로 완료되며 ID는 보호되지 않는다.

ISAKMP 단계 2( quick mode만 지원함 )에서는 1단계에서 수립한 ISAKMP SA를 이용하여 기타 보안 프로토콜을 위한 SA를 생성한다. 보통 여기에서 AH와 ESP 프로토콜을 위한 진짜 SA의 인자를 협상한다. 이 단계에서 SA 및 키를 협상하며 3번의 메시지를 교환하게 되는데 단계 1( Phase 1 )에 비해 계산량이 적다. Key refresh( Rekeying )를 위해 자주 수행하며 동일한 Phase 1 IKE SA 보호 하에 여러 번의 Phase 2 교환 가능하다.

2단계 협상 속성은 RFC 2407에 따라 프로토콜 식별자 및 변환 식별자, 수명 유형, 수명 기간, 디피-헬먼 그룹, 캡슐화 모드, 인증 알고리즘, 키 길이, 주요 라운드가 있다. 단계 2 프로토콜 식별자는 AH와 ESP를 지원하므로 TM\_DOI\_PROTO\_IPSEC\_AH와 TM\_DOI\_PROTO\_IPSEC\_ESP가 있다.

AH의 경우 다음 변환 식별자가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_DOI\_AH\_MD5 | 2 | HMAC\_MD5\_96 |
| TM\_DOI\_AH\_SHA | 삼 | HMAC\_SHA1\_96 |
| TM\_DOI\_AH\_SHA2\_256 | 5 | HMAC\_SHA2\_256\_128 / 96 |
| TM\_DOI\_AH\_SHA2\_384 | 6 | HMAC\_SHA2\_384\_192 / 96 |
| TM\_DOI\_AH\_SHA2\_512 | 7 | HMAC\_SHA2\_512\_256 / 96 |
| TM\_DOI\_AH\_RIPEMD | 8 | HMAC\_RIPEMD160\_96 |

ESP의 경우 다음 변환 식별자가 있다 :

|  |  |
| --- | --- |
| TM\_DOI\_ESP\_DES\_IV64 | 1 |
| TM\_DOI\_ESP\_DES | 2 |
| TM\_DOI\_ESP\_3DES | 3 |
| TM\_DOI\_ESP\_RC5 | 4 |
| TM\_DOI\_ESP\_IDEA | 5 |
| TM\_DOI\_ESP\_CAST128 | 6 |
| TM\_DOI\_ESP\_BLOWFISH | 7 |
| TM\_DOI\_ESP\_NULL | 11 |
| TM\_DOI\_ESP\_AES | 12 |
| TM\_DOI\_ESP\_TWOFISH | 13 |

수명 유형은 다음과 같은 종류가 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_DOI\_SATOL\_LIFE\_SECONDS | 1 | 생활 시간 (초) |
| TM\_DOI\_SATOL\_LIFE\_SECONDS | 2 | 킬로그램 단위의 수명 |

수명 기간은 2 옥텟( AF 비트 세트의 경우 0 ~ 18.2 시간의 수명 ) 또는 4 옥텟( 0 ~ 136년, AF 비트가 설정되지 않음 ) 중 하나여야함을 요구한다. 다른 데이터 길이는 지원되지 않는다.

Diffie-Hellman 그룹은 다음이 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_DHGROUP\_1 | 1 | 768 MODP 그룹 |
| TM\_DHGROUP\_2 | 2 | 1024 MODP 그룹 |
| TM\_DHGROUP\_5 | 2 | 1536 MODP 그룹 |
| TM\_DHGROUP\_14 | 2 | 2048 MODP 그룹 |

사전 정의된 MODP 그룹만 지원하며, 그룹 3, 4 및 사용자 정의 그룹은 지원되지 않는다.

캡슐화( ESP ) 모드는 다음이 있다 :

|  |  |
| --- | --- |
| TM\_DOI\_ENCAPMODE\_TUNNEL | 1 |
| TM\_DOI\_ENCAPMODE\_TRANSPORT | 2 |

즉, 전송 모드 및 터널 모드 모두 지원된다.

인증 알고리즘은 다음이 있다 :

|  |  |
| --- | --- |
| TM\_DOI\_AUTHALG\_HMAC\_MD5 | 1 |
| TM\_DOI\_AUTHALG\_HMAC\_SHA\_1 | 2 |
| TM\_DOI\_AUTHALG\_HMAC\_SHA2\_256 | 5 |
| TM\_DOI\_AUTHALG\_HMAC\_SHA2\_384 | 6 |
| TM\_DOI\_AUTHALG\_HMAC\_SHA2\_384 | 7 |
| TM\_DOI\_AUTHALG\_HMAC\_RIPEMD | 8 |

키 길이에 대한 설명은 단계 1의 키 길이 항목과 같으며 다음이 있다 :

|  |  |  |
| --- | --- | --- |
| TM\_DOI\_ESP\_DES\_IV64 | 64 비트 |  |
| TM\_DOI\_ESP\_DES | 64 비트 |  |
| TM\_DOI\_ESP\_3DES | 192 비트 |  |
| TM\_DOI\_ESP\_RC5 | 40 ~ 2040 비트 |  |
| TM\_DOI\_ESP\_CAST | 40 ~ 128 비트 |  |
| TM\_DOI\_ESP\_BLOWFISH | 40 ~ 448 비트 |  |
| TM\_DOI\_ESP\_NULL | N / A |  |
| TM\_DOI\_ESP\_AES | 128, 192 또는 256 비트 |  |
| TM\_DOI\_ESP\_TWOFISH | 128, 192 또는 256 비트 |  |

주요 라운드에서 일부의 암호화 알고리즘 경우 따라 사용자가 정의한 ROUND 값을 사용할 수 있으나 상호 운용에 영향을 미친다. 항상 기본 라운드 값을 사용하는데 예를 들면 RC5, Blowfish, Cast 알고리즘은 16 라운드를 사용한다.

1. **IPSec 보안 구성 요소**

* 보안 정책과 보안 정책 데이터베이스
* 보안 정책에 대한 개요

보안 정책은 IPSec 구현에 내장된 규칙으로, 장비가 수신하는 서로 다른 데이터그램을 어떻게 처리할지 지시하고 보안을 어떻게 제공할지에 대해 전반적인( 필요시 구체적인 ) 지침을 기술한다. 즉, IPSec의 보안 정책은 암호화되지 않은 프로토콜을 사용해 서로 통신하는 두 호스트 사이에서 서명을 하여 암호화된 통신을 제공하는 것이라고 볼 수 있다. 기존 HTTPS나 SSTP같은 보안 통신들은 해당 프로토콜 외 다른 통신은 암호화하지 않는다. 예를 들어 FTP나 Telnet같은 서비스를 이용할 때는 암호화가 되지 않아서 취약할 수 있는 부분을 IPSec을 적용하여 주고 받는 패킷 모두에게 암호화를 적용할 수 있다.

* IPSec의 오버헤드와 보안 정책

IPSec에 대해서 알아야할 사항 중 하나는 패킷 크기가 커지고 암호화 및 복호화에 필요한 오버 헤드로 인해 정상적인 IP패킷보다 속도가 느리다는 것이다. 크기가 클수록 IPSec은 기존 IP 패킷보다 더많은 네트워크 대역폭을 사용할 수 있다. 결과적으로 실제로 보안이 필요한 통신에만 IPSec을 사용하려고 할 것이다. 여기서 보안 정책을 통해 IPSec으로 통신해야하는 패킷과 기존 IP로 통신해야하는 패킷을 설정할 수 있다.

* 보안 정책의 종류

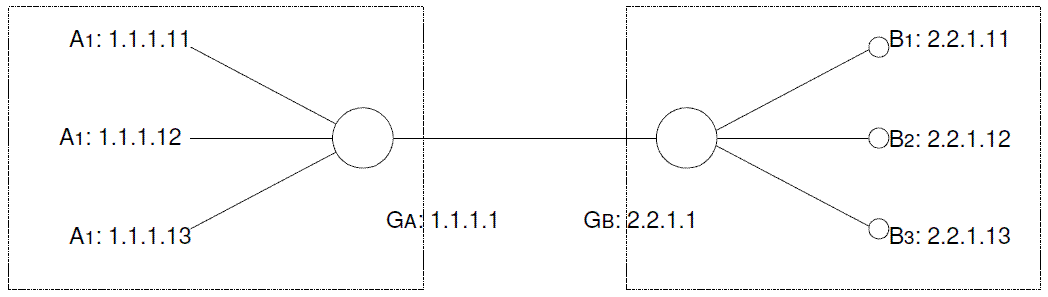
보안 정책의 종류는 3가지 방식이 있다. 첫 번째는 Secure Server 방식이다. IPSec을 사용하지 않는 모든 호스트에 대해 모두 차단시켜버리는 방식이다. 보안 수준은 매우 높지만 인터넷이라던지 내부 네트워크 밖에 잇는 호스트들은 IPSec 정책을 사용하지 않는 경우가 많기 때문에 Secure Server 방식으로 사용하면 인터넷이 되지 않고 다른 네트워크와 통신하는데 지장이 생길 수 있다. 그래서 이 방식은 보안이 요구되는 내부 네트워크 안에서만 사용하는 방식이다.

두번째 방식은 Server 방식으로 require하는 것이 아니라 request하는 것이다. 통신하는 상대방에서 IPSec을 사용하라고 요청하고 사용하지 않는다면 서버에서도 사용하지 않고 통신을 수행하고, 상대방이 사용한다면 사용하는 상대방에 맞추는 동적인 방식이다.

세번재 방식은 Client 방식으로, Server나 Secure Server가 있다면 IPSec 사용하는 것을 수락한다는 방식이다. 이 정책은 일반적으로 보안에 대해 걱정할 필요가 없는 클라이언트 컴퓨터에서 실행되도록 설계되었다. 기본적으로 보안 통신을 절대로 시작하지 않도록 설계되어있지만, 서버가 요청하면 적절하게 응답하는 방식이다. 위의 3가지 보안 정책이 일반적이지만 원한다면 처음부터 새로운 정책을 작성할 수 있다.

* 보안 정책 데이터베이스

대개 보안 정책은 장비의 보안 정책 데이터베이스( SPD, Security Policy Database )에 저장되어 있다. 보안 정책 데이터베이스는 IP 패킷을 SA에 연관시키는 방법을 정의한 데이터베이스이다. SPD는 호스트 또는 보안 게이트웨이로부터 인바운드 또는 아웃바운드하는 모든 IP 트래픽의 처리를 결정하는 정책을 지정한다. 즉, 비 IPSec 트래픽을 포함한 모든 트래픽( 인바운드, 아웃바운드 모두 )을 처리하는 동안 참조해야 한다는 것이다. SPD의 정책 항목은 정렬되어있으며 일치하는 첫번째 정책이 트래픽을 처리하는 데 사용된다. 정책의 요소는 원격 IP 주소, 로컬 IP 주소, 로컬과 원격 포트, 다음 계층 프로토콜, 동작 등을 가지고 있다. 아래의 그림을 예로 들겠다.

[](http://wiki.treck.com/File:Subnet_Network_01.gif)

두 개의 IPv4 서브넷 네트워크, 서브넷 A (1.1.1.0/24)와 서브넷 B (2.2.1.0/24), 보안 게이트웨이 GA와 GB가 있다고 가정한다. 이 다이어그램은 호스트 A에서 서브넷 B로, 또는 호스트 A에서 호스트 B로까지 연결될 수 있는데, 이 경우 GA 또는 GB는 IPSec이 가능한 호스트가 된다. 이러한 다이어그램일 때 GA의 SPD 표는 다음과 같다고 하자:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **첨자** | **방향** | **로컬**  **IP,**  **(공유?)** | **로컬**  **포트,**  **(공유?)** | **원격**  **IP**  **(공유?)** | **원격**  **포트,**  **(공유?)** | **프로**  **토콜,**  **(공유?)** | **인바운드**  **1st**  **SA entry** | **아웃바운드**  **1st**  **SA entry** | **액션** | **정책 1st content** |
| 9 | In & Out | 1.1.1.12  Yes | 80  Yes | 2.2.1.0/24 Yes | Any  Yes | TCP  Yes | sa15 | sa25 | IPSEC | Cont1 |
| 8 | In & Out | 1.1.1.0/24 Yes | Any  Yes | 2.2.1.0/24 Yes | Any  Yes | Any  No | sa10 | sa20 | IPSEC | Cont3 |
| … | ... | … | … | … | … | … | … | … | … | … |
| ... | In | … | … | … | … | … | … | … | … | … |
| ... | Out | … | … | … | … | … | … | … | … | … |
| 0 | In & Out | Any | Any | Any | Any | Any | NULL | NULL | BYPASS | NULL |

정책 순서에 따라 패킷 처리 동작이 결정된다. 정책 테이블은 항상 맨 위 (가장 선호되는 것부터)에서 맨 아래( 정책 인덱스가 0인 최하위 우선 순위 )까지 찾는다. 첫 번째 일치 정책은 패킷을 처리하는 데 사용된다. 정책 인덱스 번호 0은 ALL-BYPASS 또는 ALL-DISCARD 정책으로 권장된다. 그렇지 않으면 패킷이 적합한 정책을 찾을 수 없는 경우 ‘NO-POLICY’ 오류가 반환된다.

표에서 정책 Index.9는 인바운드 및 아웃바운드 일치 트래픽에 적용되는 기본 정책이다. 트래픽이 정책 9의 선택자와 일치하지 않으면 다음 정책( 정책 Index.8 )과 정책 Index.7을 확인하게 된다. 정책에는 하나 이상의 정책 내용이 있으며 각 내용은 IPSec 프로토콜 ( AH 또는 ESP )에 해당하지만 두 가지 모두에 해당하지는 않는다. 정책에 ESP와 AH가 모두 필요한 경우 두 개의 별도 정책 내용을 사용해야 하며 다음 내용 포인터를 통해 함께 연결된다. 예를 들어, 정책 No.9에는 ESP와 AH 모두 필요

* 보안 연관과 보안 연관 데이터베이스
* 보안 연관

보안 연관( SA, Security Association )은 한 장비와 다른 장비 사이에 맺은 특정한 종류의 보안 연결을 설명하는 보안 정보이다. 두 장비 간에 안전한 통신을 하기 위해 사용하는 보안 방법을 명시한 계약서라고 볼 수 있다. 이 때 암호 알고리즘, 키 교환 방법, 암호화 키 교환 주기 등에 대한 합의가 있어야 하는데 이 합의는 통신 연결 이전에 쌍방 간에 조율되어야 한다. 보안 연관의 특징은 각 SA마다 단방향성이고, 양 방향성을 이루려면 쌍으로 구성되게 해야한다는 것이다. 따라서 SA마다 이를 구분하는 식별자( SPI )가 있게 된다.

보안 연계( = 보안 연관 )은 두 가지 종류로 분류할 수 있는데, 인터넷 키 교환용 보안 연계( IKE SA )와 IPSEC 보안 연계( IPSEC SA )가 있다. IKE SA는 쌍방간에 암호 알고리즘, 인증 방법, 마스터키 교환 절차에 의해 설정되는 안전한 보안 연결이며, SA를 사용하여 쌍방간 다시 IPSec SA 설정을 위한 암호 협상, 상호인증, 세션키 설정 기능을 안전하게 수행할 수 있도록 한다. IKE SA를 설정시 마스터키는 일반적으로 Diffie-Hellman 알고리즘에 기반을 둔 키 교환 방식에 의해 설정된다. 중간자 공격을 방지하기 위해 STS( Station-To-Station ) 프로토콜이라고 불리어지는 키 분배 프로토콜은 공개 키 인증서와 디지털 서명을 이용하여 DH 키 교환을 수행하는 두 장치간에 서로 인증하는 절차를 동반한다.

IPSec 보안연계는 IKE SA에서 안전하게 설정되며 단방향이므로 각 방향에서 독립된 IPSec SA들이 설정되어야 한다. 이후, 이렇게 설정된 IPSec SA를 이용하여 IPSec 패킷들이 쌍방간에 안전하게 전송될 수 있다. IPSec SA까지 설정되기 위해서는 다음과 같은 두 단계를 거쳐야 하는데 이것은 IKE에서 설명하는 단계 1의 주 모드와 공격적 모드 및 단계 2의 신속 모드이다.

IPSec-Treck에서 SA는 정책 내용에 따라 생성된다. 각 SA는 AH 또는 ESP 보안 서비스를 제공할 수 있지만 둘 다 제공할 수는 없다. ESP와 AH가 모두 필요한 경우, SA 그룹( SA 번들 )이 필요하다. 각 정책의 Yes-No 공유 플래그는 정책에 따라 SA를 작성하는 방법을 결정한다. 해당 선택자 필드가 정책값 또는 패킷값을 사용하는지 여부를 나타내빈다. 공유가 정의되어 있지 않으면 패킷 값을 사용하여 SA를 작성한다. YES이면 정책값을 사용하여 SA를 작성한다. 예를 들어 정책 8번의 정책 선택자/프로토콜 필드는 ANY로 설정되어 있지만 공유는 금지되어 있으므로 TCP와 UDP는 이 정책과 일치하지만 서로 다른 SA를 사용해야 한다. 즉, 서로 다른 암호화 키와 인증 키를 사용하여 트래픽을 보호해야 한다. 공유가 허용되면 정책 값( ANY 프로토콜 )을 사용하여 SA를 작성하며 TCP와 UDP 모두 동일한 SA를 사용한다.

* 트리플

보안 연관을 식별하기 위한 매개변수는 보안 매개변수 색인( SPI : Security Parameters Index ), IP 목적지 주소, 보안 프로토콜 식별자( Security Protocol Identifier )로 구성된다. 공유가 허용된다면 정책값( Any 프로토콜 )을 사용하여 SA를 작성하며 TCP와 UDP 모두 동일한 SA를 사용한다. 이 3개를 트리플이라고 한다.

보안 매개변수 색인은 SA에 할당된 32비트짜리 비트열( 송신측이 AH 헤더 또는 ESP 헤더에 넣는 SPI 값 )이다. SA를 유일하게 식별하는 ID로 사용된다. 목적지 IP 주소는 SA 종단인 최종 목적지 네트워크( 라우터, 침입탐지시스템 등 ) 또는 호스트 IP 주소가 될 수 있다. SA가 수립된 장비의 주소라고 생각하면 된다. 각 목적지 주소마다 SA가 유일하다. 보안 프로토콜 식별자는 이 연관이 AH를 위한 것인지. ESP를 위한 것인지를 지정한다. 만약 장비에서 둘 다 사용한다면, 각각 별도의 SA를 이용해야 한다.

* 보안 연관 데이터베이스

장비의 보안 연관 보안 은 연관 데이터베이스( SAD, Security Association Database )에 저장된다. SAD는 IPsec에서 각 SA에 연관된 요소들을 정의한 데이터베이스로 다음의 요소를 갖는다:

* 1. 보안 매개변수 색인( SPI : Security Parameter Index ) : SA를 식별하도록 수신자가 선택한 32bit 값임.
  2. 순서 번호 카운터( Sequence Number Counter ) : 순서 번호 필드를 위해 생성되는 32bit 값임.
  3. 순서 계수기 오버플로우( Sequence Counter Overflow ) : 순서 번호 카운터의 오버플로우에 대한 유무를 가리키는 플래그임.
  4. 재전송 방지 윈도우( Anti-Replay Window ) : 패킷의 재전송 여부를 판별함.
  5. AH 정보( AH Information ) : AH와 함께 사용되는 인증 알고리즘, 키, 키 사용 주기 그리고 관련 매개변수의 정보를 포함함.
  6. ESP 정보( ESP Information ) : ESP와 함께 사용되는 암호화 및 인증 알고리즘, 키, 초기값, 키 사용 주기 그리고 관련 매개변수의 정보를 포함함.
  7. 보안 연관 사용주기( Lifetime of this Security Association ) : 하나의 SA가 새로운 SA로 교체/종료되는 시간 간격 또는 바이트 카운트 값
  8. IPSec 프로토콜 모드( IPSec Protocol Mode ) : 터널/전송 모드
  9. 경로 MTU( Path Maximum Transfer Unit )
* 선택자

선택자( Selector )는 IPSec이 각 SA가 자신이 적용될 데이터그램을 선택하기 위한 규칙 모음을 정의할 수 있도록 하는데, 이 규칙 모음의 각각을 선택자라고 한다. 이는 장비가 특정 데이터그램에 어떤 보안 정책이나 SA를 사용할 지를 결정할 수 있도록 해준다. 구성 항목으로는 로컬 IP, 포트, 원격 IP, 포트, 이름, 데이터 민감성 수준, 전송 계층 프로토콜이 있다.

* 보안 연관 트리플과 보안 인자 색인

SA는 단방향으로 각각 인바운드 또는 아웃바운드 트래픽 중 하나만을 담당한다. 이는 장비 A에서 장비 B로의 보안 수준과 장비 B에서 장비 A로의 보안 수준을 서로 다르게 설정할 수 있도록 해준다. SA는 트리플이라 불리는 세 개의 인자 모음으로 정의된다.

이처럼 두 보안 프로토콜( AH와 ESP )는 “SA”, “보안 정책”, “SA와 정책을 제어하는 데이터베이스( 복잡하지만 관리가 중요 )” 에 의존한다. SA는 수동으로 구성하거나 IKE라는 프로토콜을 이용하여 자동으로 구성할 수 있다.

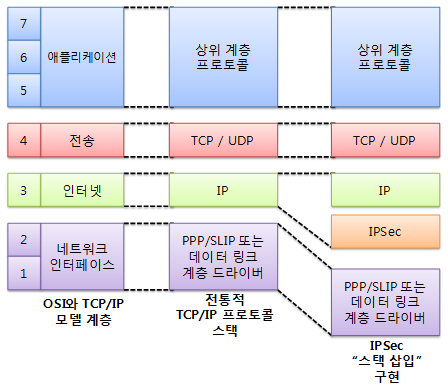
1. **IPSec 구조와 구현 방법**

IPSec을 IP에 어떻게 배치할 것인지에 대한 문제를 논의하겠다. 고려사항으로 IP Version, Application 요구 사항 등이 있으며 그 외에도 IPSec이 TCP / IP 전체 계층 아키텍처를 어떻게 수정할 수 있는 지에 대해서도 생각해야 한다.

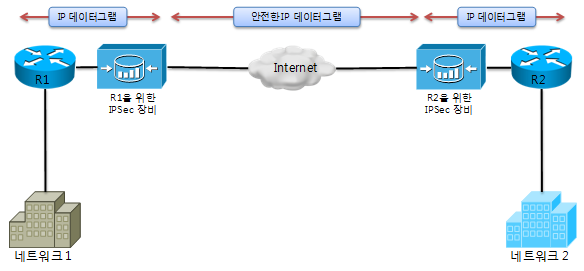
네트워크 설계상 고려사항에 있어 모든 호스트 장비에 IPSec을 구성하는 종단 호스트 구현 방법이 있다. 이 방식의 경우 유연성과 보안성이 극대화되며, 네트워크 상의 두 장치간에 엔드-투-엔드 보안을 가능하게 한다. 그러나 일반적인 네트워크에는 수많은 호스트가 있으므로 라우터에 IPSec을 구현하는 것보다 훨씬 많은 작업이 필요합니다.

라우터 구현은 수백 또는 수천 개의 클라이언트 간 IPSec 구성이 아닌 클라이언트와 연결된 라우터간만 IPSec을 구성하는 방법으로 종단 호스트 구현 방법에 비해 매우 효과적이다. 이 방법의 경우 IPSec을 구현하는 라우터쌍 간에만 보호 기능을 제공하지만 이는 VPN과 같은 응용 프로그램에 충분할 수 있다. 라우터는 데이터그램이 조직 외부로 나가는 경로의 일부분만 보호하기 때문에 라우터와 로컬 호스트간의 연결을 보호하지 못할 수 있다( 아니면 다른 방법으로 보호할 수도 있다 ).

또한 기존의 TCP/IP 프로토콜 스택과 IPSec을 결합하는 문제도 있다. 먼저 통합 구조( Integrated Architecture )는 IPSec의 프로토콜과 기능을 IP 자체에 통합하는 방법을 의미하며, 추가 하드웨어도 필요하지 않고, IPSec 보안모드와 기능이 일반 IP처럼 쉽게 제공될 수 있어 가장 효과적이다. 이 때 IPv6의 경우 IPSec을 지원하도록 설계되었으므로 구조상 IPSec을 쉽게 통합할 수 있지만, IPv4의 경우 각 장비의 IP 구현을 모두 변경해야 하기 때문에 비효율적이라는 단점이 있다.



스택 삽입( Bump In The Stack ) 구조는 스택 삽입 구조에서 IP와 데이터 링크 계층 사이에서 별도의 구조 계층으로 존재한다. 이 구조의 장점은 IPSec 기능이 IP와 분리되어 있기 때문에 어떤 IP 장비에도 IPSec을 적용할 수 있다. 단점은 통합 구조에 비해 TCP/IP 스택에서 해야할 일이 늘어난다는 것이다. 일반적으로 IPv4 호스트에 쓰인다.

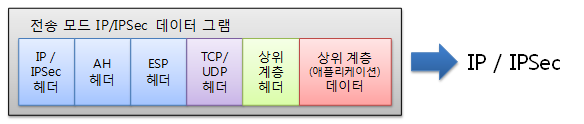


라인 삽입( Bump In The Wire ) 구조는 IPSec 서비스를 제공하는 하드웨어 장비를 추가한다. 예를 들어 두개의 사이트가 있는 회사가 있다고 가정하고, 각 라우터에는 IPSec 기능이 없다고 하자. 이 때 라우터와 인터넷 사이에 특별한 IPSec 장치를 삽일할 수 있고, 이 장치는 보내는 데이터그램을 가로채서 IPSec 보호를 추가한 다음 들어오는 데이터그램에서 제거한다. 이 구조는 비 IPSec 라우터를 개조하여 보안상의 이점을 제공할 수 있으나 네트워크가 복잡해지고 구현 비용이 비싸다는 단점이 있다. 부가적으로 설명하자면 BITS와 BITW는 다른 것처럼 보여도 실제로는 똑같다. BITS의 경우 기존 IP 데이터그램에 보안을 추가하는 추가 소프트웨어 계층이 있다. BITW에서 별개의 하드웨어 장치가 이와 동일한 작업을 수행할 뿐이다. 따라서 결과는 동일하며 IPSec 모드 선택에 대한 영향도 마찬가지이다.

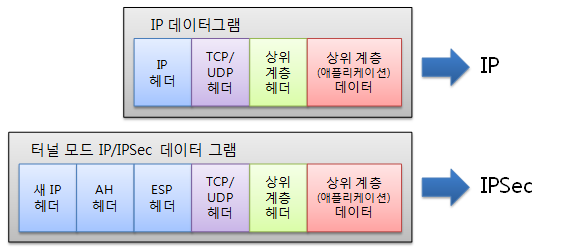
IPSec의 아키텍처 선택은 아래에서 후술할 2가지 IPSec 모드 중 어느 것을 사용할 수 있는지에 중요한 영향을 미친다.

1. **IPSec 운용 모드 : 전송과 터널**

IPSec을 구성할 때 구조에 따라 “전송모드”와 “터널모드”로 동작모드가 나눠질 수 있다. 이 두 가지 모드는 AH와 ESP의 기능과 밀접히 연관되어 있고, 이들 프로토콜은 IP를 보호하기 위해 보안 정보를 포함하는 헤더를 데이터그램에 추가한다. IPSec 모드의 선택은 IP 데이터그램의 어떤 부분이 보호되는지와 헤더가 어떻게 배열되는지에 영향을 준다.



먼저 전송 모드는 전송 계층( Transport Layer )에서 IP로 내려온 메시지를 보호한다. 이 메시지는 AH 또는 AH + ESP 조합에 의해 처리되며 적절한 헤더가 전송헤더( UDP/TCP ) 앞에 붙고, 다시 그 앞에 IP 헤더가 붙는다. 전송 계층 메시지에 대한 원본 IP 패키징이 수행될 때 AH / ESP가 적용되어야 하기 때문에 IPSec이 IP에 통합( Integrated Architecture )되어야 한다. 따라서 전송모드는 IPSec을 직접 실행하는 호스트에서 종단간 보안을 필요로하는 구현에 많이 쓰인다.



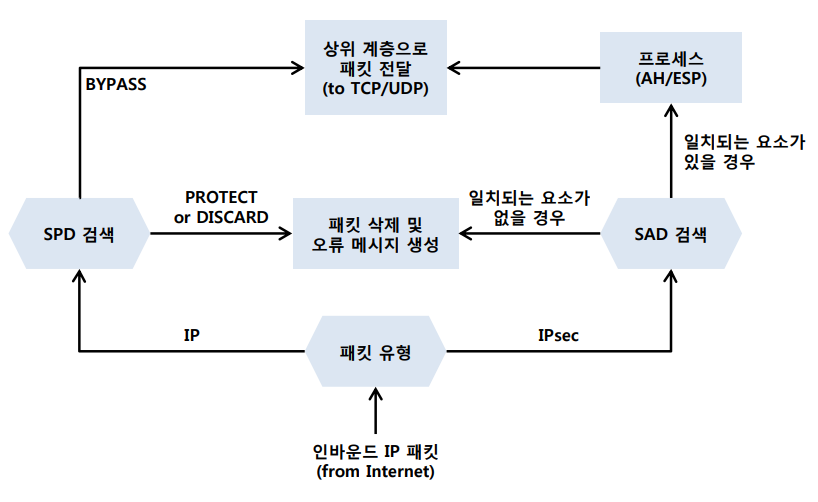
터널 모드는 IP 헤더가 이미 추가된 완전히 캡슐화된 IP 데이터그램을 보호하는데 쓰인다. IPSec 헤더는 원본 IP헤더 앞에 붙으며 새로운 IP 헤더가 다시 이 IPSec 헤더 앞에 붙는다. 즉, 터널모드는 IP와 IPSec 조합을 이용하여 IP를 캡슐화한다. 따라서 터널모드는 상위계층 메시지를 처리하고 자신의 헤더에 붙인 후에 IPSec을 적용하는 BITS와 BITW 구현과 대응된다. 터널모드는 인터넷과 같은 안전하지 않은 네트워크를 통해 IP 데이터그램을 터널링하는 VPN에 널리 쓰인다.

둘을 비교하자면, 터널 모드는 기존 IP 패킷을 감싸 처리한 후 새 IP 헤더를 추가하는 방식으로 원본 IP 데이터그램 전체를 보호하나 전송모드는 원래의 IP 헤더는 대부분 그대로 이용하고, 나머지 데이터 부분만 보호하는 방식이므로 그렇지 않다고 할 수 있겠다. 이러한 특징 때문에 전송 모드에서는 엔드-투-엔드 간에 주로 사용하고, 터널 모드는 라우터간 또는 호스트와 라우터간 또는 게이트웨이 간에 주로 사용한다.

1. **IP 패킷 처리**

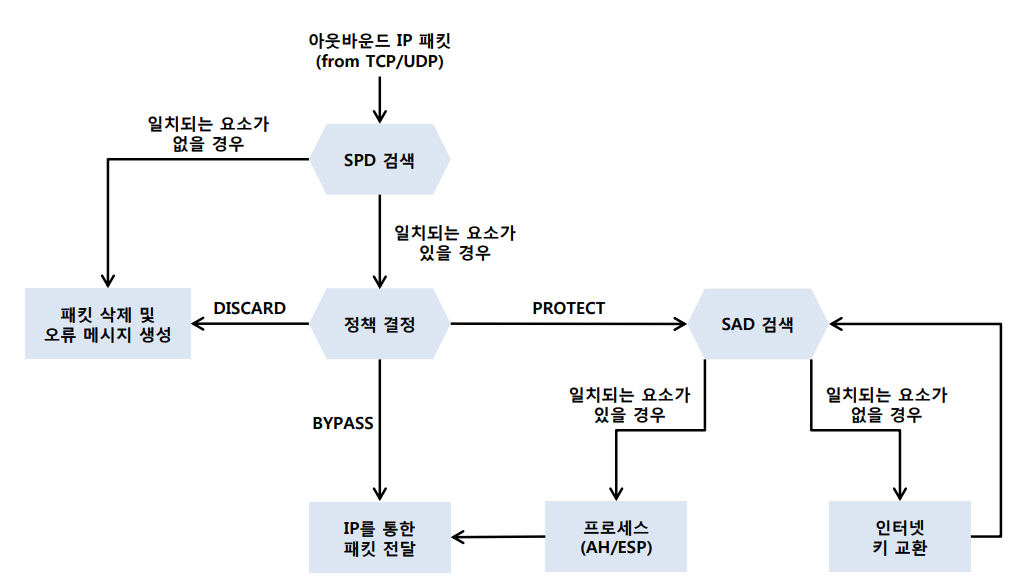
* 인바운드 패킷 처리

인터넷으로부터 인바운트 IP 패킷이 패킷 유형이 IP인지 IPSec인지 확인한다. IP일 경우 SPD에서 검색을 하여 결정된 정책이 BYPASS일 경우 상위 계층( TCP/UDP )으로 패킷을 전달하고, PROJECT 또는 DISCARD일 경우 패킷 삭제 및 오류 메시지를 생성한다IPSec일 경우 SAD에서 검색하여 일치되는 요소가 있을 경우, AH/ESP에 대한 프로세스를 수행하고 상위 계층으로 패킷을 전달하고, 일치되는 요소가 없을 경우 패킷을 삭제하고 오류 메시지를 생성한다.



* 아웃바운드 패킷 처리

아웃바운드 IP 패킷이 TCP/UDP로부터 내려오면 SPD에서 검색하여 일치하는 정책이 없을 경우 패킷을 삭제하고 오류 메시지를 생성하고 일치되는 정책이 있다면 가장 알맞은 높은 우선순위의 첫 번째 정책을 선택한다. 그 정책이 DISCARD이면 패킷 삭제 및 오류 메시지를 생성하고 PROJECT이면 SAD를 검색하여 일치하는 요소가 있다면 AH/ESP 프로세스를 수행하고 IP를 통한 패킷 전달을 수행한다. 없다면, IKE를 수행하여 SAD에 추가한 다음 다시 검색하여 IP를 통한 패킷 전달을 수행한다. 정책이 BYPASS라면 IP를 통해 곧바로 패킷을 전달한다.



1. 레 퍼 런 스

* 네이버 블로그 - IPSec 이론 1 : <https://blog.naver.com/eqelizer/20136062915>
* 네이버 블로그 – IPSec에 관한 VPN :

<https://m.blog.naver.com/PostView.nhn?blogId=routerkow&logNo=140059293811&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F>

* 네이버 블로그 - IPSec 보안 정책 :

<http://blog.naver.com/PostView.nhn?blogId=rpg2003a&logNo=221173319361&categoryNo=43&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=search>

* 위키피디아 한글 – IPSec : <https://ko.wikipedia.org/wiki/IPsec>
* 위키피디아 영문 – IPSec : <https://en.wikipedia.org/wiki/IPsec>
* 티스토리 – IPsec : <http://wooguy-tcpip.blogspot.com/2014/01/ip-securityipsec.html>
* 공대위키 – IPsec : <http://itwiki.kr/w/IPSec>
* The TCP/IP Guide – IPSec Overview :

<http://www.tcpipguide.com/free/t_IPSecOverviewHistoryandStandards.htm>

* The TCP/IP Guide – IPSec ESP :

<http://www.tcpipguide.com/free/t_IPSecEncapsulatingSecurityPayloadESP.htm>

* TechTarget – IPSec architectures and implementation methods

<https://searchdatacenter.techtarget.com/feature/IPsec-architectures-and-implementation-methods>

* 정보통신기술용어해설 – IPSec :

<http://www.ktword.co.kr/abbr_view.php?m_temp1=2029>

* 정보통신기술용어해설 – SA :

<http://www.ktword.co.kr/abbr_view.php?m_temp1=2124>

* 정보통신기술용어해설 – IKE :

<http://www.ktword.co.kr/abbr_view.php?m_temp1=2284>

* Petri – What are IPSec Policies? : <https://www.petri.com/what_are_ipsec_policies>
* treck wiki – SPD & SAD :

<http://wiki.treck.com/Security_Policy_Database_(SPD)_and_Security_Association_Database_(SAD)>

* treck wiki – IPSec / IKE 프로토콜 개요 :

<http://wiki.treck.com/IPsec/IKE_Protocol_Overview>

* networksorcery – AH, 인증헤더 :

<http://www.networksorcery.com/enp/protocol/ah.htm>

* IBM 지식 센터 – 보안 페이로드 캡슐화 :

<https://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_72/rzaja/rzajaesp.htm>

* krnet – 인터넷보안 – IPSec :

<http://www.krnet.or.kr/board/data/dprogram/690/T7-1%C0%CC%B1%A4%BC%F6.pdf>

* Cisco – OAKLEY와 SKEME의 차이점 :

<https://learningnetwork.cisco.com/thread/21468>

* StackExchange – 키 관리 – IPSec이 IKE나 ISAKMP를 사용합니까?

<https://security.stackexchange.com/questions/35872/does-ipsec-use-ike-or-isakmp>

* 안철수연구소 – VPN 4- ISAKMP 프로토콜 :

<http://image.ahnlab.com/file_upload/tech/VPN(4)_ISAKMP.pdf>

* MyKnowledge – ISAKMP 교환 타입 :

<http://myknowledge.kr/127?category=292885>

* 안동대학교 – 인터넷 보안연계 키관리 프로토콜 :

<http://contents.kocw.or.kr/document/lec/2011_2/andong/Network/09.pdf>

* IPSec은 Site-to-Site 간의 설정은 적합하지만, Teleworker는 IPSec을 적용하여 활동하기 힘들다. 하지만 SSL의 경우 웹 브라우저만 구동이 가능하면 암호화 및 인증이 가능하기 때문에 Teleworker도 IPSec처럼 인증 및 암호화가 가능하다.

따라서 Teleworker 환경의 경우 IPSec보다는 SSL을 더 많이 사용할 것이다.