2026/09/22, 2026 보안 기초 세미나

암호학과 네트워크 보안

-18장 네트워크층 보안: IPSec-

송 민 희(minhee@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

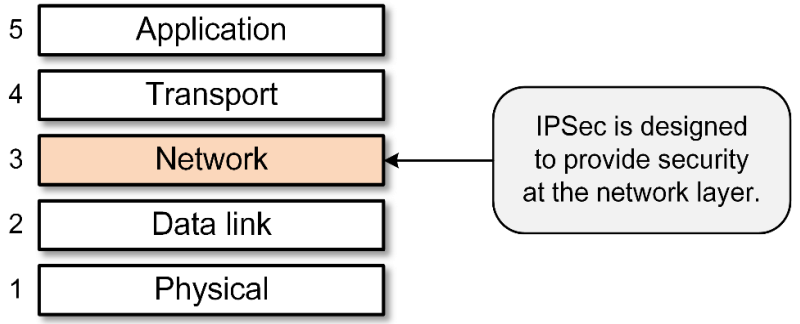
- 개요
- 프로토콜
- 보안 연관 및 정책
- 인터넷 키 교환(IKE, Internet Key Exchange)

목 차

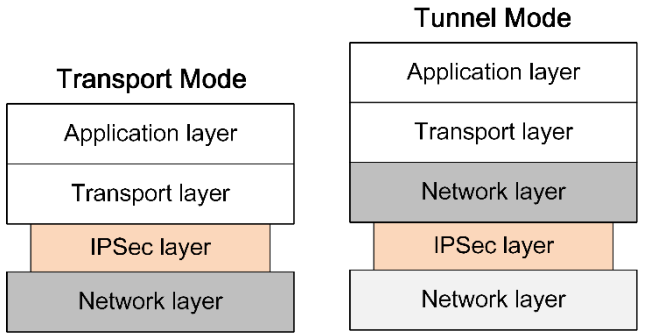
- 개요
- 프로토콜
- 보안 연관 및 정책
- 인터넷 키 교환(IKE, Internet Key Exchange)

개요

- IPSec (Internet Protocol Security) (1/3)
  - 정의
    - 네트워크 계층에서 IP 패킷의 인증, 무결성, 기밀성 등의 보안 서비스를 제공하는 프로토콜 집합
  - 필요성
    - IP (Internet Protocol)는 전송되는 데이터에 대해 암호화를 제공하지 않으므로 패킷의 기밀성을 보장하는 기능 필요
    - IP는 데이터의 변조 여부와 출처를 검증하는 기능을 제공하지 않으므로 패킷의 무결성 및 데이터 출처 인증 기능 필요
  - 모드
    - 전송 모드(Transport Mode)
    - 터널 모드(Tunnel Mode)



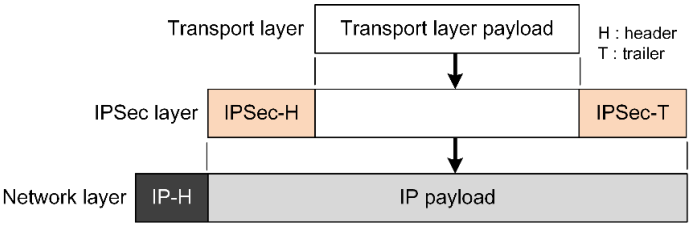
<그림 1> TCP/IP 프로토콜 그룹과 IPSec



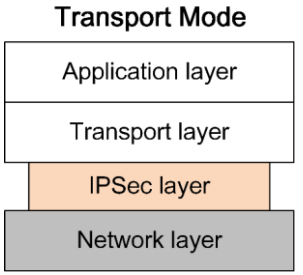
<그림 2> 전송 모드와 터널 모드에서의 IPSec 위치

개요

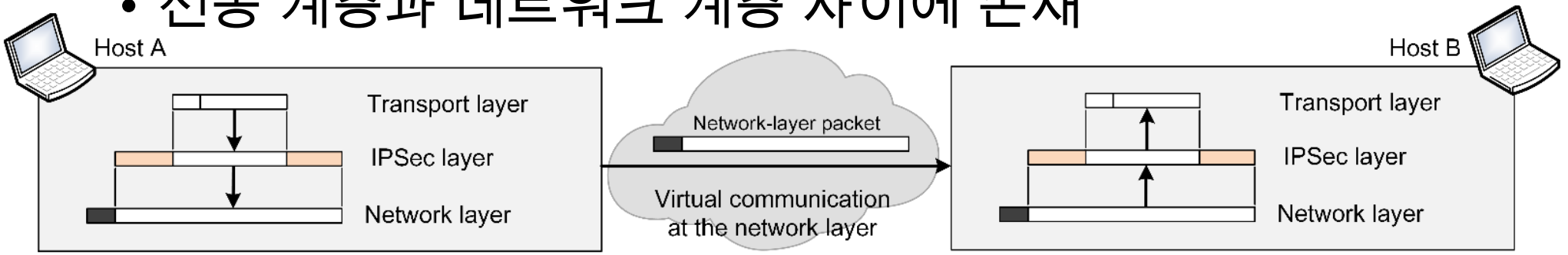
- IPSec (Internet Protocol Security) (2/3)
  - 전송 모드
    - 기존 IP 헤더를 유지하고 상위 계층 데이터를 보호하는 방식
    - 호스트-호스트 간에 데이터 보호를 필요로 할 때 주로 사용
      - 송신 호스트는 전송 계층에서 전달된 페이로드를 보안 처리 적용
      - 수신 호스트는 호스트는 보안 처리된 페이로드를 검증 후 복원
    - 전송 계층과 네트워크 계층 사이에 존재



<그림 3> IPSec 전송 모드



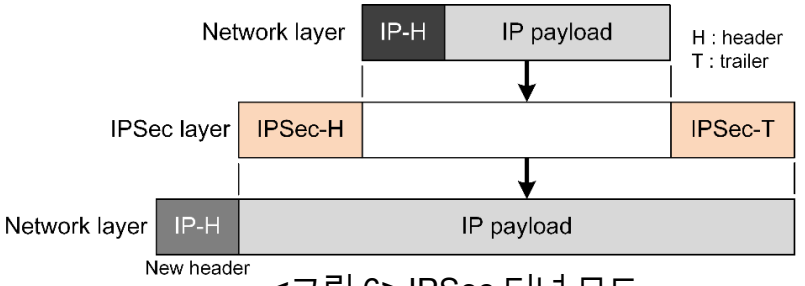
<그림 5> 전송 모드에서의 IPSec 위치



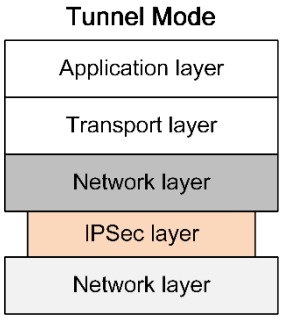
<그림 4> 전송 모드 동작 과정

개요

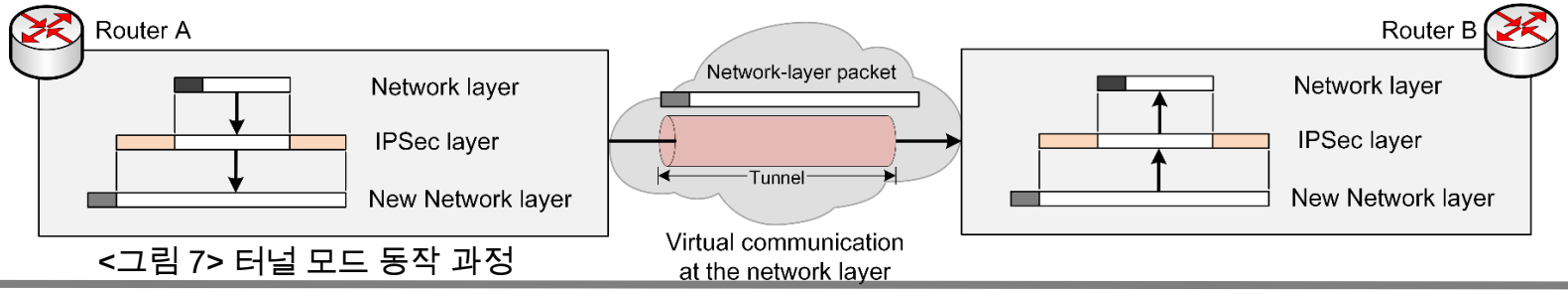
- IPSec (Internet Protocol Security) (3/3)
  - 터널 모드
    - 기존 IP 패킷 전체를 캡슐화하여 보호하고 새로운 IP 헤더를 추가하는 방식
    - 라우터-라우터 혹은 라우터-호스트 간에 데이터 보호를 필요로 할 때 주로 사용
      - 송신 라우터는 원본 IP 패킷 전체에 보안 처리 적용 후 새로운 IP 헤더를 추가
      - 수신 라우터/호스트는 새로운 IP 헤더를 제거하고 원본 IP 패킷을 검증 후 복원
    - 내부 네트워크 계층과 외부 네트워크 계층 사이에 존재



<그림 6> IPSec 터널 모드



<그림 8> 터널 모드에서의 IPSec 위치



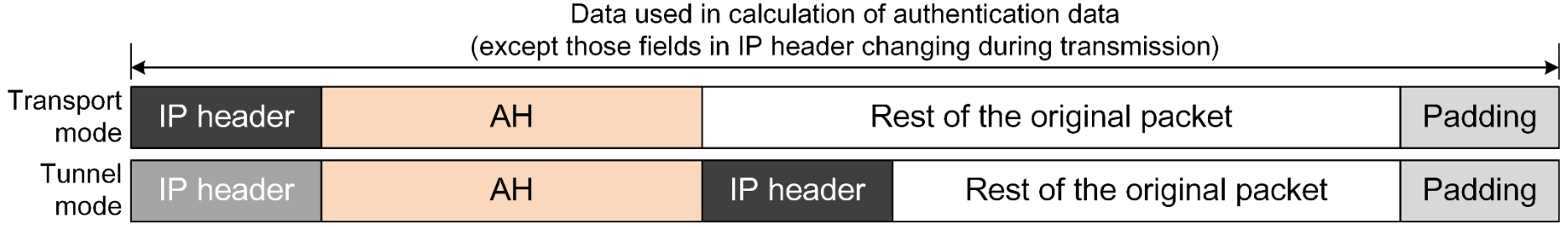
<그림 7> 터널 모드 동작 과정

목 차

- 개요
- 프로토콜
- 보안 연관 및 정책
- 인터넷 키 교환(IKE, Internet Key Exchange)

프로토콜

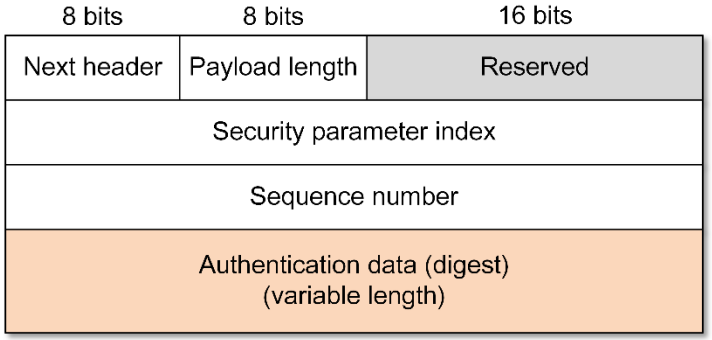
- 인증 헤더(AH, Authentication Header) 프로토콜 (1/4)
  - 정의
    - IP 패킷의 무결성과 송신자 인증을 제공하는 보안 프로토콜
  - 특징
    - 인증 데이터를 생성하기 위해 해시함수와 대칭키를 사용
    - 인증 헤더에 인증 데이터 삽입
    - IP 헤더의 프로토콜 필드 값을 51로 설정
      - 전송 모드는 기존의 IP 헤더를 변경
      - 터널 모드는 새롭게 추가된 IP 헤더를 변경



<그림 9> 인증 헤더(AH)가 추가된 IPSec 패킷 구조

프로토콜

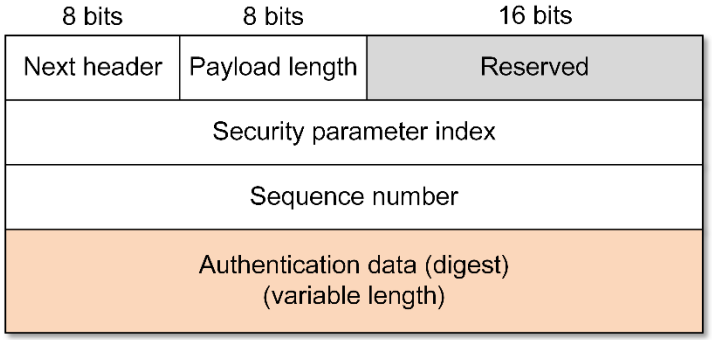
- 인증 헤더 프로토콜 (2/4)
  - 구조 (1/2)
    - 다음 헤더(Next header)
      - IP 데이터그램에 의해 전달되는 페이로드의 유형을 정의하는 8비트 길이의 필드
      - 전송 모드에서는 AH를 추가하기 전의 IP 헤더의 프로토콜 필드와 동일한 값 적용
      - 터널 모드에서는 AH 다음에 오는 데이터의 유형 적용
        - 항상 원본 IP 패킷이 오므로 IPv4(4) 또는 IPv6(41) 적용
    - 페이로드 길이(Payload length)
      - AH의 길이를 4바이트 배수로 정의하는 8비트 길이의 필드
    - 보안 매개변수 색인(SPI, Security Parameter Index)
      - 패킷에 적용할 보안 설정을 식별하는 32비트 길이의 필드
      - 동일한 보안 설정으로 전송되는 모든 패킷에서 동일한 값 사용



<그림 10> 인증 헤더(AH) 구조

프로토콜

- 인증 헤더 프로토콜 (3/4)
  - 구조 (2/2)
    - 순서 번호(Sequence Number)
      - 데이터그램의 순서 정보를 제공하는 32비트 길이의 필드
      - 재전송 공격(Replay Attack)을 방지하기 위해 사용
      - 재전송 패킷에 대해 새로운 순서 번호 부여
      - 순서 번호가 2^{32}에 도달한 이후에는 새로운 연결 필요
    - 인증 데이터(Authentication data)
      - 전송 도중에 변경될 수 있는 필드(TTL)를 제외한 완전한 IP 데이터그램에 해시함수를 적용한 메시지 다이제스트를 가지는 가변길이의 필드

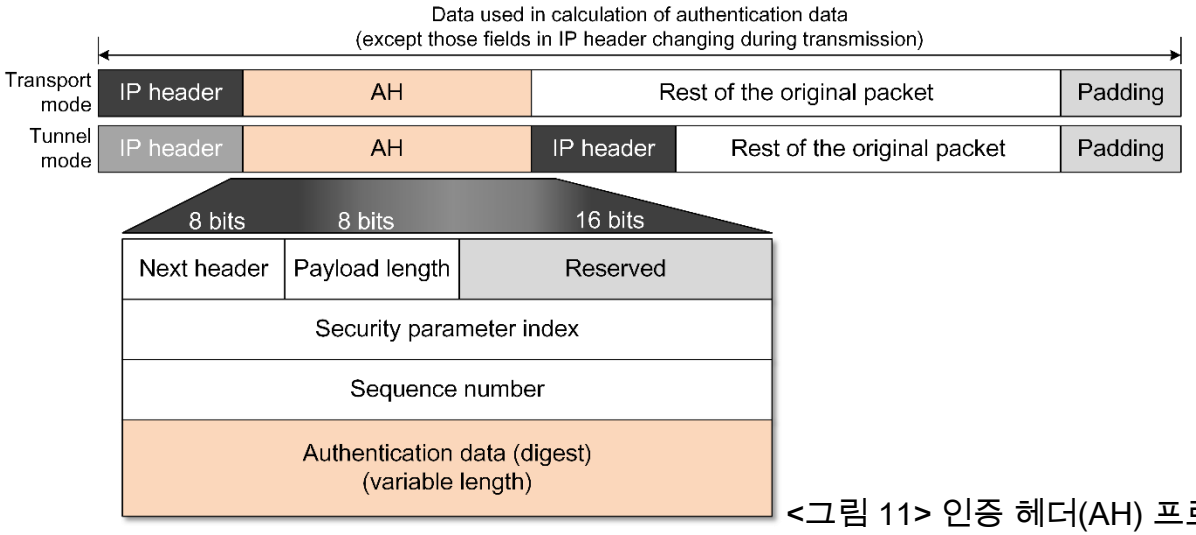


<그림 10> 인증 헤더(AH) 구조

*TTL (Time To Live)
데이터그램이 경유할 수 있는 최대 홉 수를 나타내며
홉을 지날 때마다 1씩 감소하여 0이 되면 데이터그램을 폐기하는 필드

프로토콜

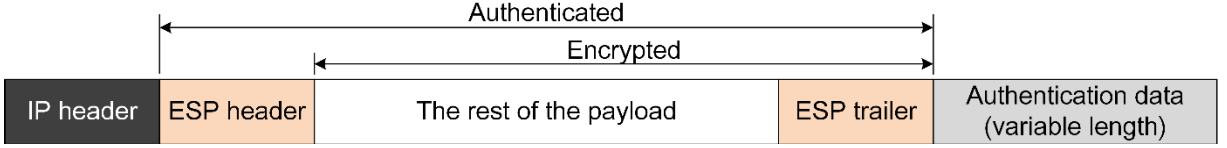
- 인증 헤더 프로토콜 (4/4)
  - 인증 헤더 추가 과정
    1. 0으로 설정된 인증 데이터 필드를 가진 페이로드에 AH 추가
    2. IP 헤더의 프로토콜 필드 값을 51로 바꾼 뒤에 추가
    3. 필요 시 해싱 알고리즘에 대해 전체 길이를 일정하게 만들기 위한 패딩(Padding) 추가
    4. 전체 패킷을 기반으로 해싱하여 인증 데이터 생성
       - 전송 중 값이 변경되는 IP 헤더의 필드는 인증 데이터 계산에서 제외
    5. 인증 데이터를 인증 헤더에 삽입



<그림 11> 인증 헤더(AH) 프로토콜

프로토콜

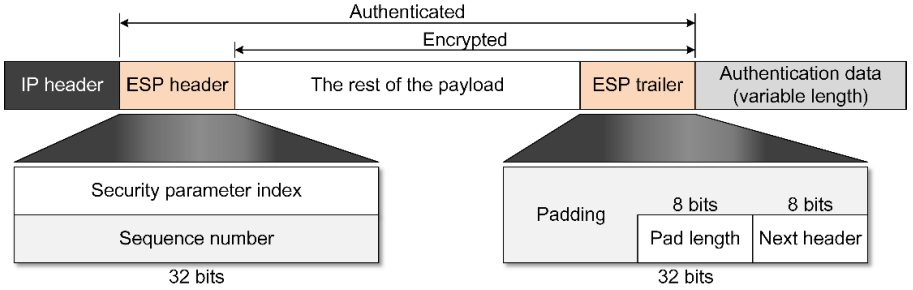
- 보안 페이로드 캡슐화(ESP, Encapsulating Security Payload) 프로토콜 (1/4)
  - 정의
    - IP 패킷의 기밀성과 무결성, 송신자 인증을 제공하는 보안 프로토콜
  - 특징
    - 패킷 암호화를 위해 대칭키 암호 알고리즘을 사용
    - 인증을 사용하는 경우, 인증 데이터를 생성하기 위해 해시함수와 대칭키를 사용
    - ESP 헤더와 ESP 트레일러를 추가하며, 인증 데이터는 트레일러의 뒤에 추가
    - IP 헤더의 프로토콜 필드의 값을 50으로 설정



<그림 12> 보안 페이로드 캡슐화(ESP)가 추가된 IPSec 패킷 구조

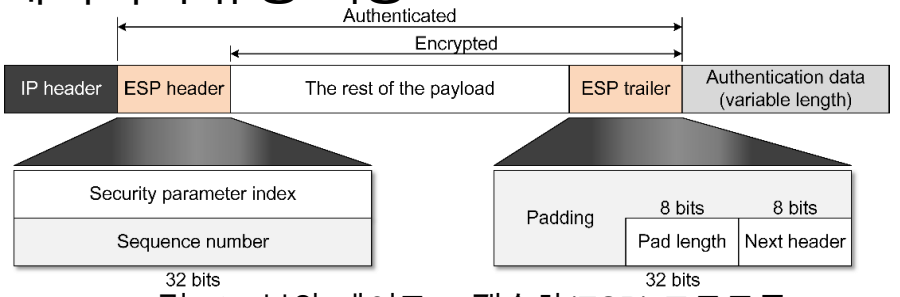
프로토콜

- 보안 페이로드 캡슐화(ESP) 프로토콜 (2/4)
  - 구조 (1/2)
    - 인증 데이터(Authentication Data)
      - ESP 헤더, 페이로드, ESP 트레일러에 키를 사용한 해시함수를 적용한 메시지 다이제스트를 가지는 가변 길이의 필드
        - AH와 달리 IP 헤더는 인증 데이터 계산에서 제외
    - ESP 헤더(ESP Header)
      - 보안 매개변수 색인(SPI, Security Parameter Index)
        - 패킷에 적용할 보안 설정을 식별하는 32비트 길이의 필드
        - 동일한 보안 설정으로 전송되는 모든 패킷에 대해 동일한 값을 사용
      - 순서 번호(Sequence Number)
        - 데이터그램의 순서를 위한 순서 정보를 제공하는 32비트 길이의 필드
        - 재전송 공격을 방지하기 위해 사용
        - 재전송 패킷에 대해 새로운 순서 번호 부여
        - 순서 번호가 2^{32}에 도달한 이후에는 새로운 연결 필요



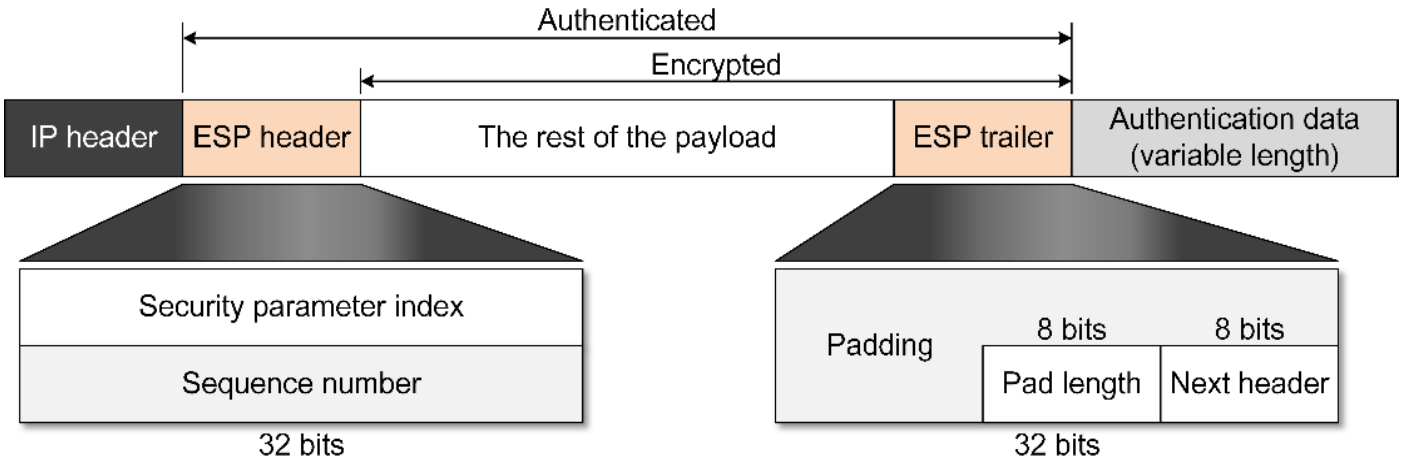
<그림 13> 보안 페이로드 캡슐화(ESP) 프로토콜

- 보안 페이로드 캡슐화(ESP) 프로토콜 (3/4)
  - 구조 (2/2)
    - ESP 트레일러(ESP Trailer)
      - 패딩(Padding)
        - 0으로만 구성되며 최대 255바이트의 길이를 갖는 필드
      - 패드 길이(Pad Length)
        - 패딩 바이트 수를 정의하는 8비트 길이의 필드
        - 0부터 255 사이의 값
      - 다음 헤더(Next Header)
        - IP 데이터그램에 의해 전달되는 페이로드의 유형을 정의하는 8비트 길이의 필드
        - 전송 모드에서는 ESP 추가 전의 IP 헤더의 프로토콜 필드와 동일한 값 적용
        - 터널 모드에서는 ESP로 보호되는 데이터의 유형 적용
          - 항상 원본 IP 패킷이 오므로 IPv4(4) 또는 IPv6(41) 적용



<그림 13> 보안 페이로드 캡슐화(ESP) 프로토콜

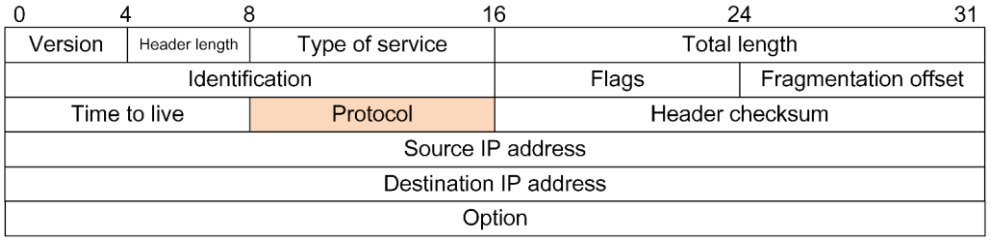
- 보안 페이로드 캡슐화(ESP) 프로토콜 (4/4)
  - 캡슐화 과정
    1. ESP 트레일러를 페이로드 뒤에 추가
    2. 페이로드와 ESP 트레일러 암호화
    3. ESP 헤더를 페이로드 앞에 추가
    4. 인증 사용 시 ESP 헤더, 페이로드, 및 ESP 트레일러로 인증 데이터 생성 후 ESP 트레일러 끝에 추가
    5. IP 헤더의 프로토콜 필드 값을 50으로 바꾼 후 추가



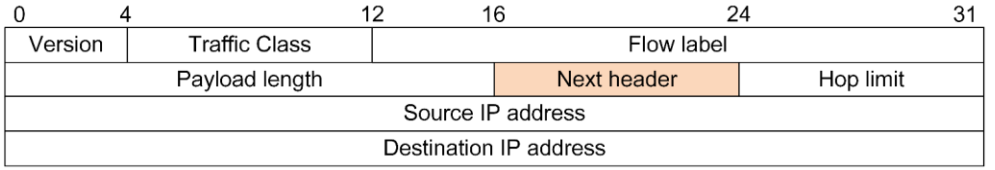
<그림 13> 보안 페이로드 캡슐화(ESP) 프로토콜

프로토콜

- IPv4와 IPv6
  - IPv4에서는 IPSec 적용 시 AH와 ESP를 별도의 프로토콜 헤더로 추가
    - AH(51), ESP(50)을 Protocol 필드 값으로 설정
  - IPv6에서는 IPSec 적용 시 AH와 ESP를 확장 헤더 형태로 처리
    - AH(51), ESP(50)을 Next Header 필드 값으로 설정



<그림 14> IPv4 헤더 구조



<그림 15> IPv6 헤더 구조

프로토콜

- IPSec 서비스 (1/3)
  - 접근 제어(Access Control)
    - 사전에 설정된 보안 정보의 존재 여부를 통한 패킷 접근 제어
    - 수신한 패킷에 적용된 보안 정보가 없는 경우 폐기
  - 메시지 무결성(Message Integrity)
    - 송신자가 생성한 메시지 다이제스트를 수신자가 검증하여 데이터 변조 여부 확인
  - 개체 인증(Entity Authentication)
    - 공유된 키를 기반으로 생성된 인증 데이터를 통한 송신자 신원 확인
  - 기밀성(Confidentiality)
    - 메시지 암호화를 통한 데이터 내용 보호
    - ESP에서만 제공

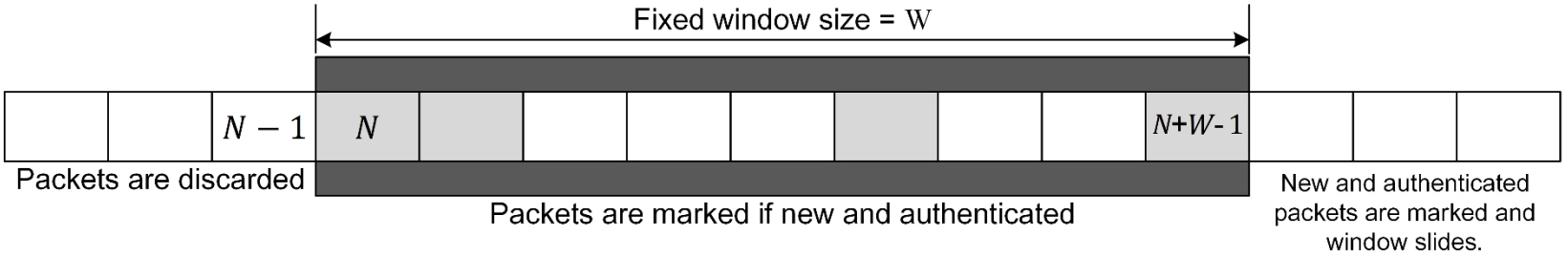
[표 1] IPSec 서비스

Services	AH	ESP
Access control	yes	yes
Message Integrity	yes	yes
Entity authentication	yes	yes
Confidentiality	no	yes
Replay attack protection	yes	yes

- IPSec 서비스 (2/3)
  - 재전송 공격 보호(Replay Attack Protection) (1/2)
    - 수신 측의 재전송 방지 창(Anti-Replay Window)을 통해 중복 패킷 탐지
      - 창 크기를 64로 설정
        - 최소 32로 설정 가능하나 RFC 4303에서 기본적으로 64를 권장
      - 각 패킷에 고유한 32비트 순서 번호 부여
      - 인증된 패킷을 통해 창을 갱신하므로 인증 기능을 사용하지 않는 경우에는 재전송 공격 보호 기능 미적용

[표 1] IPSec 서비스

Services	AH	ESP
Access control	yes	yes
Message Integrity	yes	yes
Entity authentication	yes	yes
Confidentiality	no	yes
Replay attack protection	yes	yes

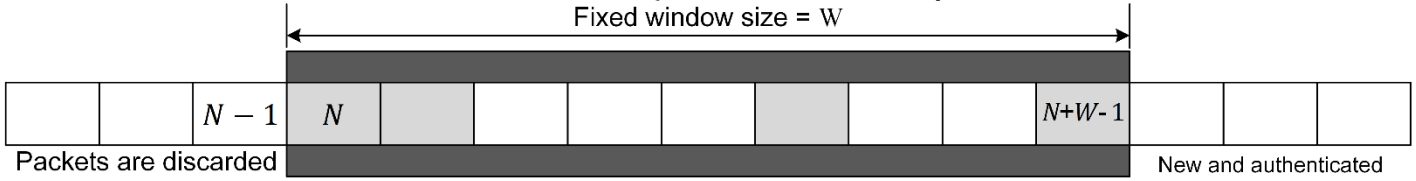


<그림 16> 재전송 방지 창

- IPSec 서비스 (3/3)
  - 재전송 공격 보호(Replay Attack Protection) (2/2)
    - 수신한 패킷의 순서 번호를 통해 상태 판단
      - 순서 번호$< N$
        - 창 범위보다 오래되어 중복 여부를 판단할 수 없는 패킷
        - 패킷 폐기
      - $N \leq$ 순서 번호 $\leq N + W - 1$
        - 현재 창 범위의 패킷
        - 인증 성공 시 패킷 수용 및 해당 순서 번호 표시
        - 이미 표시된 순서 번호면 중복 패킷으로 폐기
      - 순서 번호 $> N + W - 1$
        - 현재 창보다 이후의 새로운 패킷
        - 인증 성공 시 패킷 수용 및 창 오른쪽 이동

[표 1] IPSec 서비스

Services	AH	ESP
Access control	yes	yes
Message Integrity	yes	yes
Entity authentication	yes	yes
Confidentiality	no	yes
Replay attack protection	yes	yes



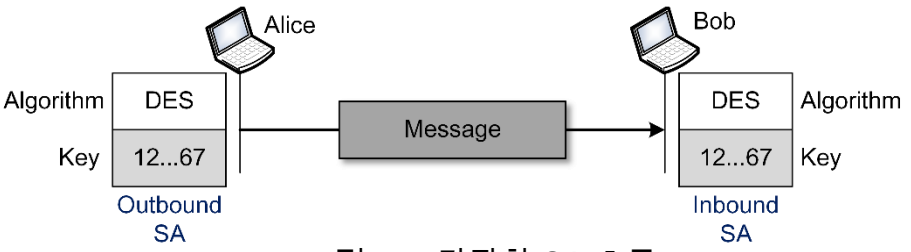
<그림 16> 재전송 방지 창

목 차

- 개요
- 프로토콜
- 보안 연관 및 정책
- 인터넷 키 교환(IKE, Internet Key Exchange)

보안 연관 및 정책

- 보안 연관(SA, Security Association)
  - 정의
    - IPSec에서 송신자와 수신자 간의 통신 보안을 위해 설정되는 논리적 연결



<그림 17> 간단한 SA 흐름

  - 필요성
    - 통신 상대와 방향에 따라 적용할 보안 설정의 지정 필요
  - 특징
    - 단방향으로 정의되므로 양방향 통신을 위해서는 한 쌍의 보안연관이 필요
  - 구분
    - Outbound SA: 송신자가 메시지를 보낼 때 적용되는 보안 연관
    - Inbound SA: 수신자가 메시지를 받을 때 적용되는 보안 연관

보안 연관 및 정책

- 보안 연관 협상(SA Negotiation) (1/2)
  - 정의
    - 통신 당사자가 SA에 사용할 보안 매개변수를 제안하고 선택하여 합의하는 과정
  - 필요성
    - 양쪽이 동일한 알고리즘, 키, 모드를 사용해야 보호된 패킷의 처리 가능
    - 통신 상대마다 지원하는 알고리즘이 다를 수 있으므로 사전 합의 필요
  - 특징
    - 일반적으로 IKE를 통해 자동으로 수행
    - SPI는 협상 과정에서 SA의 수신 측이 결정하여 전달

보안 연관 및 정책

- SA 협상(SA Negotiation) (2/2)
  - 협상 대상
    - 보호 프로토콜(AH/ESP)
    - 암호화·인증 알고리즘
    - IPSec 모드
    - SA 수명
  - 과정
    1. Initiator가 사용 가능한 보안 매개변수 조합을 제안
    2. Responder가 지원 가능한 조합 중 하나를 선택하여 응답
    3. 합의된 매개변수와 생성된 키를 SA로 설정 및 저장

보안 연관 및 정책

- 보안 연관 데이터베이스(SAD, Security Association Database) (1/4)
  - 정의
    - 여러 SA의 정보를 저장하고 관리하는 데이터베이스
  - 필요성
    - 다수의 상대와 통신하면서 생성되는 여러 SA에 대해 체계적인 관리 가능
  - 특징
    - 각 SA의 정보를 저장하는 2차원 표 형태
    - 일반적으로 하나의 개체가 Outbound SAD와 Inbound SAD를 모두 사용
      - Outbound SAD: 자신이 송신하는 패킷에 적용할 SA 정보
      - Inbound SAD: 자신이 수신하는 패킷에 적용할 SA 정보

보안 연관 및 정책

- 보안 연관 데이터베이스(SAD) (2/4)
  - 색인(Index) 구성
    - 보안 매개변수 색인(SPI, Security Parameter Index)
      - 목적지에서 SA를 식별하는 32비트 값
      - SA 협상 과정에서 결정
      - 동일한 Inbound SA에 속하는 패킷에서 동일한 SPI 사용
    - 목적지 주소(DA, Destination Address)
      - 패킷의 목적지 IP 주소
      - 일반적으로 유니캐스트 주소 사용
        - 멀티캐스트 주소도 사용 가능
      - 각 목적지 주소별로 고유한 SA 구분
    - 프로토콜(P, Protocol)
      - AH 또는 ESP의 구분
      - 프로토콜별로 다른 SA 적용

Legend:

SPI: Security Parameter Index	SN: Sequence Number Counter
DA: Destination Address	OF: Sequnce Counter Overflow
AH/ESP: Information for either one	ARW: Anti-Replay Window
P: Protocol	LT: Lifetime
Mode: IPSec Mode Flag	MTU: Path MTU

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							

<그림 18> SAD

보안 연관 및 정책

- 보안 연관 데이터베이스(SAD) (3/4)
  - SA 매개변수(SA Parameter) 구성 (1/2)
    - 순서 번호 카운터(SN, Sequence Number Counter)
      - AH 또는 ESP 헤더의 순서 번호 생성을 위한 32비트 값
    - 순서 카운터 오버플로(OF, Sequence Counter Overflow)
      - 순서 번호가 최댓값(2^{32})을 초과했을 때의 처리 방식을 나타내는 플래그
    - 재전송 방지 창(ARW, Anti-Replay Window)
      - 수신된 AH 또는 ESP 패킷의 재전송 여부 탐지
    - AH/ESP 정보(AH/ESP Information)
      - AH: 인증 알고리즘, 키, 키 수명 등
      - ESP: 암호화 알고리즘, 인증 알고리즘, 키, 키 수명, 초기화 벡터 등

Legend:

SPI: Security Parameter Index	SN: Sequence Number Counter
DA: Destination Address	OF: Sequnce Counter Overflow
AH/ESP: Information for either one	ARW: Anti-Replay Window
P: Protocol	LT: Lifetime
Mode: IPSec Mode Flag	MTU: Path MTU

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							

<그림 18> SAD

보안 연관 및 정책

- 보안 연관 데이터베이스(SAD) (4/4)
  - SA 매개변수(SA Parameter) 구성 (2/2)
    - SA 수명(LT, SA Lifetime)
      - 해당 SA의 유효 기간
    - IPSec 모드(IPSec Mode)
      - 전송 모드 또는 터널 모드의 지정
    - 경로 MTU(MTU, Path Maximum Transmission Unit)
      - 패킷 단편화와 관련된 경로상의 최대 전송 단위

Legend:

SPI: Security Parameter Index	SN: Sequence Number Counter
DA: Destination Address	OF: Sequnce Counter Overflow
AH/ESP: Information for either one	ARW: Anti-Replay Window
P: Protocol	LT: Lifetime
Mode: IPSec Mode Flag	MTU: Path MTU

Index	SN	OF	ARW	AH/ESP	LT	Mode	MTU
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							
< SPI, DA, P >							

<그림 18> SAD

보안 연관 및 정책

- 보안 정책(SP, Security Policy)
  - 정의
    - IPSec을 수행하는 호스트나 게이트웨이에서 송·수신되는 IP 패킷에 적용할 보안 처리 방식을 정의한 규칙
  - 필요성
    - 모든 패킷에 동일한 보안 처리를 적용할 수 없으므로 패킷별 처리 방식에 대한 기준 필요
    - 보호가 필요한 패킷과 그렇지 않은 패킷을 구분하여 적절한 보안 수준 적용
  - 특징
    - 정책에 따라 패킷의 보안 처리 방식을 폐기(Discard), 우회(Bypass), 적용(Protect)으로 결정
      - 폐기(Discard): 패킷 폐기
      - 우회(Bypass): IPSec 처리 없이 전송
      - 적용(Protect): IPSec 처리

보안 연관 및 정책

- 보안 정책 데이터베이스(SPD, Security Policy Database) (1/3)
  - 정의
    - 여러 SP를 저장하고 관리하는 데이터베이스
  - 필요성
    - 패킷의 주소, 프로토콜, 포트 등에 따라 요구되는 여러 SP에 대해 체계적인 관리 가능
  - 특징
    - 각 SP를 하나의 엔트리로 저장하는 2차원 표 형태
    - 일반적으로 하나의 개체가 Outbound SPD와 Inbound SPD를 모두 사용
      - Outbound SPD: 자신이 송신하는 패킷에 적용할 SP 정보
      - Inbound SPD: 자신이 수신하는 패킷에 적용할 SP 정보

보안 연관 및 정책

- 보안 정책 데이터베이스(SPD) (2/3)
  - 색인(Index) 구성 (1/2)
    - 발신지 주소(Source Address)
      - 패킷을 전송하는 호스트의 주소
      - 유니캐스트, 멀티캐스트 또는 와일드카드 주소 사용 가능
    - 목적지 주소(Destination Address)
      - 패킷을 수신하는 호스트의 주소
      - 유니캐스트, 멀티캐스트 또는 와일드카드 주소 사용 가능
    - 이름(Name)
      - 통신 개체를 식별하기 위한 이름
      - 일반적으로 도메인 이름 사용

Index	Policy
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	

Legend:

SA: Source Address	SPort: Source Port
DA: Destination Address	DPort: Destination Port
P: Protocol	

<그림 19> SPD

보안 연관 및 정책

- 보안 정책 데이터베이스(SPD) (3/3)
  - 색인(Index) 구성 (2/2)
    - 프로토콜(Protocol)
      - 상위 계층 프로토콜 구분
    - 발신지 포트(Source Port)
      - 발신지 호스트에서 실행되는 프로세스의 포트 번호
    - 목적지 포트(Destination Port)
      - 목적지 호스트에서 실행되는 프로세스의 포트 번호

Index	Policy
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	
< SA, DA, Name, P, SPort, DPort >	

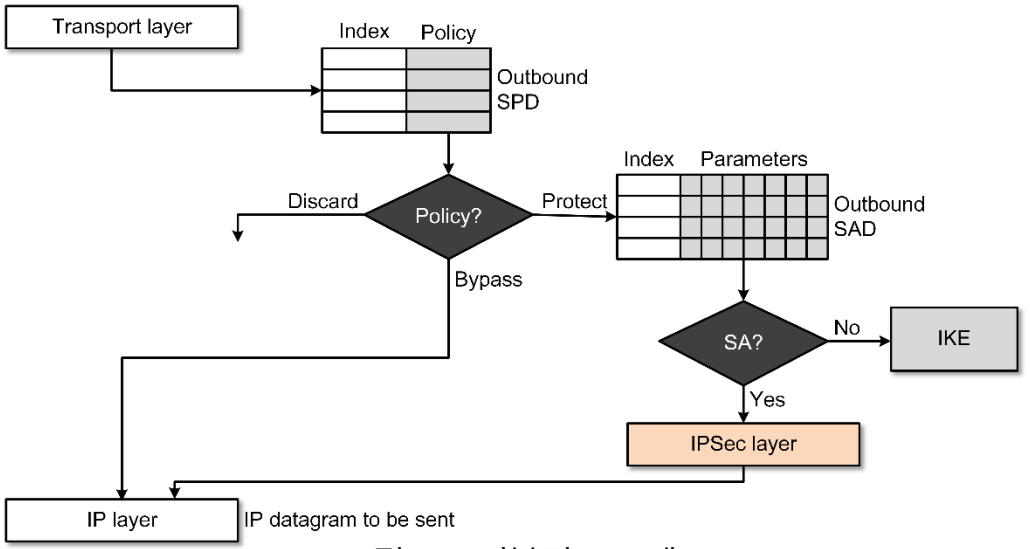
<그림 19> SPD

Legend:

SA: Source Address
DA: Destination Address
P: Protocol
SPort: Source Port
DPort: Destination Port

보안 연관 및 정책

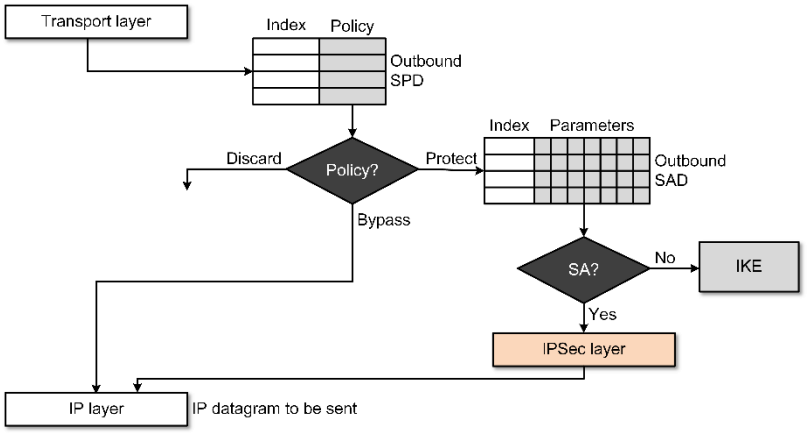
- 패킷 처리 과정 (1/5)
  - 외부적(Outbound) 프로세스 (1/2)
    - 개요
      - 송신 패킷에 적용할 SP를 확인하고, 필요 시 적절한 SA를 적용하는 과정
    - 처리 과정 (1/2)
      1. 패킷의 6개 색인을 이용하여 Outbound SPD에서 SP 탐색



<그림 20> 외부적 프로세스

보안 연관 및 정책

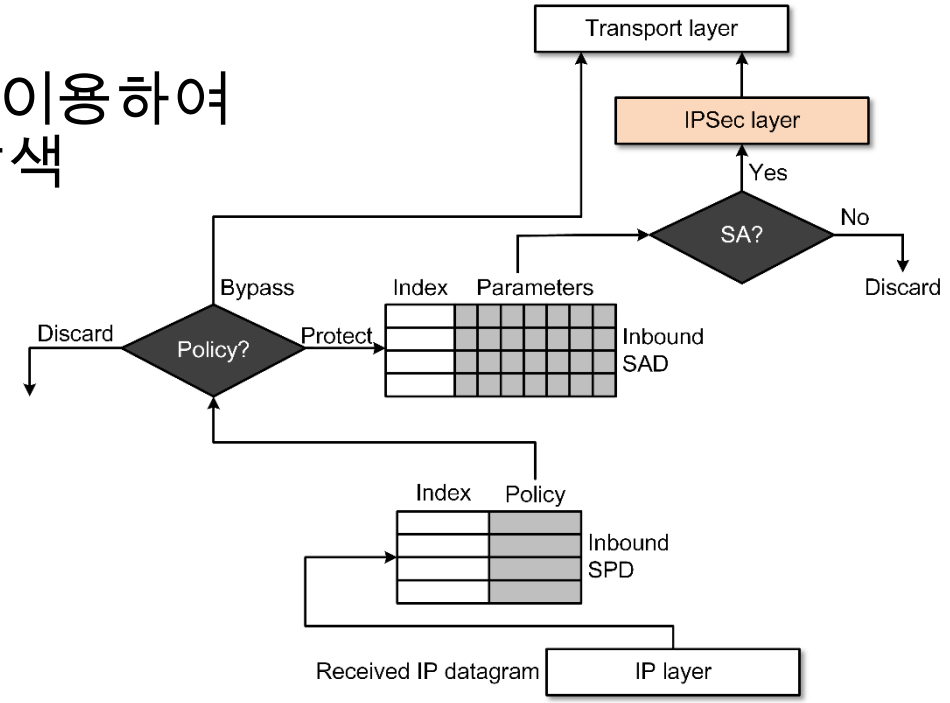
- 패킷 처리 과정 (2/5)
  - 외부적(Outbound) 프로세스 (2/2)
    - 처리 과정 (2/2)
      2. 탐색된 SP에 따라 처리 방식 결정
         - 폐기(Discard)
           - 해당 색인 조건의 패킷은 송신할 수 없도록 정의되어 있으므로 패킷 폐기
         - 우회(Bypass)
           - IPSec 처리 없이 전송하도록 정의되어 있으므로 IPSec 보안 헤더 적용 없이 전송
         - 적용(Protect)
           - 해당 패킷에 IPSec을 적용하도록 정의되어 있으므로 IPSec 적용 수행
      3. Outbound SAD에서 적용할 SA 탐색
         - 기존 SA 존재 시 Outbound SAD에서 해당 SA 선택 후 AH 또는 ESP 적용
         - 기존 SA 부재 시 IKE를 통해 필요한 SA 생성 후 AH 또는 ESP 적용
      4. IPSec 처리 후 패킷 전송



<그림 20> 외부적 프로세스

보안 연관 및 정책

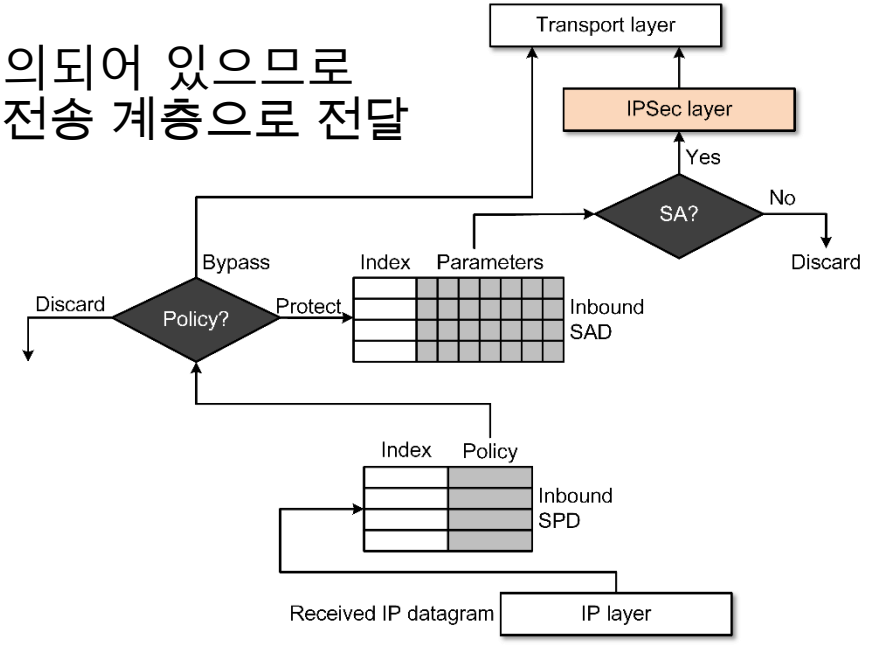
- 패킷 처리 과정 (3/5)
  - 내부적(Inbound) 프로세스 (1/3)
    - 개요
      - 수신 패킷의 보안 정책과 SA를 확인하여 적절한 보안 처리 후 상위 계층으로 전달하는 과정
    - 처리 과정 (1/3)
      1. 수신 패킷의 6개 색인을 이용하여 Inbound SPD에서 SP 탐색



<그림 21> 내부적 프로세스

보안 연관 및 정책

- 패킷 처리 과정 (4/5)
  - 내부적(Inbound) 프로세스 (2/3)
    - 처리 과정 (2/3)
      2. 탐색된 SP에 따라 처리 방식 결정
         - 폐기(Discard)
           - 해당 패킷이 폐기 대상으로 정의되어 있으므로 패킷 폐기
         - 우회(Bypass)
           - IPSec 처리 없이 전송하도록 정의되어 있으므로 AH/ESP 헤더 정보를 무시하고 전송 계층으로 전달
         - 적용(Protect)
           - 해당 패킷의 보안 헤더를 반드시 처리하도록 정의되어 있으므로 Inbound SAD에서 SA를 선택하여 AH 또는 ESP 처리 및 전송 계층으로 전달

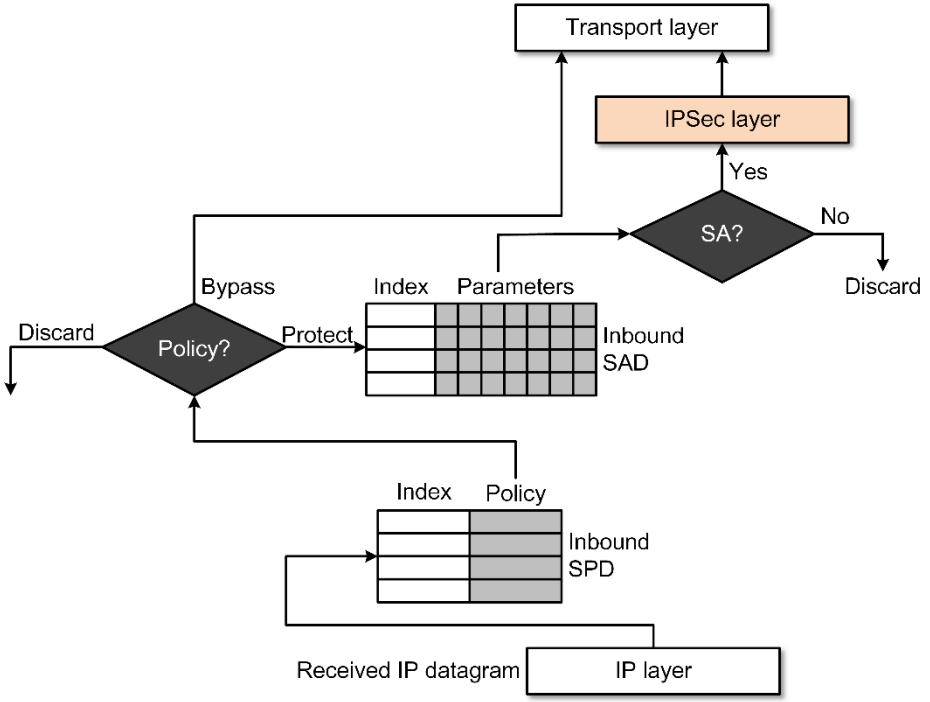


<그림 21> 내부적 프로세스

보안 연관 및 정책

- 패킷 처리 과정 (5/5)
  - 내부적(Inbound) 프로세스 (3/3)
    - 처리 과정 (3/3)

3. Inbound SAD에서 SA 탐색
   - 기존 SA 존재 시 Inbound SAD에서 해당 SA 선택 후 복호화, 인증 또는 두 가지 모두 적용
   - 기존 SA 부재 시 패킷 폐기
4. 보안 처리가 완료되면 AH/ESP 헤더를 제거하고 전송 계층으로 전달



<그림 21> 내부적 프로세스

목 차

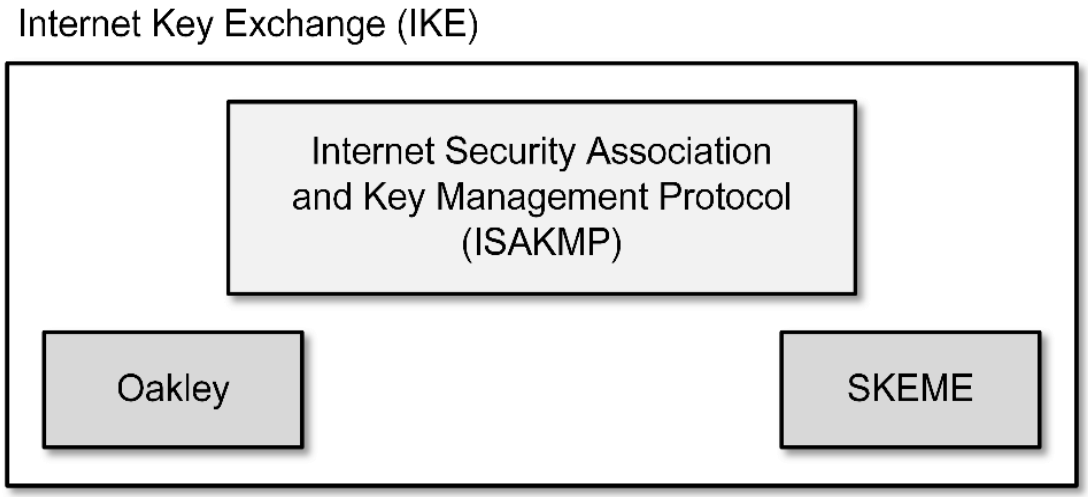
- 개요
- 프로토콜
- 보안 연관 및 정책
- 인터넷 키 교환(IKE, Internet Key Exchange)

인터넷 키 교환

- 정의
  - IPSec에서 내부적 및 외부적 보안 연관과 키를 설정하기 위한 프로토콜
- 필요성
  - 통신마다 필요한 SA와 암호를 사전에 수동으로 설정 불가
  - 기존 SA가 없는 경우 새로운 보안 연결 설정 필요
- 특징
  - SA와 키 설정 과정의 자동화
  - 개선된 Diffie-Hellman 키 교환 방식 사용
  - 2단계 구조를 통한 IKE SA와 IPSec SA의 순차적 설정

인터넷 키 교환(IKE)

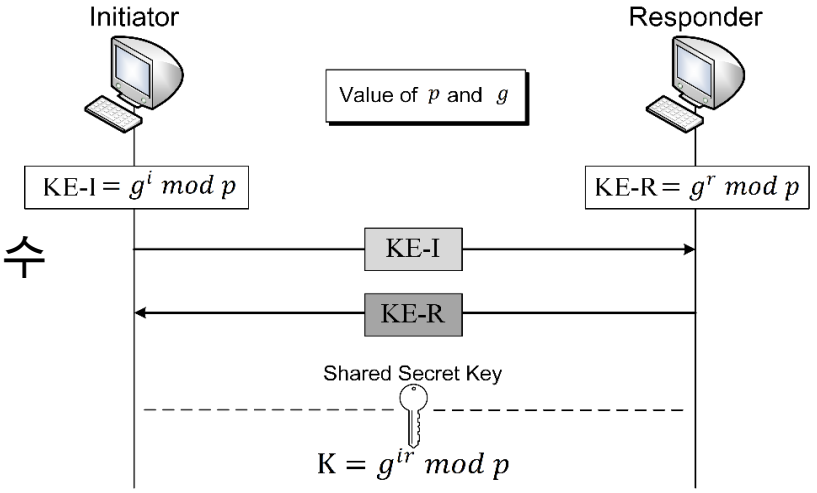
- 구성
  - Oakley 프로토콜
  - SKEME (Secure Key Exchange Mechanism)
  - ISAKMP (Internet Security Association and Key Management Protocol)



<그림 22> IKE 구성요소

인터넷 키 교환(IKE)

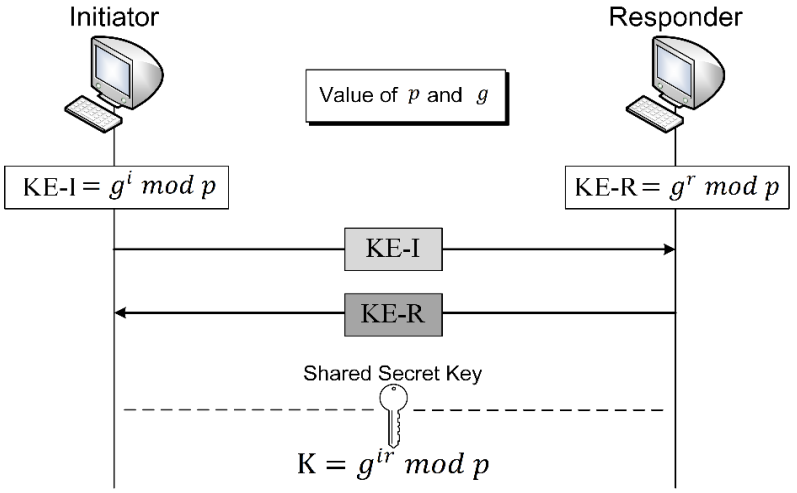
- Oakley 프로토콜
  - 개요
    - Diffie-Hellman 방식을 기반으로 한 키 교환 프로토콜
    - 교환 메시지 형식 미규정
  - Diffie-Hellman 키 교환 (1/2)
    - 개요
      - 사전 비밀 키 공유 없이 두 피어(peer)가 공통의 세션 키를 생성하는 방식
    - 동작 과정 (1/2)
      1. 공개 값 p와 g선택
         - p: 300자리의 10진수에 해당하는 큰 소수
         - g: $< Z_p^*, \times >$그룹의 생성원



<그림 23> Diffie-Hellman 키 교환

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환 (2/2)
    - 동작 과정 (2/2)
      2. Initiator가 임의의 비밀 값 i선택
         - $KE\text{-}I = g^i \bmod p$
         - $KE\text{-}I$ 를 Responder에게 전달
      3. Responder가 임의의 비밀 값 r선택
         - $KE\text{-}R = g^r \bmod p$
         - $KE\text{-}R$을 Initiator에게 전달
      4. 교환한 값을 이용해 동일한 공유 비밀 키 생성
         - $K = g^{ir} \bmod p$
         - 양쪽이 동일한 세션 키 획득



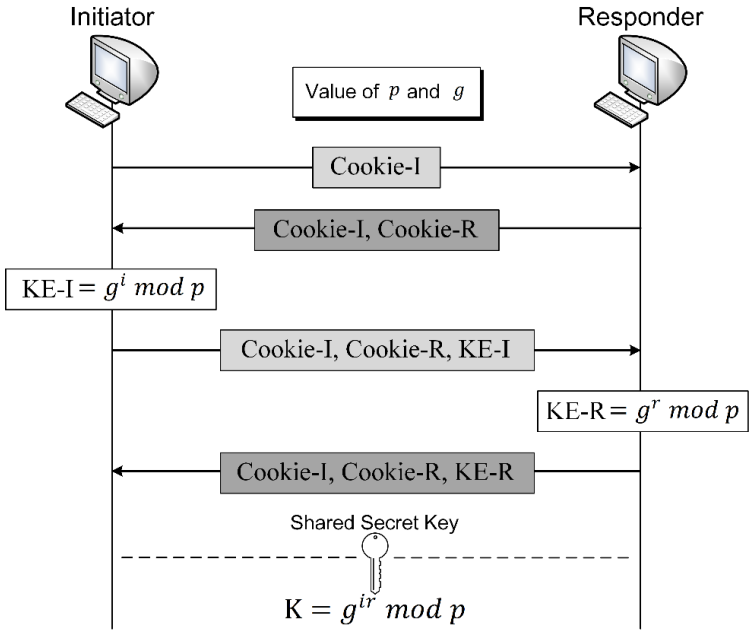
<그림 23> Diffie-Hellman 키 교환

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환의 개선 (1/5)
    - 정체 공격(Clogging Attack) (1/2)
      - 정의
        - 다수의 위조된 Diffie-Hellman 키 교환 요청을 전송하여 상대방의 계산 자원을 고갈시키는 서비스 거부(DoS, Denial of Service) 공격
      - 특징
        - 서로 다른 발신지로 위장한 다수의 절반 키(Half Key) 메시지 전송
        - 각 요청에 대한 Diffie-Hellman 연산으로 상대방의 계산 부하 증가
        - 과도한 연산으로 인한 정상 요청 처리 지연 및 서비스 거부

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환의 개선 (2/5)
    - 정체 공격(Clogging Attack) (2/2)
      - 해결 방안
        - 쿠키(Cookie) 생성
          - 식별 정보(IP 주소, 포트 번호, 프로토콜), 비밀 난수, 타임스탬프를 해싱하여 생성
        - 쿠키 교환 및 검증
          - Initiator와 Responder가 각각 자신의 쿠키를 생성하여 교환
          - 이후 메시지에 상대방의 쿠키를 포함하여 반환 여부 확인
          - 자신의 쿠키가 반환될 때까지 Diffie-Hellman 키 계산 연기
        - 정체 공격 방지
          - 쿠키를 반환하지 못하는 위조 발신자의 요청에 대한 고비용 키 계산 차단



<그림 24> 쿠키를 이용한 Diffie-Hellman 키 교환

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환의 개선 (3/5)
    - 재전송 공격(Replay Attack)
      - 정의
        - 이전 세션에서 전송된 메시지를 다시 전송하여 정상적인 키 교환으로 위장하는 공격
      - 특징
        - 과거 세션의 메시지를 공격자가 저장한 뒤 새로운 세션에서 재사용
        - 이전 메시지와 현재 메시지를 구분하기 어려운 Diffie-Hellman의 특성 악용
      - 해결 방안
        - 비표(Nonce) 사용
          - 세션마다 새로운 임의의 값 생성
          - 키 교환 메시지에 비표를 포함하여 현재 세션의 메시지임을 확인
          - 과거 메시지의 재사용 및 재전송 여부 식별

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환의 개선 (4/5)
    - 중간자 공격(Man-in-the-Middle Attack) (1/2)
      - 정의
        - 공격자가 두 통신 당사자 사이에 개입하여 각각과 별도의 키를 생성하고 통신을 중계하는 공격
      - 특징
        - 공격자가 Alice에게는 Bob으로, Bob에게는 Alice로 위장
        - Alice–공격자, 공격자–Bob 사이에 서로 다른 공유 키 생성
        - 단순한 Diffie-Hellman 키 교환만으로는 상대방의 신원 확인 불가
        - 통신 내용을 공격자가 중간에서 확인하거나 변조할 가능성

인터넷 키 교환(IKE)

- Oakley 프로토콜
  - Diffie-Hellman 키 교환의 개선 (5/5)
    - 중간자 공격(Man-in-the-Middle Attack) (2/2)
      - 해결 방안
        - 통신 당사자에 대한 개체 인증 수행
        - 메시지 무결성과 상대방 신원 확인을 통해 정상 피어 여부 검증
        - IKE에서 다음과 같은 인증 수단 사용
          - 미리 공유된 비밀 키(Pre-Shared Key)
          - 공개키 암호화용 공개키 쌍
          - 디지털 서명용 공개키 쌍

인터넷 키 교환(IKE)

- SKEME (1/2)
  - 개요
    - 인증과 안전한 키 설정을 지원하는 키 교환 메커니즘
    - 공개 키 암호화를 기반으로 개체 인증 제공
  - 기본 동작 (1/2)
    1. SHARE (Shared Key Establishment)
       - 인증용 공유키 K_0확보 단계
         - 양쪽이 상대방의 공개키로 임의 값을 암호화하여 교환
         - 양쪽의 임의 값을 결합해 공유키 K_0생성
         - 이후 Diffie-Hellman 교환을 인증하는 데 K_0사용
    2. EXCH (Exchange)
       - Diffie-Hellman 키 교환 단계
         - Initiator와 Responder가 각각 Diffie-Hellman 공개값 g^x, g^y교환
         - 이를 이용해 Diffie-Hellman 공유 비밀 생성
         - 이 값을 기반으로 실제 세션 키 생성

인터넷 키 교환(IKE)

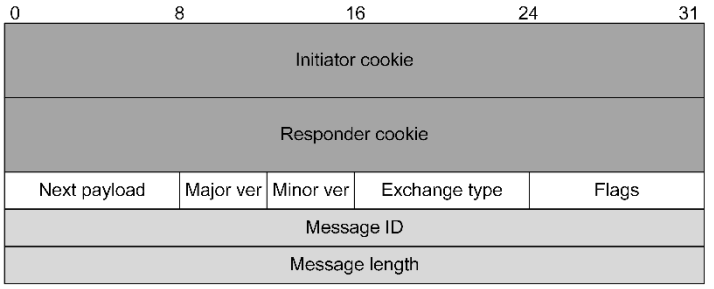
- SKEME (2/2)
  - 기본 동작 (2/2)
    3. AUTH (Authentication)
       - Diffie-Hellman 교환 인증 단계
         - SHARE에서 만든 K_0로 MAC을 생성
         - 앞에서 교환한 Diffie-Hellman 값과 양 당사자의 ID 등을 인증
         - 이를 통해 중간자 공격 방지

인터넷 키 교환(IKE)

- ISAKMP
  - 개요
    - IKE 교환을 위한 메시지를 전달하기 위해 설계된 프로토콜
    - 페이로드를 통해 SA 협상, 키 교환, 인증 등에 필요한 실제 정보 전달
  - 구조
    - 한 개의 일반 헤더와 여러 개의 페이로드(Payload)로 구성
    - 페이로드는 교환 목적에 따라 필요한 종류로 선택되어 연결

인터넷 키 교환(IKE)

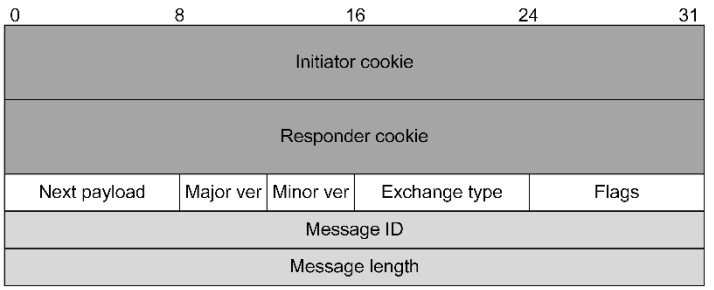
- ISAKMP
  - 일반 헤더 (1/3)
    - 개요
      - ISAKMP 메시지 전체를 관리하기 위한 기본 정보
    - 구조 (1/3)
      - 개시자 쿠키(Initiator cookie)
        - SA 설정·협상·삭제를 시작하는 개체의 쿠키를 나타내는 64비트 길이의 필드
      - 응답자 쿠키(Responder Cookie)
        - 응답 개체의 쿠키를 나타내는 64비트 길이의 필드
        - Initiator의 최초 메시지에서는 0으로 설정



<그림 25> ISAKMP 일반 헤더

인터넷 키 교환(IKE)

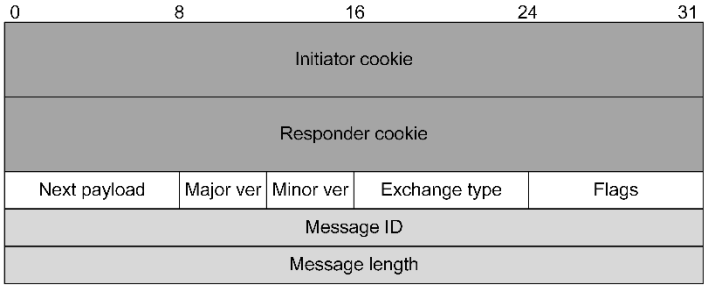
- ISAKMP
  - 일반 헤더 (2/3)
    - 구조 (2/3)
      - 다음 페이로드(Next payload)
        - 일반 헤더 뒤에 이어지는 첫 번째 페이로드의 종류를 나타내는 8비트 길이의 필드
      - 주 버전(Major version)
        - ISAKMP 프로토콜의 주 버전을 나타내는 4비트 길이의 필드
      - 부 버전(Minor version)
        - ISAKMP 프로토콜의 부 버전을 나타내는 4비트 길이의 필드
      - 교환 유형(Exchange type)
        - ISAKMP 메시지가 수행하는 교환 방식으로 IKE 모드 종류를 나타내는 8비트 길이의 필드



<그림 25> ISAKMP 일반 헤더

인터넷 키 교환(IKE)

- ISAKMP
  - 일반 헤더 (3/3)
    - 구조 (3/3)
      - 플래그(Flags)
        - 교환 과정의 옵션을 지정하는 8비트 길이의 필드
          - 암호화 비트: 이후 페이로드의 암호화 여부
          - 위임 비트(Commitment Bit): SA 설정이 완료되기 전에 암호화된 데이터를 수신하지 않도록 동기화
          - 인증 비트(Authentication Bit): 전체 메시지에 대한 인증 여부
      - 메시지 ID(Message ID)
        - 교환을 식별하는 32비트 메시지 식별자
        - 채널 설정 과정에서는 0으로 설정
        - SA 협상 과정에서는 교환마다 다른 값을 사용하여 동시에 진행되는 협상 구분
      - 메시지 길이(Message Length)
        - 일반 헤더와 모든 페이로드를 포함한 전체 ISAKMP 메시지 길이를 나타내는 32비트 길이의 필드



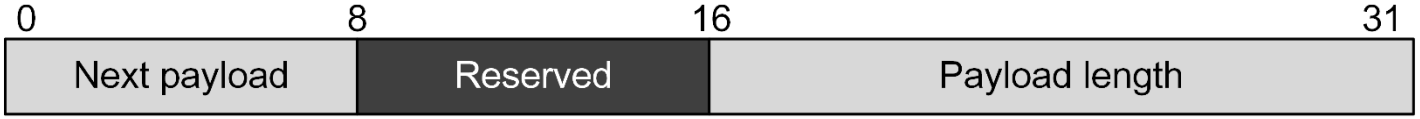
<그림 25> ISAKMP 일반 헤더

인터넷 키 교환(IKE)

- ISAKMP
  - 페이로드
    - 개요
      - ISAKMP 메시지에서 SA 협상, 키 교환, 인증 등에 필요한 실제 정보를 전달하는 영역
      - 하나의 ISAKMP 메시지에 하나 이상의 페이로드를 연속적으로 포함 가능
      - 각 페이로드는 일반 페이로드 헤더와 페이로드 종류별 고유 데이터로 구성

인터넷 키 교환(IKE)

- ISAKMP
  - 페이로드
    - 일반 페이로드 헤더
      - 다음 페이로드(Next payload)
        - 다음에 이어지는 페이로드의 종류를 식별하는 8비트 길이의 필드
        - 다음 페이로드가 없을 경우 0으로 설정
        - 현재 페이로드의 종류는 이전 페이로드 또는 일반 헤더의 Next Payload 값으로 결정
      - 페이로드 길이(Payload length)
        - 일반 페이로드 헤더를 포함한 현재 페이로드 전체 길이를 나타내는 16비트 길이의 필드
        - 길이는 바이트 단위로 표현



<그림 26> 일반 페이로드 헤더

인터넷 키 교환(IKE)

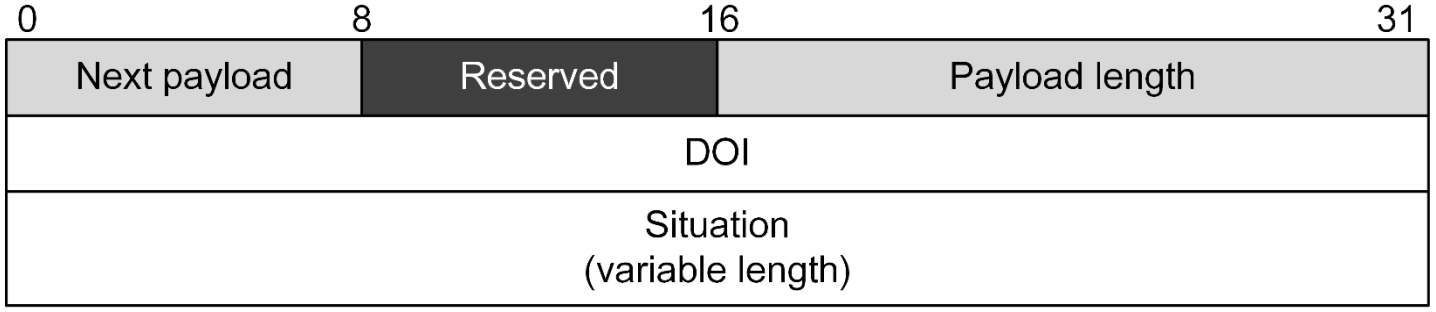
- ISAKMP
  - 페이로드
    - 종류

[표 2] ISAKMP 페이로드 종류

유형 번호	이름	설명
0	None	페이로드의 끝을 표시
1	SA	SA 협상 시작에 사용
2	Proposal	SA 협상에 사용되는 정보 포함
3	Transform	안전한 채널 생성을 위한 보안 변환 정의
4	Key Exchange	키 생성에 사용되는 데이터 전달
5	Identification	통신 상대의 식별 정보 전달
6	Certification	공개키 인증서 전달
7	Certification Request	상대에게 인증서 요청
8	Hash	해시 함수로 생성된 데이터 전달
9	Signature	서명 함수로 생성된 데이터 전달
10	Nonce	비표로 사용되는 임의 생성 데이터 전달
11	Notification	SA와 관련된 오류 메시지 또는 상태 정보 전달
12	Delete	송신자가 삭제한 하나 이상의 SA 정보 전달
13	Vendor	벤더별 확장 기능 정의

인터넷 키 교환(IKE)

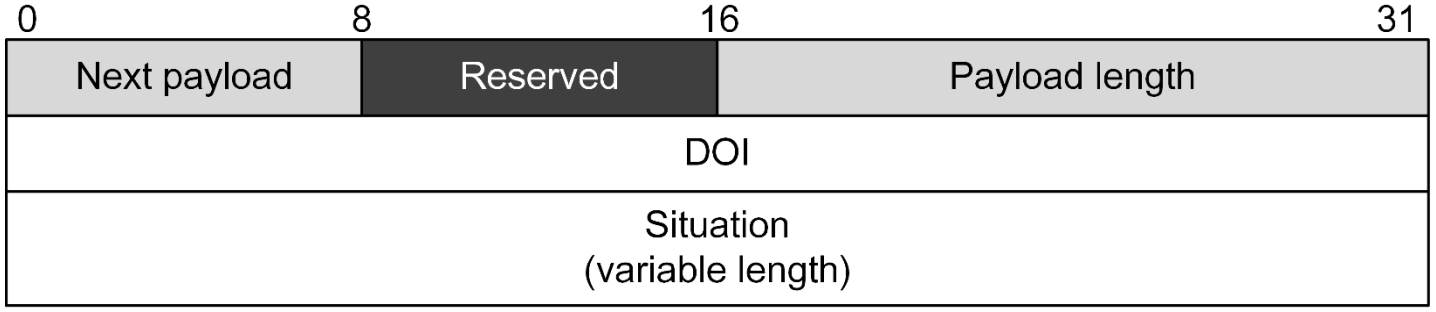
- ISAKMP
  - SA 페이로드(SA Payload) (1/2)
    - 개요
      - SA 협상을 시작하기 위한 페이로드
      - 실제 보안 매개변수는 SA 페이로드에 직접 포함하지 않고 Proposal Payload와 Transform Payload를 통해 정의
    - 특징
      - 하나의 SA Payload 뒤에 하나 이상의 Proposal Payload 연결 가능



<그림 27> SA 페이로드

인터넷 키 교환(IKE)

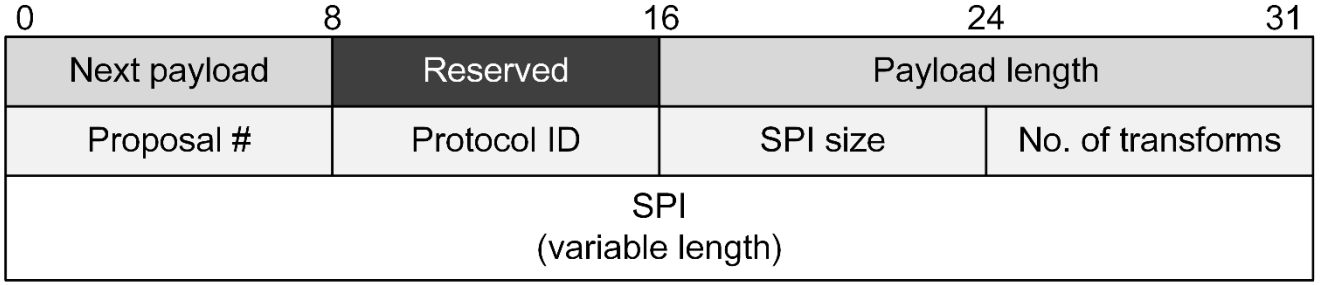
- ISAKMP
  - SA 페이로드(SA Payload) (2/2)
    - 구성
      - 해석 도메인(DOI, Domain of Interpretation)
        - SA가 적용되는 보안 환경의 종류를 정의하는 32비트 길이의 필드
        - 값에 따라 일반 SA(0) 또는 IPSec SA(1) 등을 구분
      - 상황(Situation)
        - SA가 적용되어야 하는 보안 요구사항이나 상황을 정의하는 가변 길이 필드



<그림 27> SA 페이로드

인터넷 키 교환(IKE)

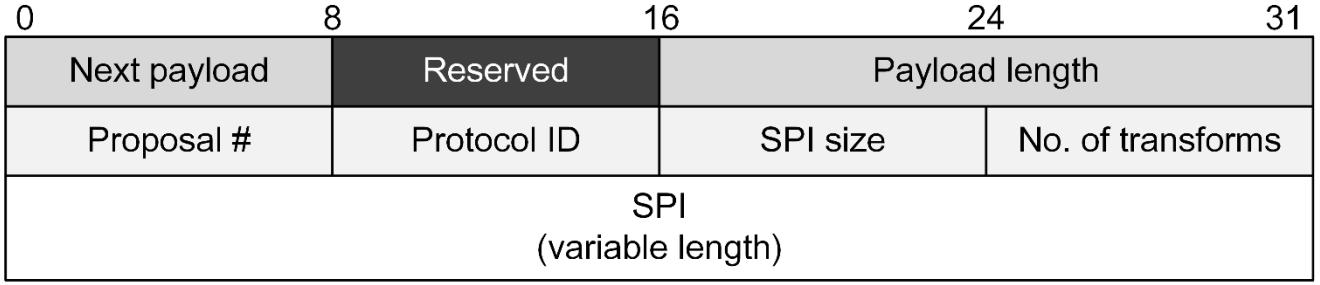
- ISAKMP
  - 제안 페이로드(Proposal Payload) (1/3)
    - 개요
      - SA 협상에서 사용할 프로토콜과 SPI를 제안하기 위한 페이로드
      - 구체적인 보안 매개변수는 뒤따르는 Transform Payload를 통해 제안
    - 특징
      - 각 Proposal Payload 뒤에 하나 이상의 Transform Payload 연결
      - 여러 Transform Payload를 통해 선택 가능한 보안 매개변수 집합 제시



<그림 28> 제안 페이로드

인터넷 키 교환(IKE)

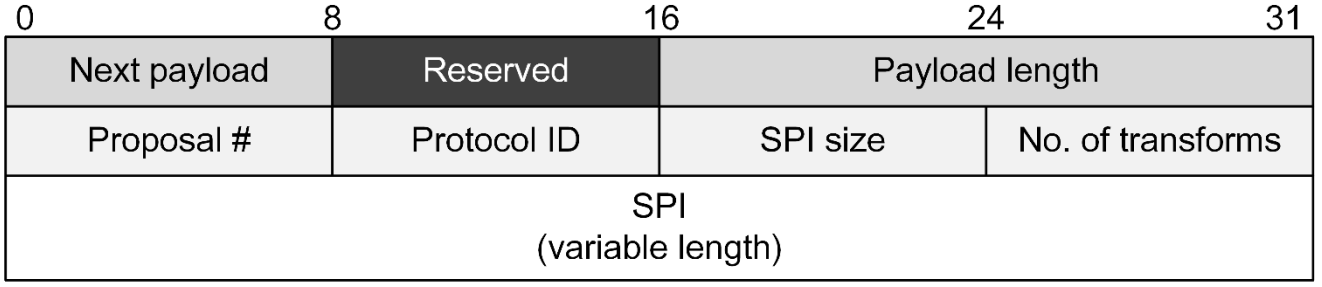
- ISAKMP
  - 제안 페이로드(Proposal Payload) (2/3)
    - 구성 (1/2)
      - 제안 번호(Proposal #)
        - Responder가 제안을 참조할 수 있도록 식별하는 8비트 길이의 필드
      - 프로토콜 ID(Protocol ID)
        - SA 협상 대상 프로토콜을 식별하는 8비트 길이의 필드
        - ISAKMP(1), ESP(2), AH(3)
      - SPI 크기(SPI size)
        - SPI의 길이를 바이트 단위로 나타내는 8비트 길이의 필드



<그림 28> 제안 페이로드

인터넷 키 교환(IKE)

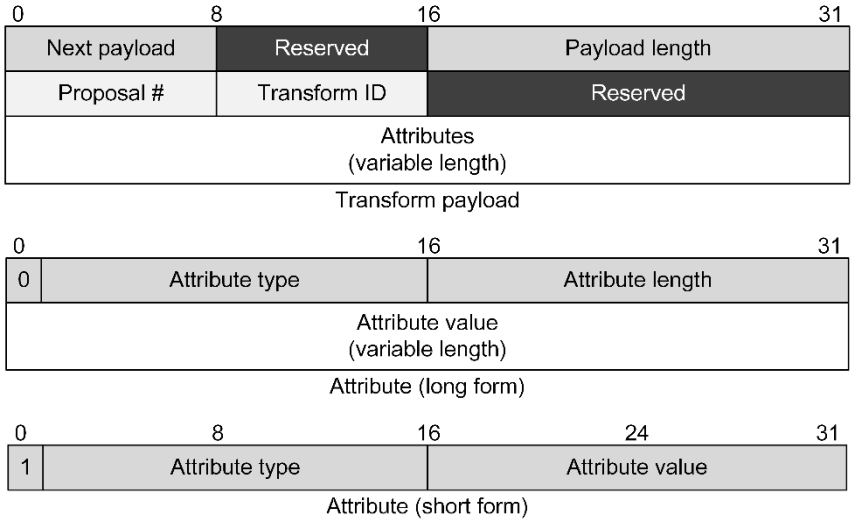
- ISAKMP
  - 제안 페이로드(Proposal Payload) (3/3)
    - 구성 (2/2)
      - 변환 개수(Number of transforms)
        - 해당 Proposal Payload 뒤에 이어지는 Transform Payload의 개수를 나타내는 8비트 길이의 필드
      - SPI
        - 제안에 사용되는 실제 SPI 값을 저장하는 가변 길이 필드
        - 32비트 경계에 맞지 않아도 Padding 미사용



<그림 28> 제안 페이로드

인터넷 키 교환(IKE)

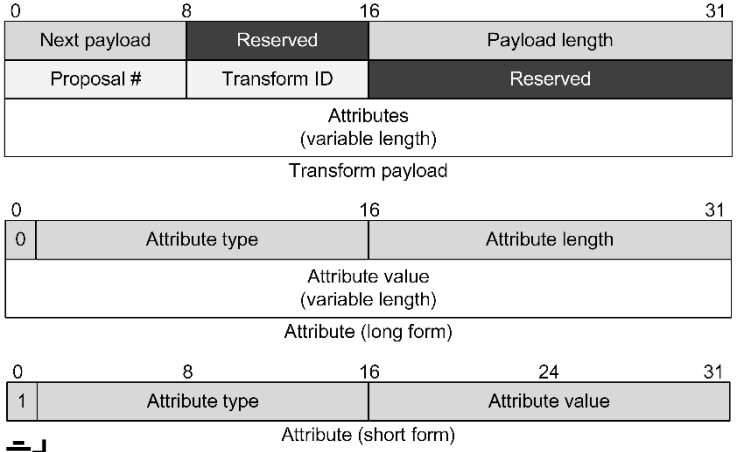
- ISAKMP
  - 변환 페이로드(Transform Payload) (1/2)
    - 개요
      - SA 협상에 사용할 실제 보안 속성을 전달하는 페이로드
      - Proposal Payload에서 제안한 프로토콜에 적용할 구체적인 보안 매개변수 정의
    - 특징
      - 하나의 Proposal Payload 뒤에 하나 이상의 Transform Payload 연결 가능
      - 각 Transform Payload는 서로 다른 변환 번호 부여
      - 각 Transform Payload에 0개 이상의 속성(Attribute) 포함 가능



<그림 29> 변환 페이로드

인터넷 키 교환(IKE)

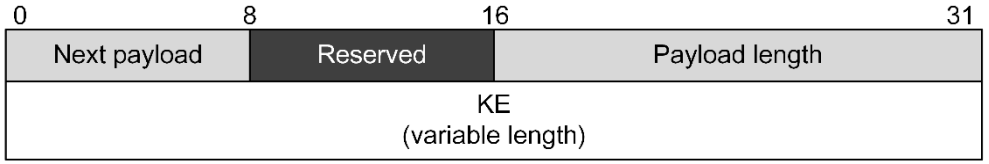
- ISAKMP
  - 변환 페이로드(Transform Payload) (2/2)
    - 구성
      - 변환 번호(Transform #)
        - 각 변환을 식별하는 8비트 길이의 필드
      - 변환 ID(Transform ID)
        - 변환의 종류를 식별하는 8비트 길이의 필드
      - 속성(Attributes)
        - 해당 변환에 사용할 구체적인 보안 속성을 정의하는 가변 길이 필드
          - 속성 유형(Attribute Type)
            - DOI에 따라 속성의 종류 정의
          - 속성 길이(Attribute Length)
            - 속성 값의 길이 정의
          - 속성 값(Attribute Value)
            - 실제 속성 값 저장
            - 속성 값은 2바이트의 짧은 형식 또는 가변 길이의 긴 형식으로 표현



<그림 29> 변환 페이로드

인터넷 키 교환(IKE)

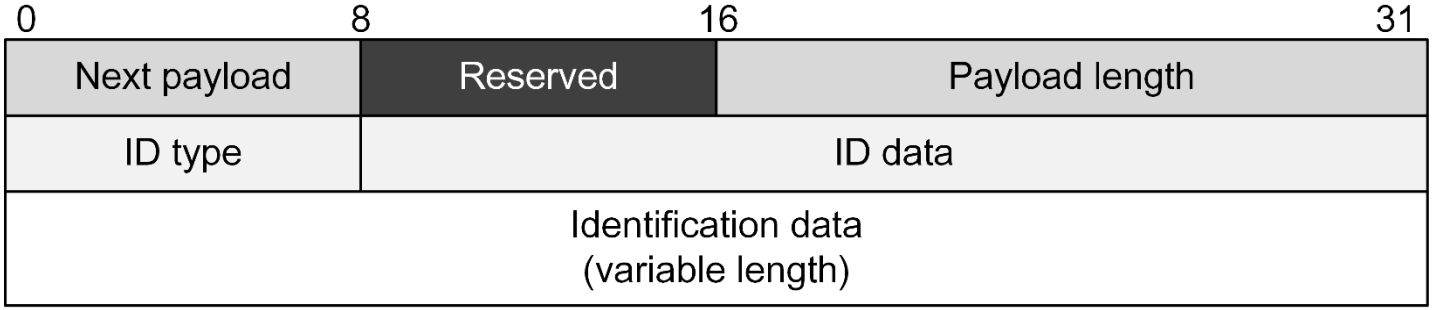
- ISAKMP
  - 키 교환 페이로드(Key-Exchange Payload)
    - 개요
      - 세션 키 생성에 필요한 키 교환 데이터를 전달하는 페이로드
      - Diffie-Hellman 방식의 절반 키와 같은 키 생성용 데이터 전달에 사용
    - 특징
      - 키 교환 방식에 따라 필요한 예비 키 데이터 전달
      - 전달된 데이터를 기반으로 양 당사자가 공통 세션 키 생성
    - 구성
      - KE(Key Exchange Data)
        - 세션 키 생성에 필요한 데이터를 저장하는 가변 길이 필드
        - Diffie-Hellman 사용 시 절반 키 값 전달



<그림 30> 키 교환 페이로드

인터넷 키 교환(IKE)

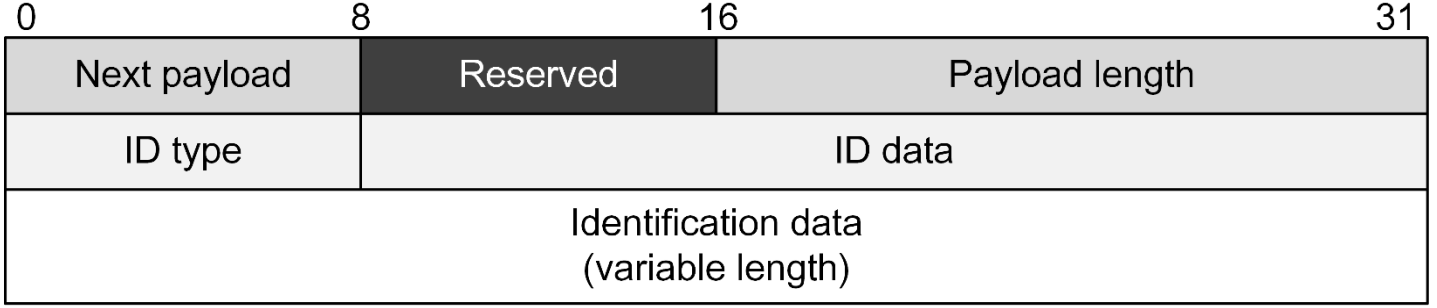
- ISAKMP
  - 식별 페이로드(Identification Payload) (1/2)
    - 개요
      - 통신 당사자의 식별 정보를 전달하기 위한 페이로드
      - 상대방이 통신 개체의 신원을 확인할 수 있도록 사용
    - 특징
      - DOI에 따라 사용할 식별자 유형을 지정
      - 실제 식별 정보는 가변 길이 데이터로 전달



<그림 31> 식별 페이로드

인터넷 키 교환(IKE)

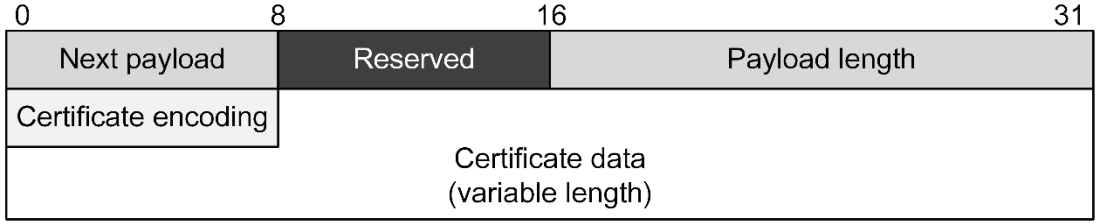
- ISAKMP
  - 식별 페이로드(Identification Payload) (2/2)
    - 구성
      - ID 유형(ID type)
        - 사용할 식별 정보의 종류를 나타내는 8비트 길이의 필드
        - DOI에 따라 정의된 ID 유형 사용
      - ID 데이터(ID data)
        - DOI에 따른 추가 식별 정보(프로토콜, 포트)를 담는 24비트 길이의 필드
        - 사용하지 않을 경우 0으로 처리
      - 식별 데이터(Identification data)
        - 실제 통신 개체의 식별 정보를 저장하는 가변 길이 필드



<그림 31> 식별 페이로드

인터넷 키 교환(IKE)

- ISAKMP
  - 인증 페이로드(Certificate Payload) (1/2)
    - 개요
      - 통신 당사자의 공개키·서명 검증 등에 필요한 인증 정보를 전달하는 페이로드
      - 교환 과정에서 선택적으로 사용
    - 특징
      - 인증서의 유형을 지정하여 다양한 인증 형식 지원
      - 실제 인증 정보는 가변 길이 데이터로 전달



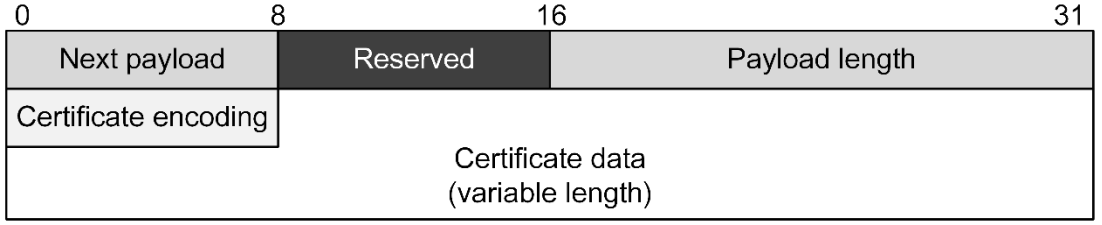
<그림 32> 인증 페이로드

[표 3] 인증 유형

유형 번호	이름
0	None
1	Wrapped X.509 Certificate
2	PGP Certificate
3	DNS Signed Key
4	X.509 Certificate – Signature
5	X.509 Certificate – Key Exchange
6	Kerberos Tokens
7	Certificate Revocation List
8	Authority Revocation List
9	SPKI Certificate
10	X.509 Certificate – Attribute

인터넷 키 교환(IKE)

- ISAKMP
  - 인증 페이로드(Certificate Payload) (2/2)
    - 구성
      - 인증서 부호화(Certificate encoding)
        - 전달되는 인증서의 유형을 나타내는 8비트 길이의 필드
        - 유형 번호를 값으로 사용
      - 인증서 데이터(Certificate data)
        - 실제 인증 정보를 저장하는 가변 길이 필드
        - 필드 길이는 선택된 인증서 유형에 따라 결정



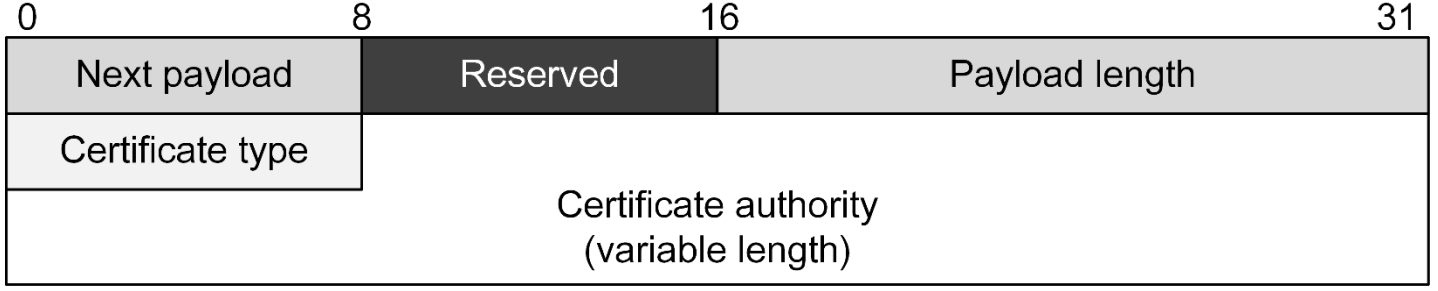
<그림 32> 인증 페이로드

[표 3] 인증 유형

유형 번호	이름
0	None
1	Wrapped X.509 Certificate
2	PGP Certificate
3	DNS Signed Key
4	X.509 Certificate – Signature
5	X.509 Certificate – Key Exchange
6	Kerberos Tokens
7	Certificate Revocation List
8	Authority Revocation List
9	SPKI Certificate
10	X.509 Certificate – Attribute

인터넷 키 교환(IKE)

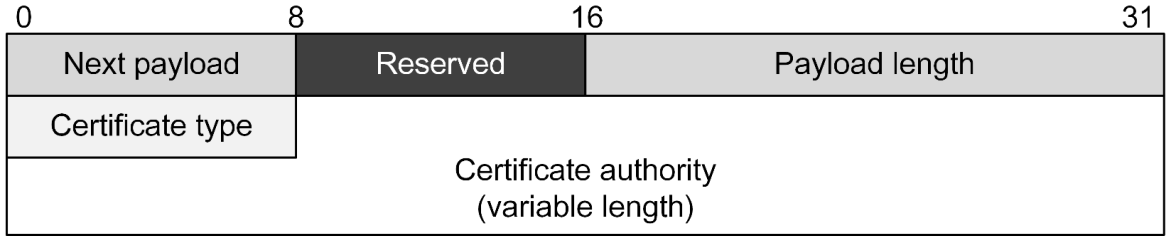
- ISAKMP
  - 인증서 요청 페이로드(Certificate Request Payload) (1/2)
    - 개요
      - 상대방에게 특정 유형의 인증서를 요청하기 위한 페이로드
      - 필요한 인증서 형식과 인증기관 정보를 함께 전달
    - 특징
      - 요청할 인증서 유형을 명시
      - 특정 인증기관이 발급한 인증서를 요청 가능



<그림 33> 인증서 요청 페이로드

인터넷 키 교환(IKE)

- ISAKMP
  - 인증서 요청 페이로드(Certificate Request Payload) (2/2)
    - 구성
      - 인증서 유형(Certificate Type)
        - 요청할 인증서의 유형을 나타내는 8비트 길이의 필드
        - 인증 페이로드에서 정의된 인증서 유형 사용
      - 인증기관(Certificate Authority)
        - 요청하는 인증서를 발급한 인증기관을 나타내는 가변 길이 필드



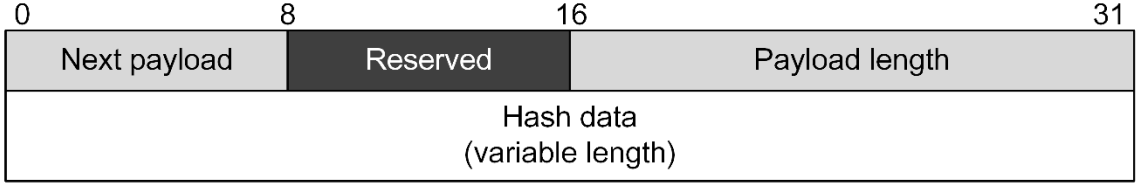
<그림 33> 인증서 요청 페이로드

[표 3] 인증 유형

유형 번호	이름
0	None
1	Wrapped X.509 Certificate
2	PGP Certificate
3	DNS Signed Key
4	X.509 Certificate – Signature
5	X.509 Certificate – Key Exchange
6	Kerberos Tokens
7	Certificate Revocation List
8	Authority Revocation List
9	SPKI Certificate
10	X.509 Certificate – Attribute

인터넷 키 교환(IKE)

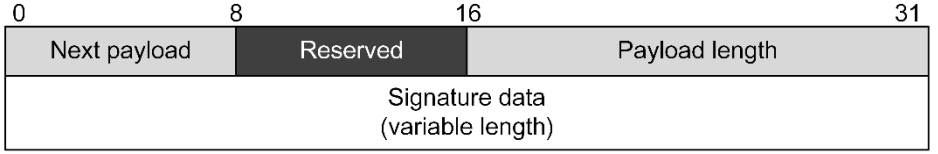
- ISAKMP
  - 해시 페이로드(Hash Payload)
    - 개요
      - IKE 교환 과정에서 생성된 해시값을 전달하기 위한 페이로드
      - ISAKMP 상태 메시지 또는 메시지 일부의 무결성 보호에 사용
    - 특징
      - 해시 함수를 통해 생성된 검증용 데이터 전달
      - 메시지 전체 또는 일부에 적용된 해시값으로 변조 여부 확인
    - 구성
      - 해시 데이터(Hash Data)
        - 해시 함수 적용 결과를 저장하는 가변 길이 필드
        - 적용 범위는 ISAKMP 상태 메시지 전체 또는 일부



<그림 34> 해시 페이로드

인터넷 키 교환(IKE)

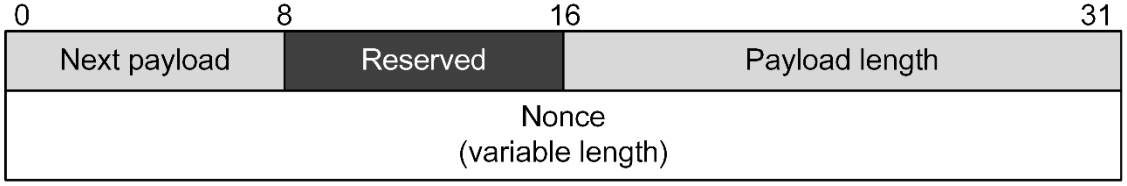
- ISAKMP
  - 서명 페이로드(Signature Payload)
    - 개요
      - 디지털 서명값을 전달하기 위한 페이로드
      - 메시지 또는 ISAKMP 상태의 일부에 대해 생성된 서명을 이용한 송신자 인증 및 무결성 확인
    - 특징
      - 디지털 서명 절차를 통해 생성된 검증용 서명 데이터 전달
      - 수신자는 상대방의 공개키를 이용하여 서명 검증
    - 구성
      - 서명 데이터(Signature Data)
        - 메시지 또는 ISAKMP 상태의 일부에 디지털 서명 절차를 적용하여 생성한 값을 저장하는 가변 길이 필드



<그림 35> 서명 페이로드

인터넷 키 교환(IKE)

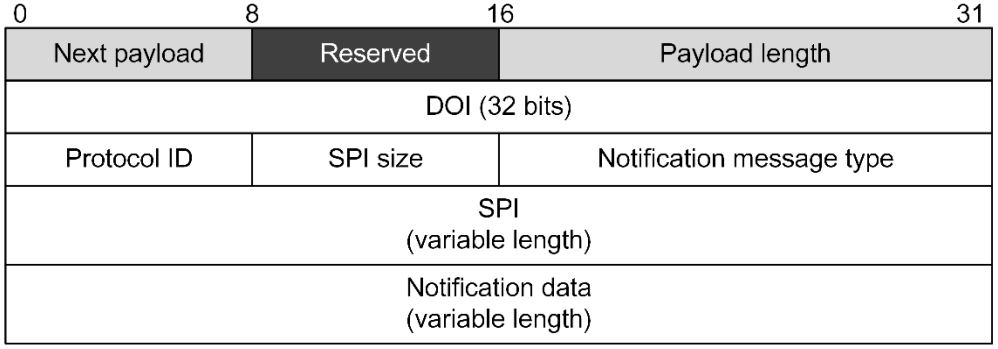
- ISAKMP
  - 비표 페이로드(Nonce Payload)
    - 개요
      - 메시지의 신선성(Freshness)을 보장하고 재전송 공격을 방지하기 위한 임의값을 전달하는 페이로드
      - IKE 교환 과정에서 생성한 Nonce 값 전달에 사용
    - 특징
      - 세션마다 새로운 임의값 사용
      - 이전 세션의 메시지와 현재 메시지를 구분하여 재전송 여부 확인
    - 구성
      - 비표(Nonce)
        - Nonce 값을 저장하는 가변 길이 필드



<그림 36> 비표 페이로드

인터넷 키 교환(IKE)

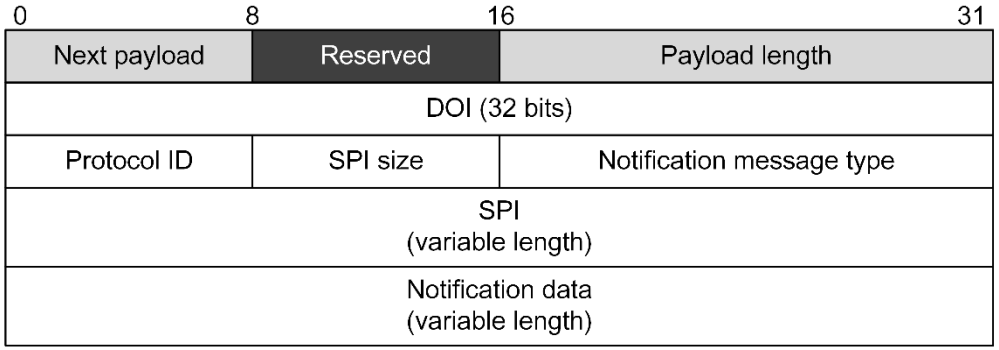
- ISAKMP
  - 통지 페이로드(Notification Payload) (1/4)
    - 개요
      - SA 협상 과정에서 발생한 상태 또는 오류 정보를 상대방에게 전달하기 위한 페이로드
      - 오류 보고뿐 아니라 SA와 관련된 상태 정보 전달에도 사용
    - 특징
      - 통지 대상이 되는 프로토콜과 SA를 함께 식별
      - 통지 메시지 유형에 따라 오류 또는 상태 정보 구분
      - 필요한 경우 추가적인 통지 데이터를 함께 전달



<그림 37> 통지 페이로드

인터넷 키 교환(IKE)

- ISAKMP
  - 통지 페이로드(Notification Payload) (2/4)
    - 구성 (1/3)
      - 해석 도메인(DOI)
        - 통지 정보가 적용되는 보안 환경을 나타내는 32비트 길이의 필드
        - SA Payload의 DOI와 동일한 의미
      - 프로토콜 ID(Protocol ID)
        - 통지 대상 프로토콜을 식별하는 8비트 길이의 필드
      - SPI 크기(SPI Size)
        - 뒤에 포함되는 SPI의 길이를 나타내는 8비트 길이의 필드



<그림 37> 통지 페이로드

인터넷 키 교환(IKE)

- ISAKMP
  - 통지 페이로드(Notification Payload) (3/4)
    - 구성 (2/3)
      - 통지 메시지 유형(Notification message type)
        - 전달할 오류 또는 상태 정보의 종류를 나타내는 16비트 길이의 필드
          - 오류 통지 값: 오류의 종류를 나타내는 값
          - 상태 통지 값: 정상적인 상태나 SA 관련 상태를 나타내는 값

[표 4] 상태 통지 값

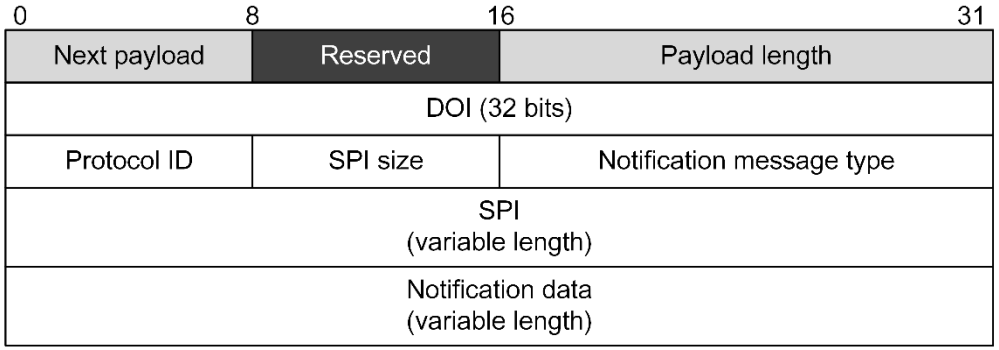
값	설명
16384	CONNECTED
24576-32767	DOI-specific codes

[표 5] 오류 통지 값

번호	이름	번호	이름	번호	이름	번호	이름
1	INVALID-PAYLOAD-TYPE	8	INVALID-FLAGS	15	BAD-PROPOSAL-SYNTAX	23	INVALID-HASH-INFORMATION
2	DOI-NOT-SUPPORTED	9	INVALID-MESSAGE-ID	16	PAYLOAD-MALFORMED	24	AUTHENTICATION-FAILED
3	SITUATION-NOT-SUPPORTED	10	INVALID-PROTOCOL-ID	17	INVALID-KEY-INFORMATION	25	INVALID-SIGNATURE
4	INVALID-COOKIE	11	INVALID-SPI	18	INVALID-ID-INFORMATION	26	ADDRESS-NOTIFICATION
5	INVALID-MAJOR-VERSION	12	INVALID-TRANSFORM-ID	19	INVALID-CERT-ENCODING	27	NOTIFY-SA-LIFETIME
6	INVALID-MINOR-VERSION	13	ATTRIBUTE-NOT-SUPPORTED	20	INVALID-CERTIFICATE	28	CERTIFICATE-UNAVAILABLE
7	INVALID-EXCHANGE-TYPE	14	NO-PROPOSAL-CHOSEN	21	CERT-TYPE-UNSUPPORTED	29	UNSUPPORTED-EXCHANGE-TYPE
				22	INVALID-CERT-AUTHORITY	30	UNEQUAL-PAYLOAD-LENGHTS

인터넷 키 교환(IKE)

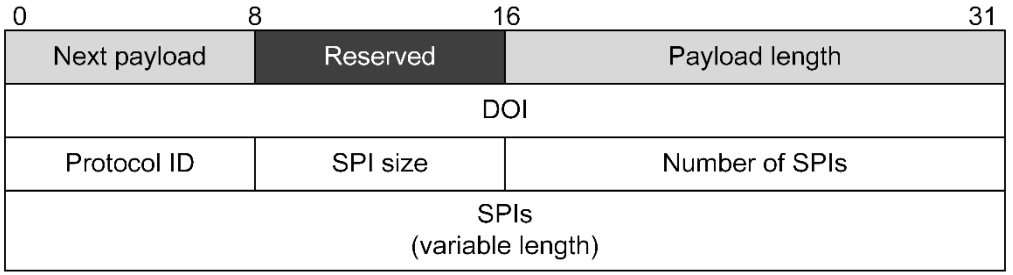
- ISAKMP
  - 통지 페이로드(Notification Payload) (4/4)
    - 구성 (3/3)
      - SPI
        - 통지 대상이 되는 SA를 식별하는 가변 길이 필드
      - 통지 데이터(Notification data)
        - 오류 또는 상태에 대한 추가 정보를 전달하는 가변 길이 필드



<그림 38> 통지 페이로드

인터넷 키 교환(IKE)

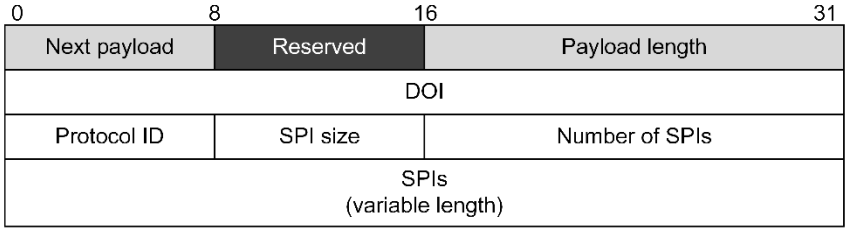
- ISAKMP
  - 삭제 페이로드(Delete Payload) (1/2)
    - 개요
      - 하나 이상의 SA 삭제 사실을 상대방에게 알리기 위한 페이로드
      - 더 이상 사용하지 않는 SA의 지원 종료를 통지하는 데 사용
    - 특징
      - 하나의 Delete Payload에서 여러 SA의 삭제를 동시에 알림 가능
      - 삭제 대상 SA를 SPI를 통해 식별



<그림 39> 삭제 페이로드

인터넷 키 교환(IKE)

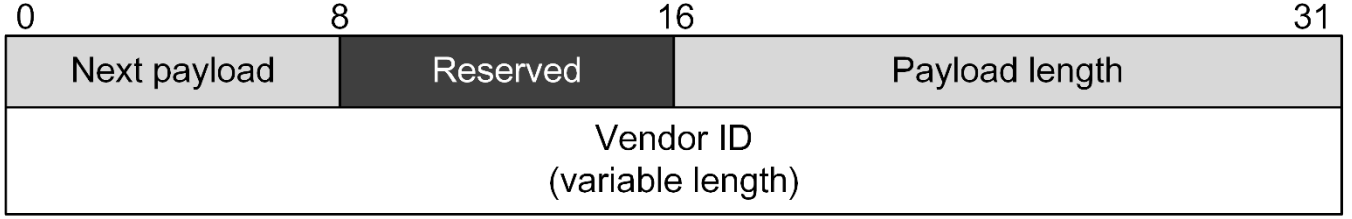
- ISAKMP
  - 삭제 페이로드(Delete Payload) (2/2)
    - 구성
      - DOI(Domain Of Interpretation)
        - 삭제 정보가 적용되는 보안 환경을 나타내는 32비트 길이의 필드
      - 프로토콜 ID(Protocol ID)
        - 삭제 대상 SA가 사용하는 프로토콜을 식별하는 8비트 길이의 필드
      - SPI 크기(SPI size)
        - 각 SPI의 길이를 바이트 단위로 나타내는 8비트 길이의 필드
      - SPI 번호(Number of SPIs)
        - 삭제 대상으로 포함된 SPI의 개수를 나타내는 16비트 길이의 필드
      - SPIs
        - 삭제할 SA들을 식별하는 SPI 값을 저장하는 가변 길이 필드



<그림 39> 삭제 페이로드

인터넷 키 교환(IKE)

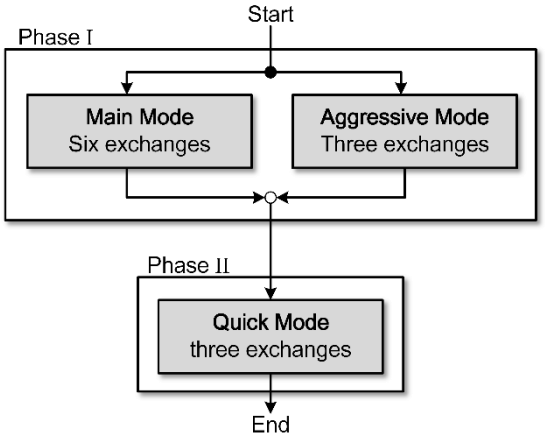
- ISAKMP
  - 벤더 페이로드(Vendor Payload)
    - 개요
      - 특정 벤더와 관련된 정보를 교환하기 위한 페이로드
      - ISAKMP에서 벤더 별 확장 기능이나 고유 정보 식별에 사용
    - 특징
      - 벤더마다 서로 다른 고유 식별 값 사용
      - 표준 ISAKMP 기능 외에 벤더 별 기능을 구분하는 데 활용
    - 구성
      - 벤더 ID(Vendor ID)
        - 해당 벤더를 식별하기 위한 고유 값을 저장하는 가변 길이 필드



<그림 40> 벤더 페이로드

인터넷 키 교환(IKE)

- 동작 구조
  - Phase I
    - 개요
      - IKE 당사자 간 인증 및 안전한 통신 채널 설정
      - Phase II에서 사용할 IKE SA 생성
    - 모드
      - 주 모드(Main Mode)
      - 공격적 모드(Aggressive Mode)
  - Phase II
    - 개요
      - AH 또는 ESP에서 사용할 키와 SA 협상
      - Phase I에서 생성한 IKE SA를 이용하여 실제 IPSec SA 생성
    - 모드
      - 신속 모드(Quick Mode)



<그림 41> IKE Phase와 모드

인터넷 키 교환(IKE)

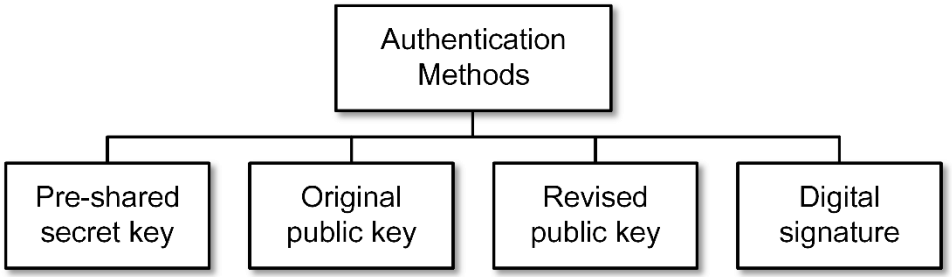
- Phase Ⅰ
  - 주 모드(Main Mode)
    - 개요
      - 6개의 메시지 교환을 통해 단계 II에서 사용할 SA를 생성하는 모드
    - 메시지 교환 구조
      - 1·2번째 메시지
        - 쿠키 교환
        - SA 매개변수 제안 및 선택
        - 정체 공격 방지를 위한 상대방 존재 확인
      - 3·4번째 메시지
        - Diffie-Hellman 절반 키(KE_I, KE_R), Nonce(N_I, N_R) 교환
        - 이후 공통 비밀 및 인증·암호화에 사용할 키(SKEYID) 계산
      - 5·6번째 메시지
        - 각 당사자의 신원 확인을 위한 인증 정보 교환
        - 인증 방법에 따라 ID, 해시, 서명 등의 구성에 차이
        - 인증 정보의 보호를 통해 상호 인증 수행

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 키 생성
      - SKEYID
        - 3·4번째 메시지 교환 후 두 피어가 계산하는 공통 비밀
        - 이후 사용할 여러 키와 인증 값 생성의 기준 값
        - 인증 방법에 따라 서로 다른 계산 방식 사용
        - PRF(Pseudorandom Function)를 이용하여 계산
          - 협상 단계에서 정의되는 키가 있는 해시 함수
      - 목적별 키 유도
        - 다른 키 생성을 위한 유도 키
          - $\mathrm{SKEYID_d} = prf(\mathrm{SKEYID}, \mathrm{g^{ir}}|\mathrm{Cookie-I}|\mathrm{Cookie-R}|0)$
        - 인증을 위한 키
          - $\mathrm{SKEYID_a} = prf(\mathrm{SKEYID}, \mathrm{SKEYID_d}|\mathrm{g^{ir}}|\mathrm{Cookie-I}|\mathrm{Cookie-R}|1)$
        - 암호화를 위한 키
          - $\mathrm{SKEYID_e} = prf(\mathrm{SKEYID},\ \mathrm{SKEYID_a}|\mathrm{g^{ir}}|\mathrm{Cookie-I}|\mathrm{Cookie-R}|2)$

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 인증 값 생성
      - $\mathrm{HASH\text{-}I}$ / $\mathrm{HASH\text{-}R}$
        - 당사자 인증을 위해 생성하는 해시 다이제스트
          - Initiator의 인증 값
            - $\mathrm{HASH\text{-}I} = prf(\mathrm{SKEYID}, \mathrm{KE\text{-}I}|\mathrm{KE\text{-}R}|\mathrm{Cookie\text{-}I}|\mathrm{Cookie\text{-}R}|\mathrm{SA\text{-}I}|\mathrm{ID\text{-}I})$
          - Responder의 인증 값
            - $\mathrm{HASH\text{-}R} = prf(\mathrm{SKEYID}, \mathrm{KE\text{-}I}|\mathrm{KE\text{-}R}|\mathrm{Cookie\text{-}I}|\mathrm{Cookie\text{-}R}|\mathrm{SA\text{-}I}|\mathrm{ID\text{-}R})$
        - Diffie-Hellman 절반 키, 쿠키, SA 정보, ID 등을 기반으로 계산
        - 교환 정보의 변조 방지 및 상대방 인증에 활용
    - 인증 방법
      - 사전공유 비밀 키(Pre-shared Secret Key)
      - 본래의 공개 키(Original Public Key)
      - 개정된 공개 키(Revised Public Key)
      - 디지털 서명(Digital Signature)

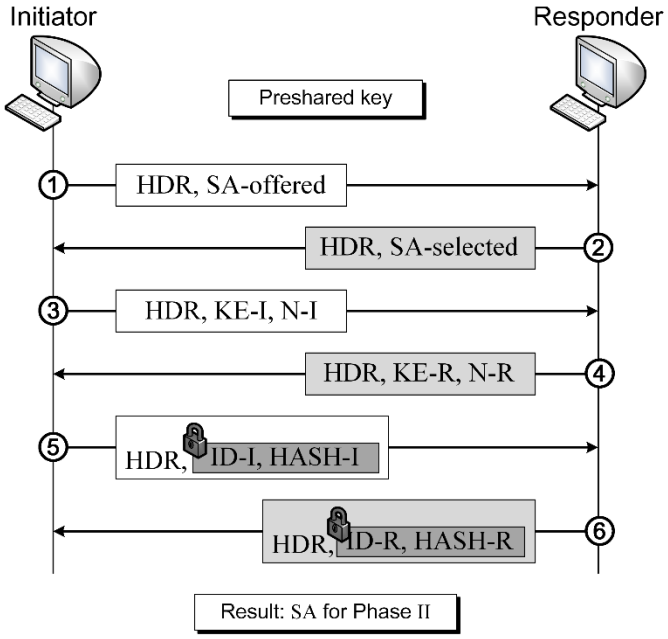


<그림 42> 주 모드와 공격 모드의 인증 방법

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 사전공유 비밀 키 방법 (1/4)
      - 개요
        - 통신 당사자가 미리 공유한 비밀 키를 이용하여 상호 인증하는 방법
        - 사전 공유 비밀 키는 두 당사자만 알고 있으며, 공격자는 해당 키에 접근 불가
      - 특징
        - 사전 공유 비밀 키를 아는 당사자만 올바른 SKEYID와 HASH 생성 가능
          $\text{SKEYID} = prf(\text{preshared key}, \text{N_I} \mid \text{N_R})$
        - 5·6번째 메시지의 ID와 HASH에 대한 암호화 보호

KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's hash
HDR: General header including cookies
Encrypted with SKEYID_e

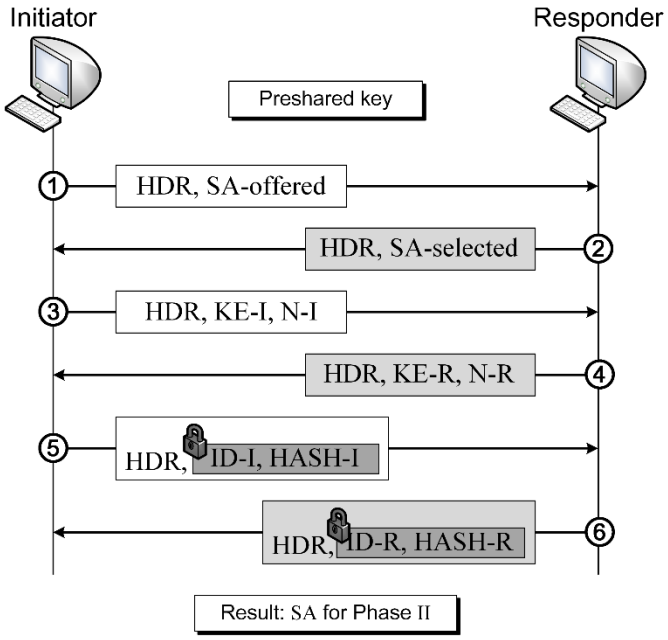


<그림 43> 주 모드에서의 사전공유 비밀 키 방법

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 사전공유 비밀 키 방법 (2/4)
      - 메시지 교환 (1/2)
        - 1·2번째 메시지
          - 쿠키와 SA 매개변수 교환
          - Initiator가 SA 제안
            → Responder가 하나를 선택
        - 3·4번째 메시지
          - Diffie-Hellman 절반 키(KE_I, KE_R) 교환
          - Nonce(N_I, N_R) 교환 사전 공유
          - 비밀 키와 Nonce를 이용하여 SKEYID 생성
          - SKEYID를 기반으로 HASH_I, HASH_R 및 관련 키 생성

KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's hash
HDR: General header including cookies
Encrypted with SKEYID_e

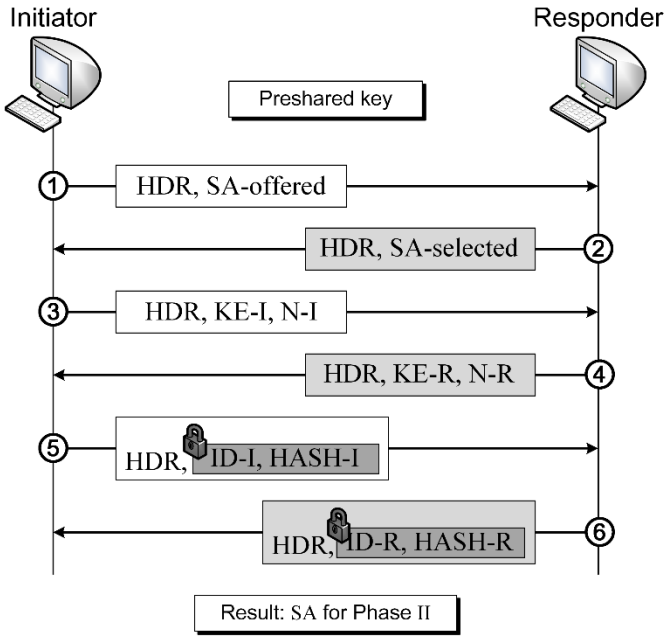


<그림 43> 주 모드에서의 사전공유 비밀 키 방법

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 사전공유 비밀 키 방법 (3/4)
      - 메시지 교환 (2/2)
        - 5·6번째 메시지
          - ID와 HASH 값 교환
            - 5번째: Initiator → ID–I, HASH–I
            - 6번째: Responder → ID–R, HASH–R
          - ID와 HASH의 보호를 위해 SKEYID_e로 암호화
          - 수신자가 동일한 HASH를 계산하여 비교함으로써 상대방 인증

KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's hash
HDR: General header including cookies
Encrypted with SKEYID_e



<그림 43> 주 모드에서의 사전공유 비밀 키 방법

인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 사전공유 비밀 키 방법 (4/4)
      - 한계
        - 사전 공유 비밀 키 선택을 위한 상대방 식별 한계
          - 수신자는 복호화 전 사용할 사전 공유 비밀 키를 선택해야 하나, 이를 위한 송신자 ID가 암호화되어 있어 확인 불가
        - IP 주소 기반 키 선택의 한계
          - 대안으로 암호화되지 않은 송신자 IP 주소를 ID 대신 활용하여 키 선택
          - IP가 고정된 환경에서만 사용 가능하며, IP가 바뀌는 원격 사용자에게는 적용 불가

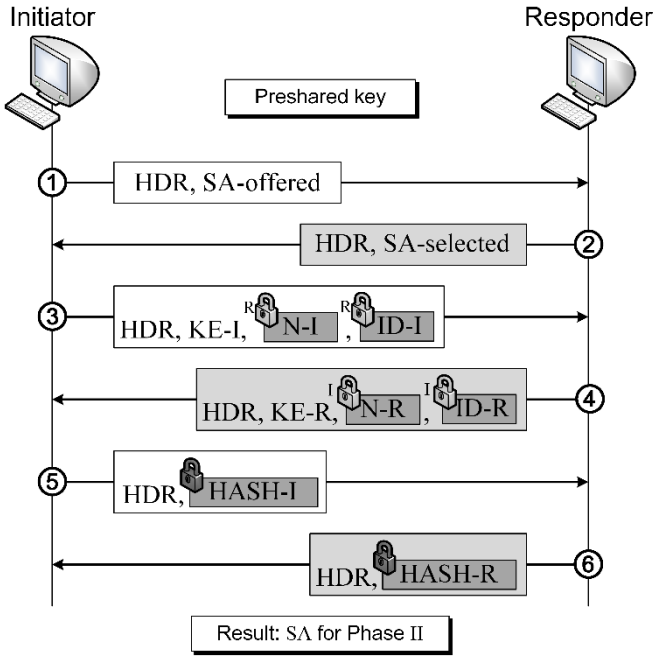
인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 본래의 공개 키 방법 (1/3)
      - 개요
        - Initiator와 Responder가 서로 알려진 공개키와 이에 대응하는 개인키를 이용하여 자신의 신원을 인증하는 방법
        - 각 당사자는 자신의 개인키를 보유하고, 상대방의 공개키를 알고 있는 상태
        - 사전공유 비밀 키 방법의 키 공유 문제를 공개 키로 개선
      - 특징
        - Nonce와 Diffie-Hellman 공유 비밀을 이용하여 SKEYID 계산
          $\text{SKEYID} = prf(\text{hash}(\text{N_I} \mid \text{N_R}), \text{Cookie_I} \mid \text{Cookie_R})$
        - 상대방의 공개 키로 암호화된 Nonce는 해당 개인 키를 가진 당사자만 복호화 가능
        - 수신한 HASH와 직접 계산한 HASH의 일치를 통해 상대방 신원 확인

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 본래의 공개 키 방법 (2/3)
      - 메시지 교환 구조
        - 1·2번째 메시지
          - 사전 공유 비밀 키 방법과 동일
          - 쿠키 교환
          - SA 매개변수 제안 및 선택
        - 3·4번째 메시지
          - Diffie-Hellman 절반 키 교환
          - Nonce와 ID 함께 전송
          - Nonce와 ID는 수신자의 공개키로 암호화
          - 수신자는 자신의 개인키로 복호화
        - 5·6번째 메시지
          - HASH-I / HASH-R만 교환
          - 사전 공유 비밀 키 방법과 달리 ID는 이미 3·4번째 메시지에서 교환 완료

KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash
HDR: General header including cookies
$_R$ Encrypted with initiator's public key
$_I$ Encrypted with responder's public key
Encrypted with SKEYID_e



<그림 44> 주 모드에서의 본래의 공개 키 방법

인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 본래의 공개 키 방법 (3/3)
      - 한계
        - 공개키 암·복호화에 따른 높은 연산 부하
          - 메시지 교환 과정에서 공개키 암호화와 복호화를 수행
          - Initiator와 Responder 모두에게 계산 부하 발생
        - Initiator의 인증서 전송 문제
          - Initiator가 Responder의 공개키로 암호화된 자신의 인증서를 전송하기 어려움
          - 상대방 인증을 위한 인증서 활용에 제약

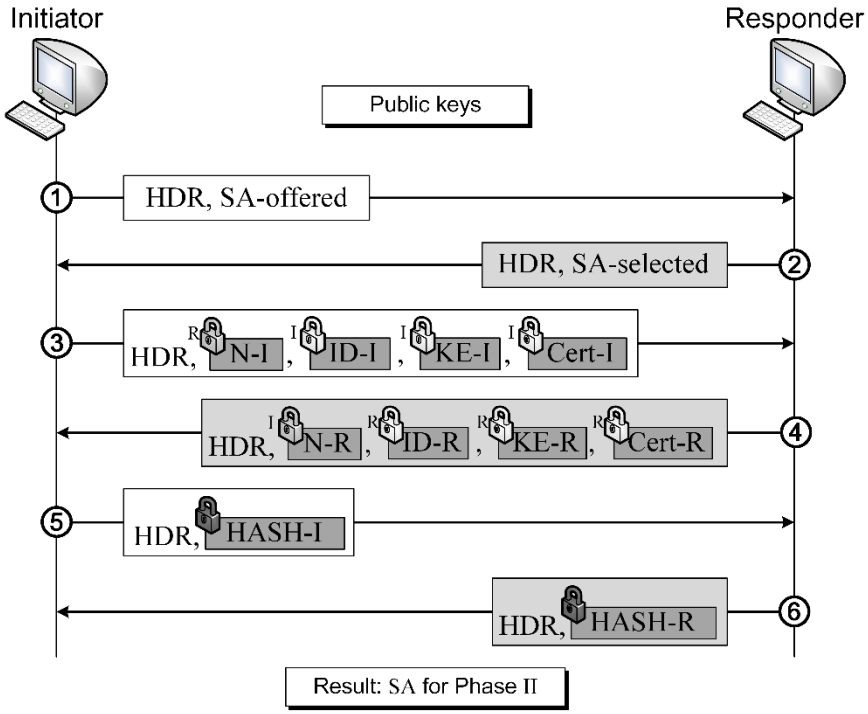
인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 개정된 공개 키 방법 (1/4)
      - 개요
        - 공개키를 임시 비밀 키 생성에 필요한 정보 보호에만 사용하도록 개선한 인증 방법
        - 본래의 공개키 방법의 높은 공개키 연산 부하와 인증서 전송 문제를 개선
      - 특징
        - 공개키 암·복호화의 사용 범위 축소를 통한 계산 부하 감소
        - 공개키로 보호한 Nonce를 정상적으로 복호화한 상대방만 동일한 임시 비밀 키 계산 가능
          - $\mathrm{K\text{-}I} = prf(\mathrm{N\text{-}I}, \mathrm{Cookie\text{-}I})$
          - $\mathrm{K\text{-}R} = prf(\mathrm{N\text{-}R}, \mathrm{Cookie\text{-}R})$
        - Initiator와 Responder의 인증서 전송 지원

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 개정된 공개 키 방법 (2/4)
      - 메시지 교환 구조 (1/2)
        - 1·2번째 메시지
          - 쿠키 교환
          - SA 매개변수 제안 및 선택
          - 본래의 공개키 방법과 동일
        - 3번째 메시지
          - N-I를 Responder의 공개키로 암호화
          - N-I + Cookie-I를 이용하여 임시 비밀 키 K-I 생성
          - K-I를 이용하여 KE-I, ID-I, 선택적 인증서(Cert-I) 보호

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
Cert-I (Cert-R): Indicator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash
$_R$ Encrypted with initiator's secret key
$_I$ Encrypted with responder's secret key
$_R$ Encrypted with initiator's public key
$_I$ Encrypted with responder's public key
Encrypted with SKEYID_e

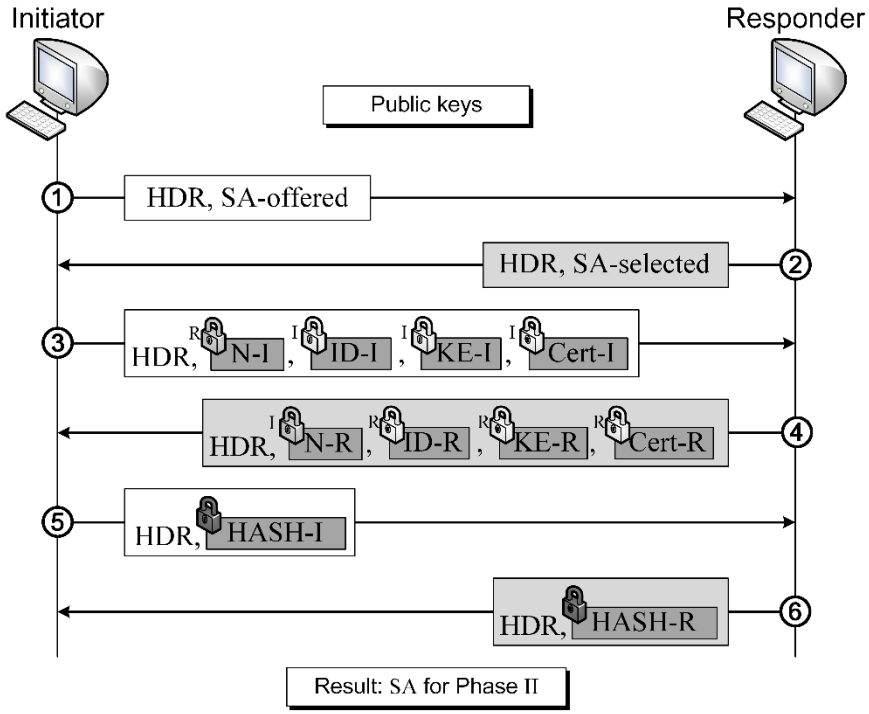


<그림 45> 주 모드에서의 개정된 공개 키 방법

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 개정된 공개 키 방법 (3/4)
      - 메시지 교환 구조 (2/2)
        - 4번째 메시지
          - N–R을 Initiator의 공개키로 암호화
          - N–R + Cookie–R을 이용하여 임시 비밀 키 K–R 생성
          - K–R을 이용하여 KE–R, ID–R, 선택적 인증서(Cert–R) 보호
        - 5·6번째 메시지
          - HASH–I / HASH–R 교환
          - SKEYID_e를 이용한 보호

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
Cert-I (Cert-R): Indicator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash
$_R$ Encrypted with initiator's secret key
$_I$ Encrypted with responder's secret key
$_R$ Encrypted with initiator's public key
$_I$ Encrypted with responder's public key
Encrypted with SKEYID_e



<그림 45> 주 모드에서의 개정된 공개 키 방법

인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 개정된 공개 키 방법 (4/4)
      - 한계
        - 공개키 기반 연산 부담
          - 본래의 공개키 방법보다 공개키 암·복호화 횟수가 줄었으나 공개키 연산 자체는 여전히 필요
          - 대량의 세션 설정 시 계산 비용 증가 가능
        - 공개키 및 인증서 관리 부담
          - Initiator와 Responder 간의 서로의 공개키를 신뢰 필요
          - 인증서를 사용하는 경우 발급·배포·검증·폐기 등의 관리 필요

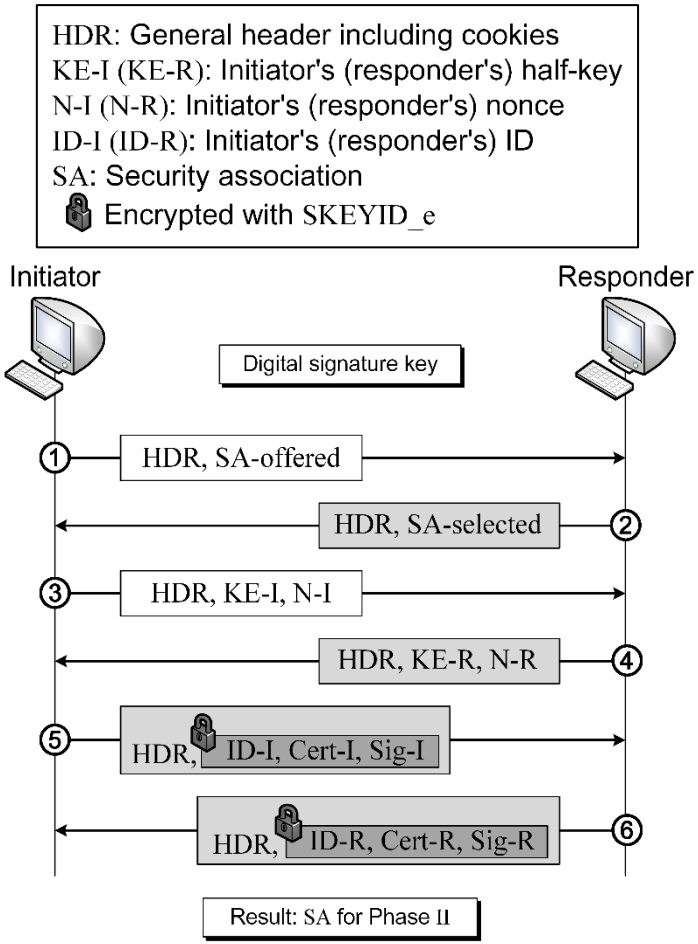
인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 디지털 서명 방법 (1/4)
      - 개요
        - 각 당사자가 자신의 개인키로 디지털 서명을 생성하고, 상대방이 공개키로 이를 검증하여 상호 인증하는 방법
        - 주 모드의 기본적인 6개 메시지 구조를 사용하며, SKEYID 계산 방식에서 사전 공유 비밀 키 방법과 차이 존재
        - 공개 키 방법의 상대 공개 키 사전 확보 문제를 인증서 교환으로 개선
      - 특징
        - 개인키를 이용한 서명 생성과 공개키를 이용한 서명 검증
        - 지금까지 교환된 메시지 정보에 대한 서명을 통한 상대방 인증 및 메시지 무결성 확인
        - 인증서(Certificate)의 선택적 전송
        - 사전 공유 비밀 키 없이 공개키 기반의 상호 인증 가능

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 디지털 서명 방법 (2/4)
      - 메시지 교환 구조 (1/2)
        - 1·2번째 메시지
          - 쿠키 교환
          - SA 매개변수 제안 및 선택
        - 3·4번째 메시지
          - Diffie-Hellman 절반 키 $\mathrm{KE-I}$, $\mathrm{KE-R}$ 교환
          - Nonce $\mathrm{N-I}$, $\mathrm{N-R}$ 교환
          - 교환 정보를 이용하여 SKEYID 및 관련 키 계산

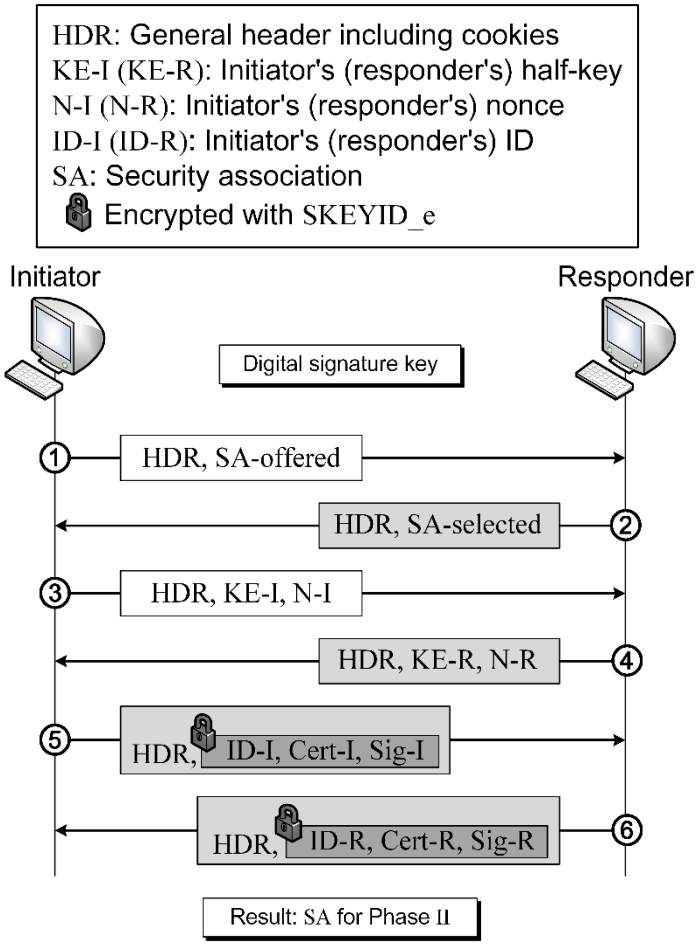
$$\mathrm{SKEYID} = prf(\mathrm{N-I} \mid \mathrm{N-R}, g^{ir})$$



<그림 46> 주 모드에서의 디지털 서명 방법

인터넷 키 교환(IKE)

- Phase I
  - 주 모드(Main Mode)
    - 디지털 서명 방법 (3/4)
      - 메시지 교환 구조 (2/2)
        - 5번째 메시지
          - ID–I, 선택적 Cert–I
          - Sig–I: Initiator가 1~4번째 메시지에서 교환된 정보에 자신의 개인키로 생성한 서명
          - Responder가 Initiator의 공개키로 Sig–I 검증
          - 검증 성공 시 Initiator 인증
        - 6번째 메시지
          - ID–R, 선택적 Cert–R
          - Sig–R: Responder가 지금까지 교환된 정보에 자신의 개인키로 생성한 서명
          - Initiator가 Responder의 공개키로 Sig–R 검증
          - 검증 성공 시 Responder 인증



<그림 46> 주 모드에서의 디지털 서명 방법

인터넷 키 교환(IKE)

- Phase Ⅰ
  - 주 모드(Main Mode)
    - 디지털 서명 방법 (4/4)
      - 한계
        - 디지털 서명 생성·검증에 따른 공개키 연산 부담
          - 세션 설정마다 양 당사자의 서명 생성 및 검증 필요
        - 공개키 및 인증서 관리 부담
          - 상대방 공개키의 신뢰성 확인 필요
          - 인증서 사용 시 발급·검증·폐기 등의 관리 필요
        - 개인키 유출에 따른 신원 위조 위험
          - 장기 개인키가 노출될 경우 공격자가 해당 당사자의 정상 서명을 위조할 가능성 증가

인터넷 키 교환(IKE)

- Phase Ⅰ
  - 공격적 모드(Aggressive Mode)
    - 개요
      - 주 모드의 메시지 교환을 압축한 모드
        - 주 모드의 6개 메시지를 3개의 메시지로 결합하여 교환
      - 주 모드와 동일한 SA 설정 및 인증 개념 사용
    - 필요성
      - 메시지 수를 줄여 SA 설정 시간 단축
        - 왕복 횟수가 절반으로 줄어 빠른 연결 설정 가능
      - 사전 공유 키 방법의 IP 주소 기반 키 선택 문제 해결
        - 첫 메시지에 ID를 포함하여 IP가 바뀌는 원격 사용자도 사전 공유 키 사용 가능

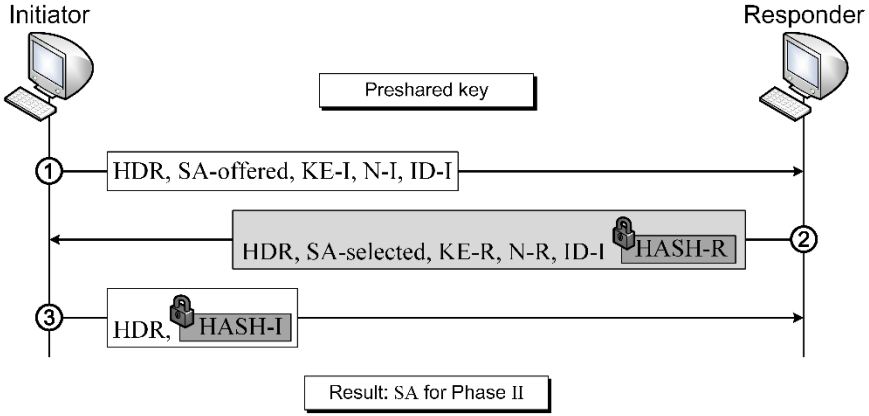
인터넷 키 교환(IKE)

- Phase Ⅰ
  - 공격적 모드(Aggressive Mode)
    - 메시지 교환 구조
      - 1번째 메시지
        - 주 모드의 1번째, 3번째 메시지를 결합
        - SA 제안 및 Initiator의 키 교환 정보 전달
      - 2번째 메시지
        - 주 모드의 2번째, 4번째, 6번째 메시지를 결합
        - SA 선택 및 Responder의 키 교환 정보 및 인증 정보 전달
      - 3번째 메시지
        - 주 모드의 5번째 메시지
        - Initiator의 인증 정보 전달 및 상호 인증 완료

인터넷 키 교환(IKE)

- Phase I
  - 공격적 모드(Aggressive Mode)
    - 사전공유 비밀 키 방법
      - 특징
        - 주 모드의 사전 공유 비밀 키 방식을 3개 메시지로 압축
        - Responder가 Initiator보다 먼저 SKEYID 및 인증 값 계산 가능
      - 메시지 교환 구조
        - 1번째 메시지
          - SA 제안 및 KE-I, N-I, ID-I 전달
        - 2번째 메시지
          - SA 선택 및 KE-R, N-R, ID-R, HASH-R 전달
          - Responder는 1번째 메시지를 받은 후 SKEYID와 HASH-R 계산 가능
        - 3번째 메시지
          - Initiator의 HASH-I 전달
          - HASH-I는 SKEYID_e로 암호화 가능

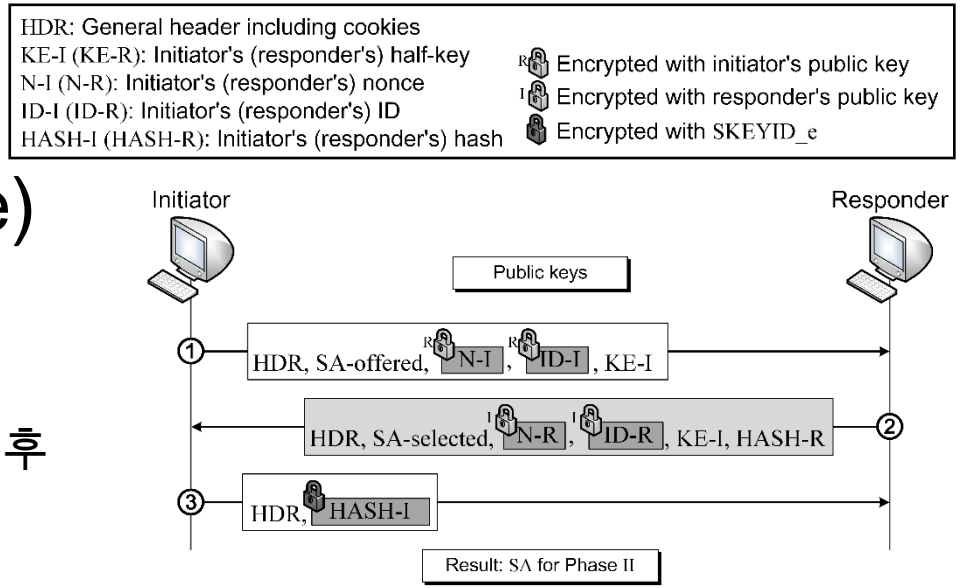
KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
HASH-I (HASH-R): Initiator's (responder's) hash
HDR: General header including cookies
ID-I (ID-R): Initiator's (responder's) ID
Encrypted with SKEYID_e



<그림 47> 공격적 모드에서의 사전공유 비밀 키 방법

인터넷 키 교환(IKE)

- Phase I
  - 공격적 모드(Aggressive Mode)
    - 본래의 공개 키 방법
      - 특징
        - Responder는 1번째 메시지 수신 후 SKEYID와 HASH-R 계산 가능
        - Initiator는 2번째 메시지를 받은 후에 SKEYID 계산 가능
      - 공개키를 이용한 정보 보호 방식은 주 모드의 본래 공개키 방법과 동일
    - 메시지 교환 구조
      - 1번째 메시지
        - SA 제안과 Initiator의 키 교환·인증 관련 정보 전달
      - 2번째 메시지
        - SA 선택과 Responder의 키 교환·인증 정보 및 HASH-R 전달
        - Responder는 1번째 메시지를 받은 후 SKEYID와 HASH-R 계산 가능
      - 3번째 메시지
        - Initiator의 HASH-I 전달



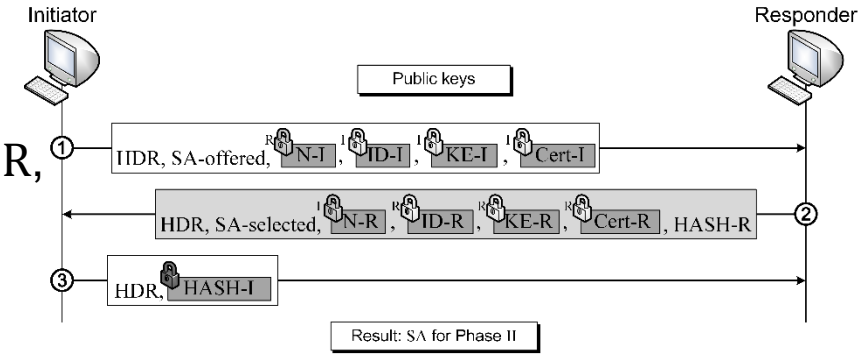
<그림 48> 공격적 모드에서의 본래의 공개 키 방법

인터넷 키 교환(IKE)

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
Cert-I (Cert-R): Indicator's (responder's) certificate
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
HASH-I (HASH-R): Initiator's (responder's) hash

Encrypted with initiator's secret key
Encrypted with responder's secret key
Encrypted with initiator's public key
Encrypted with responder's public key
Encrypted with SKEYID_e

- Phase I
  - 공격적 모드(Aggressive Mode)
    - 개정된 공개 키 방법
      - 특징
        - 주 모드의 개정된 공개키 방법과 동일한 인증 구조
        - 주 모드의 여러 메시지를 3개의 메시지로 결합하여 교환
      - 메시지 교환 구조
        - 1번째 메시지
          - 주 모드의 1·3번째 메시지 결합
          - SA 제안 및 N–I, ID–I, KE–I, Cert–I 전달
        - 2번째 메시지
          - 주 모드의 2·4·6번째 메시지 결합
          - SA 선택 N–R, ID–R, KE–R, Cert–R, HASH–R 전달
        - 3번째 메시지
          - 주 모드의 5번째 메시지 전송
          - HASH–I 전달

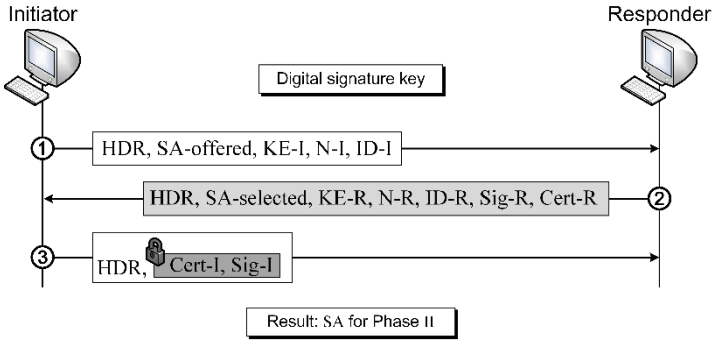


<그림 49> 공격적 모드에서의 개정된 공개 키 방법

인터넷 키 교환(IKE)

HDR: General header including cookies
Sig-I (Sig-R): Initiator's (responder's) signature
Cert-I (Cert-R): Initiator's (responder's) certificate
KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
Encrypted with SKEYID_e

- Phase I
  - 공격적 모드(Aggressive Mode)
    - 디지털 서명 방법
      - 특징
        - 주 모드의 디지털 서명 방법과 동일한 인증 구조
        - 주 모드의 여러 메시지를 3개의 메시지로 결합하여 교환
      - 메시지 교환 구조
        - 1번째 메시지
          - 주 모드의 1·3번째 메시지, 5번째 메시지의 ID-I 결합
          - SA 제안 및 KE-I, N-I, ID-I 전달
        - 2번째 메시지
          - 주 모드의 2·4·6번째 메시지 결합
          - SA 선택 및 KE-R, N-R, ID-R, Sig-R, Cert-R 전달
        - 3번째 메시지
          - 주 모드의 5번째 메시지에서 남은 정보
          - Sig-I, Cert-I 전달



<그림 50> 공격적 모드에서의 디지털 서명 방법

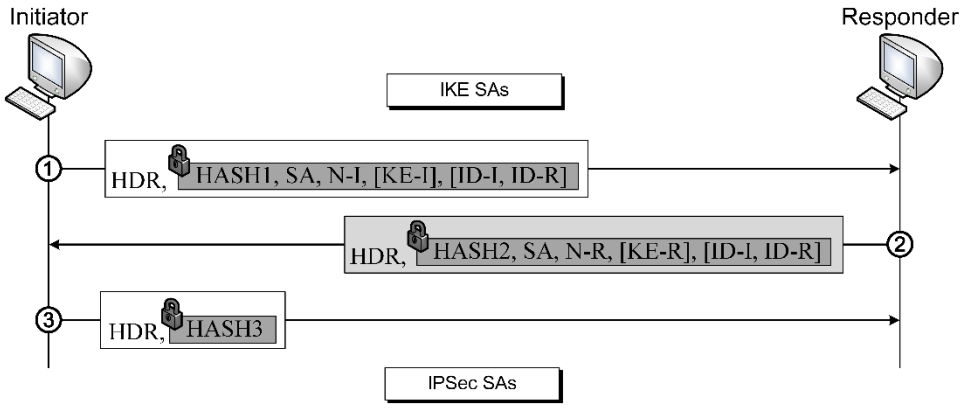
인터넷 키 교환(IKE)

- Phase II
  - 신속 모드(Quick Mode) (1/5)
    - 개요
      - Phase I에서 생성된 IKE SA를 이용하여 IPSec SA를 생성하는 모드
      - 3개의 메시지 교환으로 구성
      - Phase II의 Initiator는 Phase I의 Initiator와 Responder 모두 가능
    - 메시지 보호
      - 3개의 메시지 모두 Phase I에서 생성한 SKEYID_e로 암호화
      - HASH1, HASH2, HASH3는 SKEYID_a를 이용하여 계산
        - $\mathrm{HASH1} = prf(\mathrm{SKEYID_a}, \mathrm{MsgID} \mid \mathrm{SA} \mid \mathrm{N-I})$
        - $\mathrm{HASH2} = prf(\mathrm{SKEYID_a}, \mathrm{MsgID} \mid \mathrm{SA} \mid \mathrm{N-R})$
        - $\mathrm{HASH3} = prf(\mathrm{SKEYID_a}, 0 \mid \mathrm{MsgID} \mid \mathrm{SA} \mid \mathrm{N-I} \mid \mathrm{N-R})$
      - Message ID(MsgID)를 사용하여 동시에 진행되는 여러 Phase II 교환을 구분

인터넷 키 교환(IKE)

- Phase II
  - 신속 모드(Quick Mode) (2/5)
    - 메시지 교환 구조
      - 1번째 메시지
        - 새로운 SA 제안
        - N-I 전달
        - 선택적 KE-I, ID-I, ID-R 전달
        - HASH1을 통한 메시지 인증
      - 2번째 메시지
        - SA 선택
        - N-R 전달
        - 필요 시 KE-R, ID 전달
        - HASH2를 통한 메시지 인증
      - 3번째 메시지
        - HASH3 전달
        - 앞선 메시지 교환에 대한 확인 및 인증 완료

HDR: General header including cookies
KE-I (KE-R): Initiator's (responder's) half-key
N-I (N-R): Initiator's (responder's) nonce
ID-I (ID-R): Initiator's (responder's) ID
SA: Security association
Encrypted with SKEYID_e



<그림 51> 신속 모드

인터넷 키 교환(IKE)

- Phase II
  - 신속 모드(Quick Mode) (3/5)
    - 키 재료 생성 (1/3)
      - 완전한 전방 보안(PFS, Perfect Forward Secrecy)
        - 선택적으로 신속 모드에서 새로운 Diffie-Hellman 키 교환 수행
        - 각 신속 모드에서 새로운 Diffie-Hellman 공유 비밀을 생성하여 이전 키 노출이 이후 키에 미치는 영향 감소
          - PFS 미적용 시 모든 IPsec 키가 SKEYID_d로부터 생성되므로, SKEYID_d 유출 시 해당 IKE SA로 생성된 모든 키 노출
          - 공격자가 저장해 둔 과거 트래픽도 유출된 키로 복호화 가능
          - 세션마다 독립된 Diffie-Hellman 비밀을 사용하여 키 유출 피해를 해당 세션으로 한정

인터넷 키 교환(IKE)

- Phase II
  - 신속 모드(Quick Mode) (4/5)
    - 키 재료 생성 (2/3)
      - 키 재료(Key Material)
        - 교환된 SA와 비표 등을 기반으로 실제 AH/ESP에서 사용할 키 생성
        - PFS 적용 여부에 따라 키 재료 계산 방식 구분
          - PFS 미적용: $\mathrm{K} = prf(\mathrm{SKEYID_d}, \mathrm{protocol} \mid \mathrm{SPI} \mid \mathrm{N_I} \mid \mathrm{N_R})$
          - PFS 적용: $\mathrm{K} = prf(\mathrm{SKEYID_d}, \mathrm{g^{ir}} \mid \mathrm{protocol} \mid \mathrm{SPI} \mid \mathrm{N_I} \mid \mathrm{N_R})$
            - 새로운 Diffie-Hellman 공유 비밀 g^{ir}을 추가하여 키 재료 생성
        - IPSec 키는 송신 측에서 수신 측으로의 단방향으로 생성
        - 양방향 IPSec 통신에서는 각 방향별 서로 다른 키 재료 생성

인터넷 키 교환(IKE)

- Phase II
  - 신속 모드(Quick Mode) (5/5)
    - 키 재료 생성 (3/3)
      - 키 길이 확장
        - 한 번의 PRF 출력으로 필요한 키 길이를 충족하지 못할 경우 이전 키 값을 이용하여 키 재료를 연속 생성
        - 생성된 키 값을 연결하여 최종 키 재료 구성
          - $\mathrm{K}_1 = prf(\mathrm{SKEYID_d}, \mathrm{protocol} \mid \mathrm{SPI} \mid \mathrm{N_I} \mid \mathrm{N_R})$
            $\mathrm{K}_2 = prf(\mathrm{SKEYID_d}, \mathrm{K}_1 \mid \mathrm{protocol} \mid \mathrm{SPI} \mid \mathrm{N_I} \mid \mathrm{N_R})$
            $\mathrm{K}_3 = prf(\mathrm{SKEYID_d}, \mathrm{K}_2 \mid \mathrm{protocol} \mid \mathrm{SPI} \mid \mathrm{N_I} \mid \mathrm{N_R})$
            ...
            $\mathrm{K} = \mathrm{K}_1 \mid \mathrm{K}_2 \mid \mathrm{K}_3 \mid \ldots$

인터넷 키 교환(IKE)

- SA 알고리즘 (1/2)
  - 개요
    - SA 협상 과정에서 사용할 Diffie-Hellman 그룹, 해시 알고리즘, 암호화 알고리즘을 선택
  - Diffie-Hellman 그룹
    - 절반 키 교환을 위해 사용

[표 6] SA에서 사용하는 Diffie-Hellman 그룹

그룹 번호	설명
1	768비트 모듈러스를 사용하는 모듈러 지수승 그룹
2	1024비트 모듈러스를 사용하는 모듈러 지수승 그룹
3	155비트 필드 크기를 사용하는 타원곡선 그룹
4	185비트 필드 크기를 사용하는 타원곡선 그룹
5	1536비트 모듈러스를 사용하는 모듈러 지수승 그룹

인터넷 키 교환(IKE)

- SA 알고리즘 (2/2)
  - 해시 알고리즘
    - 인증을 위해 [표 7]의 알고리즘 사용
  - 암호화 알고리즘
    - 기밀성을 위해 [표 8]의 알고리즘 사용
    - CBC (Cipher Block Chaining) 방식으로 사용

[표 7] SA에서 사용하는 해시 알고리즘

번호	해시 알고리즘
1	MD5
2	SHA
3	Tiger
4	SHA2-256
5	SHA2-384
6	SHA2-512

[표 8] SA에서 사용하는 암호화 알고리즘

번호	암호화 알고리즘
1	DES
2	IDEA
3	Blowfish
4	RC5
5	3DES
6	CAST
7	AES

Thanks!

송 민 희(minhee@pel.sejong.ac.kr)