네트워크 보안

2023.07.14 하계 워크샵 2주차

한림대학교 정보과학대학 씨애랑

(HALLYM SECURITY TEAM SHIELD)

목차

- TLS
  - ✓ Backgrounds
  - ✓ TLS 란 ?
  - ✓ IPSec
  - ✓ VPN

BACKGROUNDS

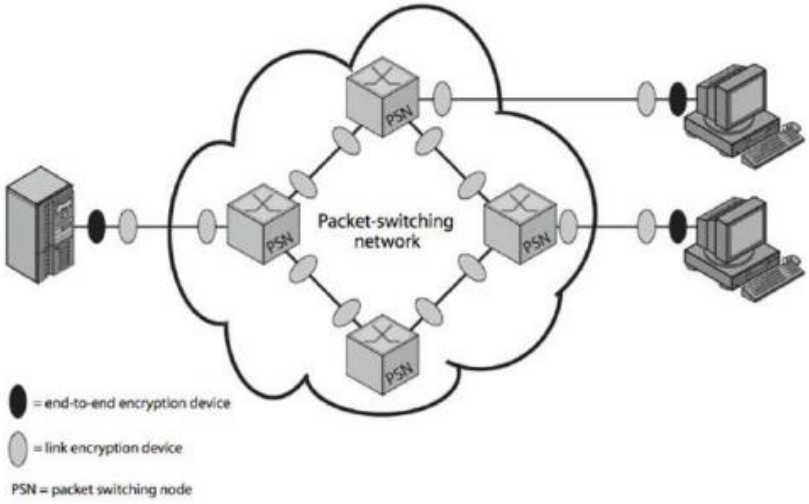
Backgrounds

■ Link encryption

- □ 각 링크마다 암호화가 진행되는 것
- □ 링크 사이의 공유된 암호화 키가 필요함

■ End-to-End encryption

- □ 암호화가 Source와 Destination에서 이루어짐



Backgrounds

■ **End-to-End 암호화에서 헤더 부분 (e.g., IP address)은 암호화 되지 않음**

□ 헤더 정보를 통해 인터넷 패킷들을 라우팅 할 수 있음

- 하지만, 헤더 정보가 암호화 되지 않으므로 traffic의 흐름이 노출됨

■ **따라서 End-to-End 암호화와 Link 암호화가 동시에 사용되기도 함**

□ End-to-End 암호화는 전송하는 데이터를 보호함

□ Link 암호화는 traffic 흐름을 숨길 수 있음

- Traffic흐름 노출을 방지하기 위해 Tor 혹은 VPN(Virtual Private Network)을 사용해야 함

■ **인증을 위해서는 메시지 인증 코드(MAC)와 전자 서명(Digital Signature)기술이 사용됨**

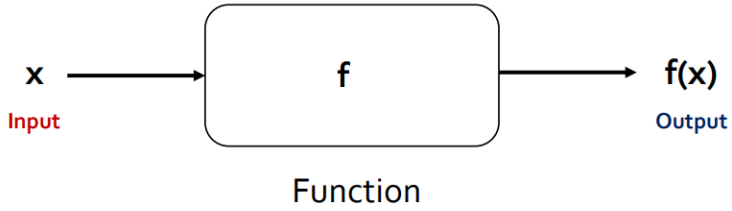
□ 개체 인증 (Entity Authentication)에는 주로 전자서명 (Digital Signature) 기술이 활용됨

- E.g., 크롬 브라우저와 네이버 서버의 인증

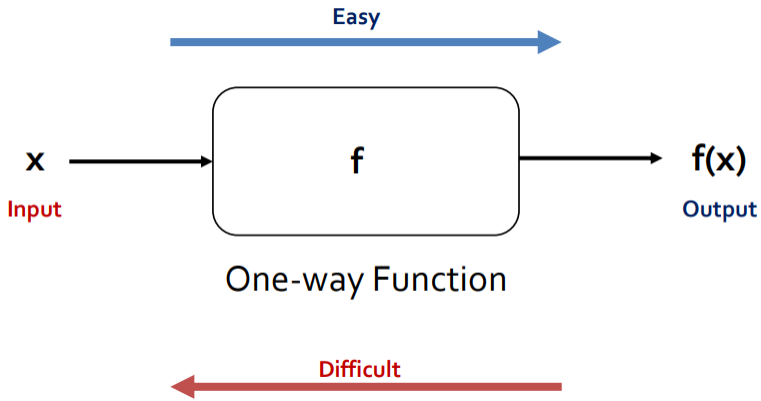
□ 평문 메세지 인증 (Message Authentication)에는 주로 MAC 이 사용됨

Backgrounds

■ 함수 (Function) 이란?

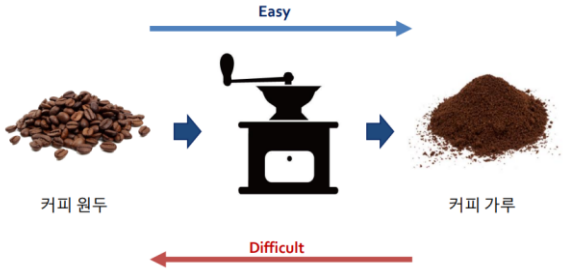


■ 일 방향 함수 (One-way function) 이란?



Backgrounds

■ 일 방향 함수 (One-way function) 예시



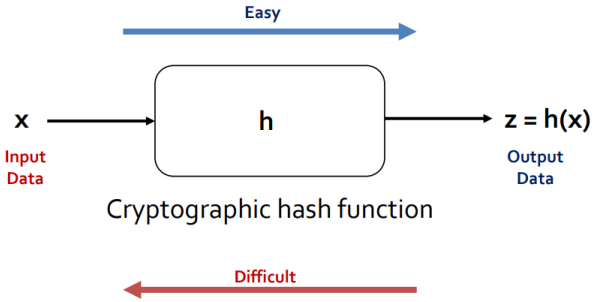
■ Hash function (해시 함수)

□ Input data를 x, hash function 를 h, output을 z라고 할 때,

- x는 z의 프리이미지 (preimage)라고도 함
- z는 이미지 (image)라고 함

□ 이러한 경우, 해시 함수 h는 일대일의 함수가 아니기 때문에,

- 동일한 해시 값 z를 가질 수 있는 Input data x가 여러 개 존재할 수 있음



Backgrounds

■ 해시 함수의 조건

- □ 프리 이미지 저항성 (Preimage resistance)
- □ 제 2 프리 이미지 저항성 (Second preimage resistance)
- □ 충돌 저항성 (Collision resistance)

■ 눈사태 효과 (Avalanche effect)

- □ 암호 알고리즘에서 입력 값에 미세한 변화를 줄 경우 출력 값에 상당한 변화가 일어나는 성질
  - 해시 함수에서의 눈사태 효과 (Avalanche effect)
    - 해시 함수에서 Input Data가 1 bit만 차이를 보여도 Output Data는 다르게 나타남.

Hash Algorithm Types	Input Data	Hash values
SHA1	1	356A192B7913B04C54574D18C28D46E6395428AB
	2	DA4B9237BACCCDF19C0760CAB7AEC4A8359010B0
	3	77DE68DAECD823BABBB58EDB1C8E14D7106E83BB
	4	1B6453892473A467D07372D45EB05ABC2031647A

Backgrounds

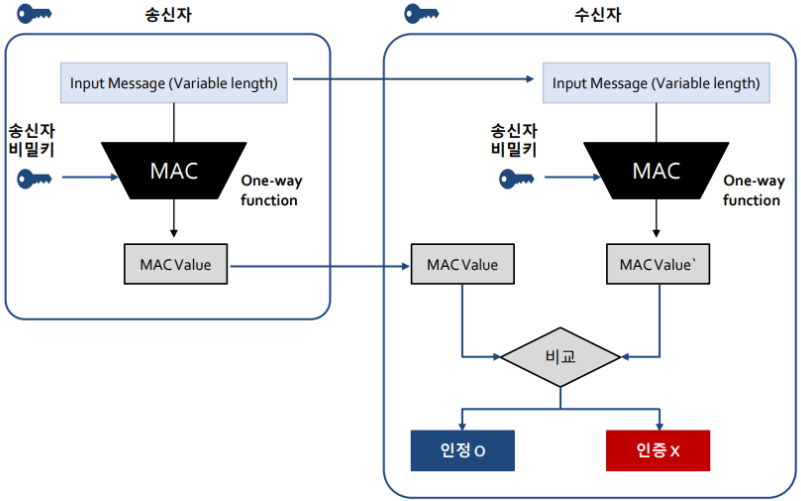
■ Message Authentication Code (MAC)

□ 해시 알고리즘으로 정상 메시지의 수정 또는 변경을 검출할 수 있도록 덧붙이는 코드

- 하지만, 거짓 행세를 검출하는 것은 불가능
- 무결성 외에 "인증" 이라는 절차가 필요하게 됨.

□ 오리지널 값을 데이터에 덧붙여서 확인하도록 하는 것 이 필요

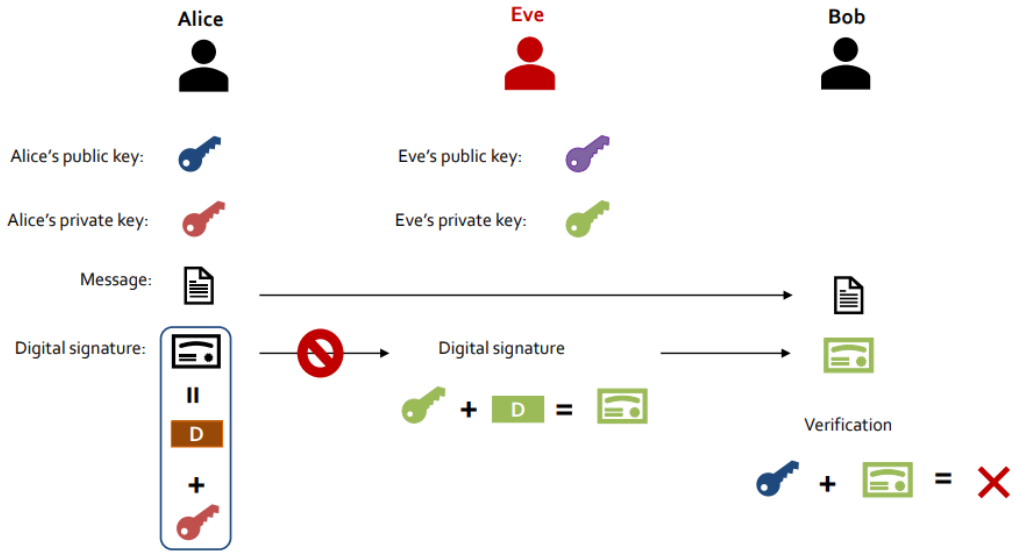
- 공격자 입장에서 변조된 데이터에 대해서 MAC 을 생성하여 MAC 도 바꿔치기 할 가능성이 있음
  - MAC 의 생성과 검증은 공유키를 사용하여 수행



Backgrounds

■ 전자 서명 (Digital signature)

- □ 인증 (Authentication) : 전자 서명을 통해 서명자를 검증할 수 있음
- □ 메시지 무결성 (Integrity) : 메시지는 서명 후에 변경될 수 없음
- □ 부인방지 (Non-repudiation) : 발신자는 서명한 사실을 부인할 수 없음
- □ 전자 서명 알고리즘의 종류 : RSA, ECDSA, …



Backgrounds

■ 암호학 소개

□ 암호 알고리즘은 다음과 같이 세 개의 알고리즘 ($\boldsymbol{G}$, $\boldsymbol{E}$, $\boldsymbol{D}$)로 구분됨

- 키 생성 (Key Generation) 알고리즘 G
  - 키 생성 알고리즘 G는 가능한 키들의 집합 $\boldsymbol{K}$에서 암호화 키 $\boldsymbol{k1}$와 복호화 키 $\boldsymbol{k2}$를 선택함
  - 이 때, 집합 $\boldsymbol{K}$를 키 공간(Key Space)이라 함
  - 한 개의 키가 집합 $\boldsymbol{K}$에서 선택될 확률은 키 공간의 분포에 따라 상이하지만,
    - 일반적으로 균일분포를 따르는 확률 $\frac{1}{|k|}$을 가지 는 난수 (Uniformly Distributed Random Number)를 선택함
  - ex) 4 bit 길이의 키를 사용하는 경우 키 공간은 집합 {0, 1, . . . , 15} (2의 4승 이므로 16개) 이며,
    - 균일분포를 따라 이 중 하나의 값이 키로 선택될 확률은 $\frac{1}{16}$ 으로 볼 수 있음

Backgrounds

■ 암호 알고리즘

□ 암호 알고리즘은 다음과 같이 세 개의 알고리즘 ($\boldsymbol{G}$, $\boldsymbol{E}$, $\boldsymbol{D}$)로 구분됨

- 암호화 (Encryption) 알고리즘 E
  - 암호화 알고리즘 $\boldsymbol{E}$는 암호화 키 $\boldsymbol{k1} \in \boldsymbol{K}$를 사용하여 평문 $\boldsymbol{m}$을 입력으로 받아 암호문 $\boldsymbol{c}$를 출력함.

$$E_{k_1}(m) = c$$

- 복호화 (Decryption) 알고리즘 D
  - 복호화 키 $k2 \in K$를 사용하여 암호문 $\boldsymbol{c} \in \boldsymbol{C}$ 를 복호화하면 메시지 $m \in M$을 얻어낼 수 있음

$$D_{k_2}(c) = m$$

- 암호 알고리즘이 올바르게 설계되었다면 평문 $\boldsymbol{m}$의 암호문을 복호화 했을 때 평문 $\boldsymbol{m}$을 얻을 수 있어야 함.

즉, $D_{k_2}(E_{k_1}(m)) = m$ 수식을 만족해야 함

Backgrounds

■ 암호 알고리즘

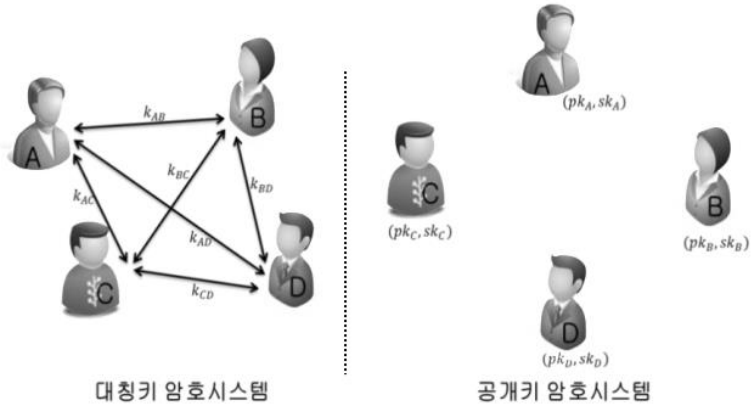
□ 이러한 상황에서, 암호 알고리즘은 암호화 키와 복화화 키가 1. 같은 경우와 2. 다른 경우로 구분됨

- 암호 알고리즘이 동일한 암호화 키와 복호화 키를 사용하는 경우,
  - 즉, $\boldsymbol{k1} = \boldsymbol{k2}$인 경우, 대칭키 알고리즘(Symmetric Key Algorithm)이라 함
- 만약 암호 알고리즘이 다른 키($\boldsymbol{k1} \neq \boldsymbol{k2}$)를 사용하는 경우,
  - 비대칭키 알고리즘(Asymmetric Algorithm) 또는 공개키 알고리즘(Public Key Algorithm)이라 함

- 대칭키 알고리즘은 암호화되는 평문의 크기에 따라 블록 암호(Block Cipher)와 스트림 암호(Stream Cipher)로 분류됨.

□ 대칭키 암호

- 안전한 채널을 통해서 사용자가 서로 동일한 키를 사전 공유
- 송신자나 수신자의 부인방지를 제공하지 못함.

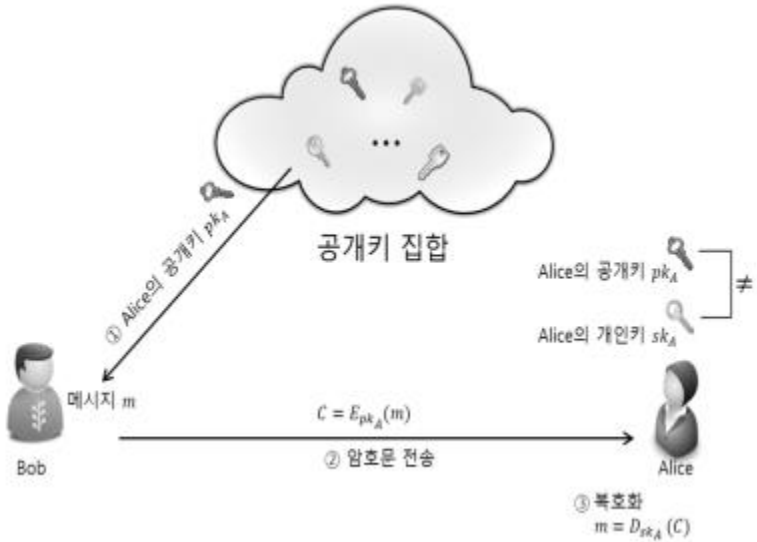


대칭키 암호시스템 공개키 암호시스템

Backgrounds

■ 공개키 (or 비대칭키(Asymmetric) 암호시스템)

- □ 각 사람마다 한 쌍의 키(공개키 pk, 개인키 sk)
- □ 공개키는 모두에게 공개되고, 개인키는 비밀로 보관
  - 공개키 pk로부터 개인키 sk를 도출하는 것은 계산적으로 불가능 (Computationally Infeasible)



Backgrounds

■ 공개키 및 대칭키 암호시스템의 차이점

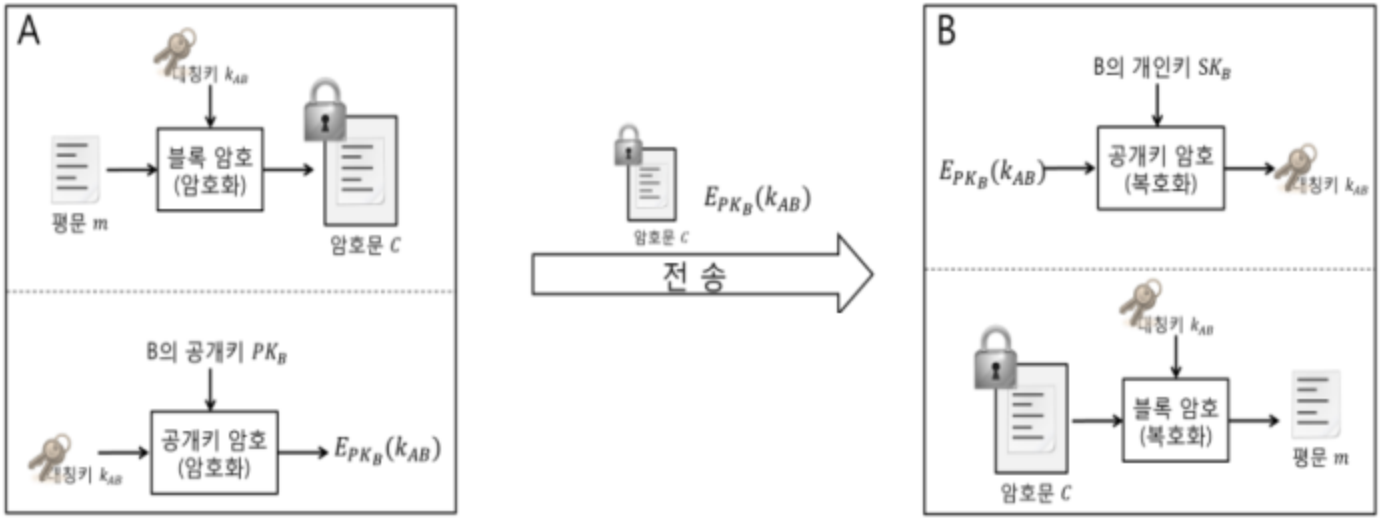
	대칭키 암호시스템	공개키 암호시스템
비밀키 분배 및 안전한 전달 시스템	필요	불필요
개인별 보유 비밀키 개수 (n명이 비밀통신 하는 경우)	$(n-1)$개 (상대방별로 키가 필요)	1개 (자신의 비밀키만 보유)
암호화 & 복호화 속도	빠름	느림
대표 예	DES, AES, SEED, ARIA	RSA, ElGamal

Backgrounds

■ 하이브리드 암호시스템

□ 대용량의 데이터를 암호화하기 위해서 대칭키 암호 시스템에서 사용되는 비밀키 k 를

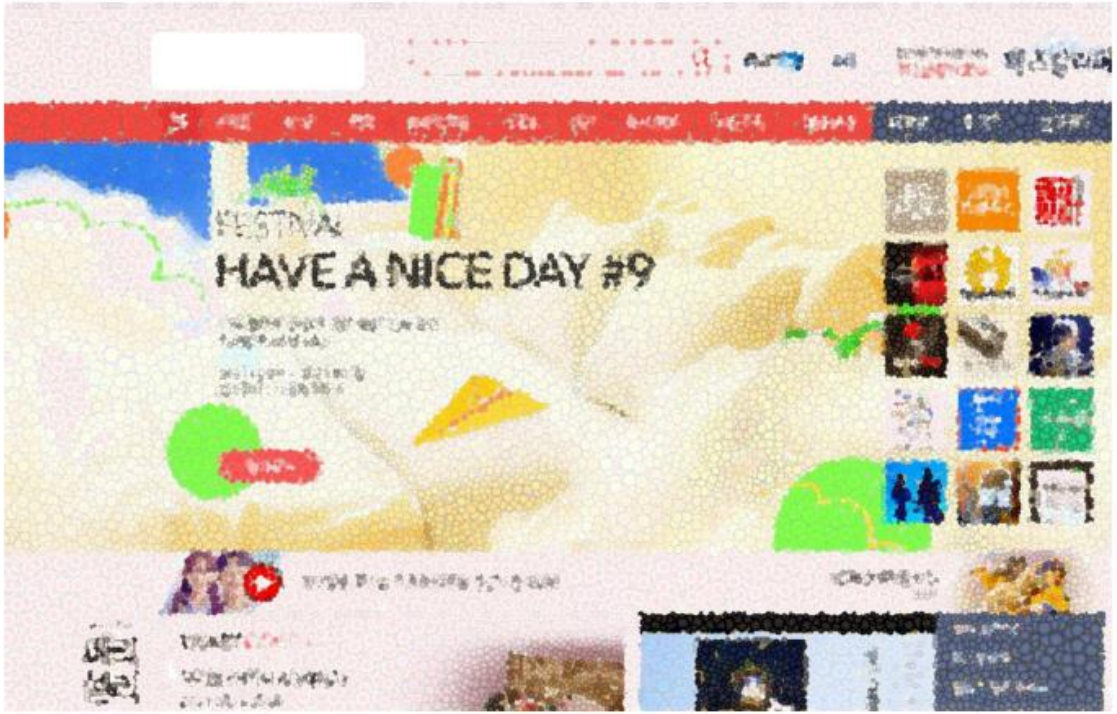
- 공개키 암호시스템으로 암호화(Epk (비밀키 $\boldsymbol{k}$))하여 분배하고,
  - 수신자는 분배된 비밀키 k를 이용하여 대용량의 데이터를 대칭 키 암호시스템으로 암호화



TLS

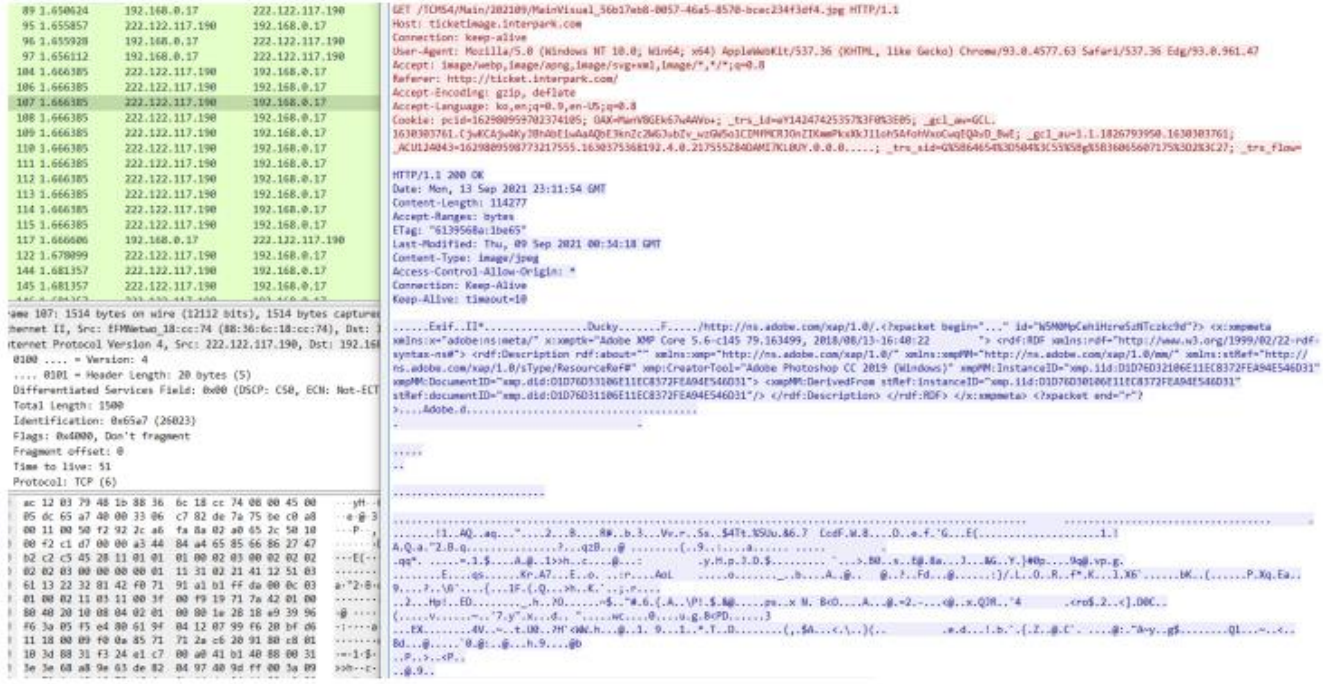
TLS

■ 일반적인 HTTP를 사용한 인터넷 접속



TLS

- **일반적인 HTTP를 사용한 인터넷 접속**
  - 최근에는 대부분의 사이트에서 HTTPS를 사용하고 있지만,
    - 일부 사이트에서는 로그인 페이지 대해서만 HTTPS를 사용

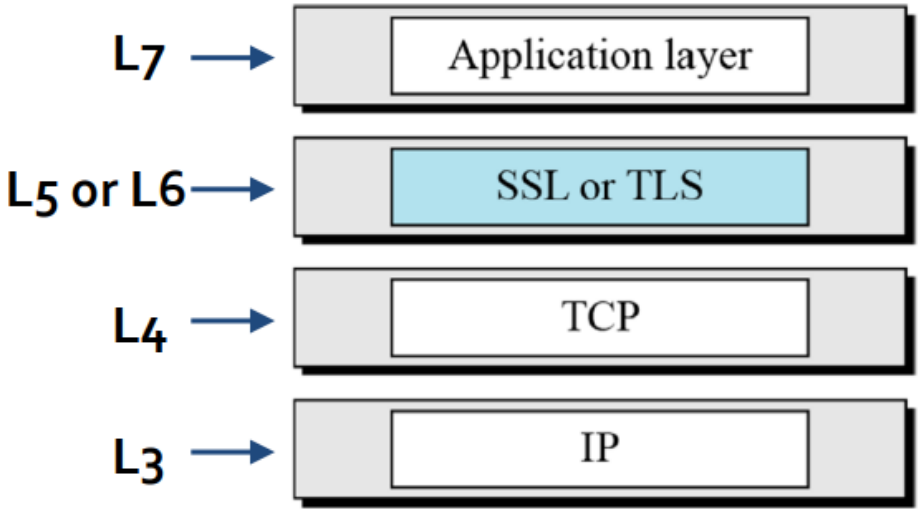


TLS

■ **TLS (Transport Layer Security)**

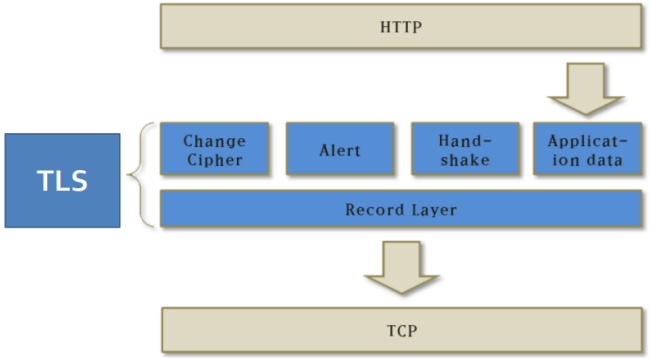
□ 클라이언트와 서버사이의 통신과정에 메시지 인증과 메시지 기밀성 (End-to-End 암호화)를 제공함.

- OSI7계층에서 5, 6계층 (Session layer, Presentation layer)으로 분류됨.



TLS

■ **TLS Architecture**



■ **Handshake protocol**
- □ 암호화/인증 통신을 위해 사용할 알고리즘의 종류 그리고 암호화/ 인증 키에 대해 협의하는 프로토콜

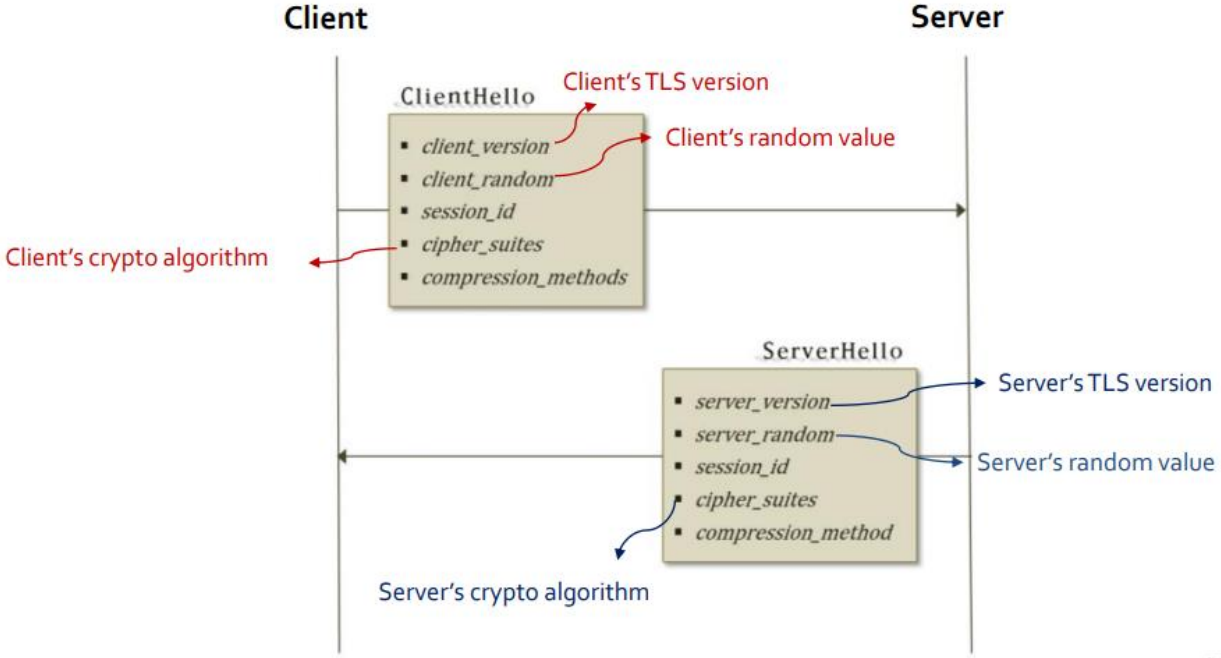
■ **ChangeCipherSpec protocol**
- □ Handshake 프로토콜을 통해 교환된 정보들을(e.g., 암호 알고리즘 종류, 암호화 키 등) 확인하는 프로토콜

■ **Alert protocol**
- □ TLS 과정 중에 발생하는 에러 혹은 비정상적인 상황을 리포트하는 프로토콜

■ **Record protocol**
- □ TLS에서 전송되는 상위 레이어의 메시지를 처리하는 프로토콜

TLS - Handshake 프로토콜

■ **Phase 1 : 단계 1에서는 클라이언트와 서버간 세션에 사용 할 보안 알고리즘을 협의함**



TLS - Handshake 프로토콜

■ **Phase 2 : 단계 2에서 서버는 인증서와 키 교환 메시 지를 전송하며,**

□ 필요할 경우 클라이언트로부터 인 증서 요청 메시지를 전송함

□ 단계 2 과정이 완료되면,

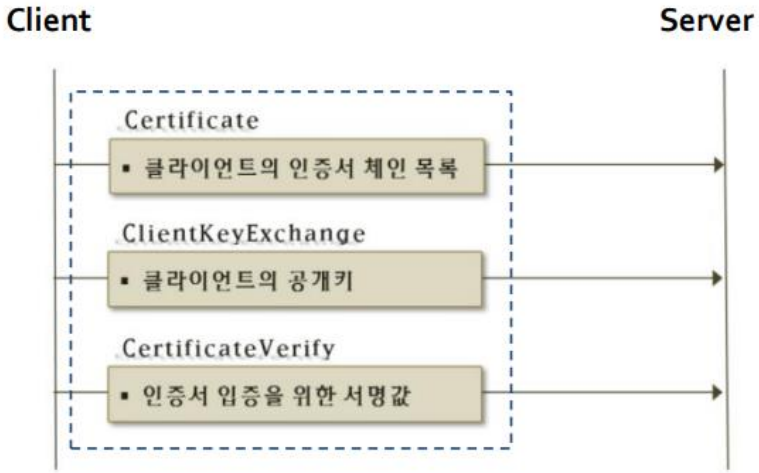
- 클라이언트는 전송받은 인증서 를 통해 서버를 인증하게 되고, 클라이언트는 서버의 공개키를 알게 됨

TLS - Handshake 프로토콜

■ **Phase 3 : 단계 3은 서버가 클라이언트를 인증하기 위해서 필요한 과정임**

□ 단계 3이 완료되면 서버는 클라이언트를 인증할 수 있게 되고, 서버와 클라이언트는 pre-master secret 공유

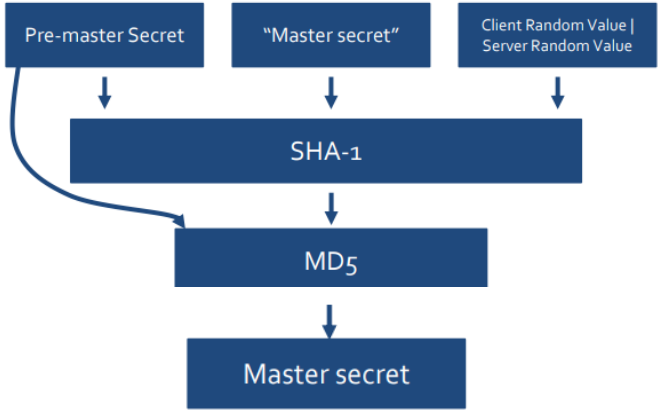
- Pre-master secret를 통해 암호화 키 (master secret) 유도



TLS : pre-master secret

■ **서버의 Random Value와 클라이언트의 Random Value를 조합하여 pre-master secret을 생성함**

□ 또한, 생성된 pre-master secret를 이용하여 master secret (비밀키로 이용) 생성



■ **Client에서 pre-master secret을 이용하여 mastersecret (48bytes)를 생성**

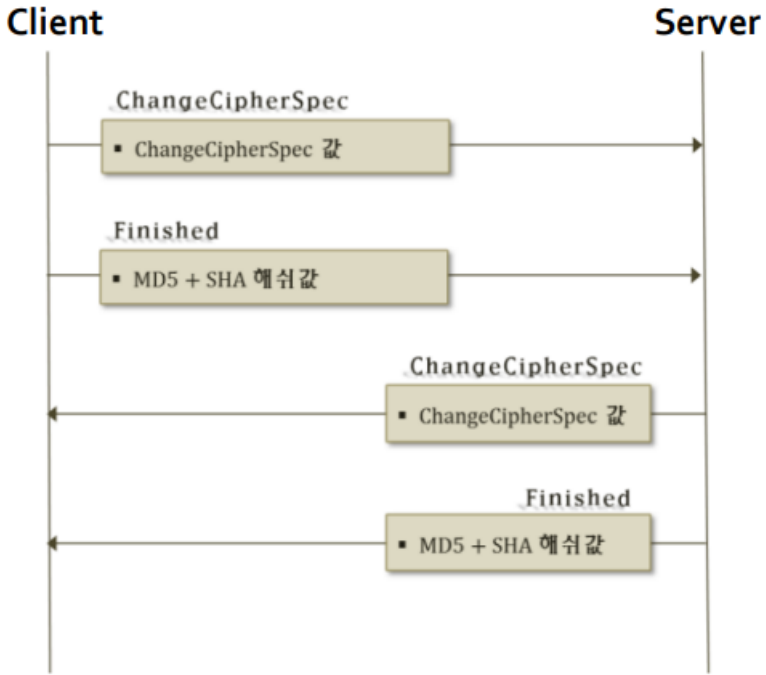
□ master-secret를 통해 session 암호에 필요한 session key (대칭키)를 생성함

- TLS 과정에서 클라이언트는 자신의 pre-master secret 값을 서버의 공개키로 암호화하여 서버로 전송
- 서버는 자신의 개인키로 암호화된 pre-master secret 값을 복호화 하고,
  - 이를 통해 master secret 및 session key (대칭키)를 생성함 - 모두가 동일한 session key를 가짐

TLS - Handshake 프로토콜

■ **Phase 4: 단계 4 에서 클라이언트와 서버는 암호 알고리즘과**

□ 보안 파라메터 값을 변경하는 메시지를 주고 받음



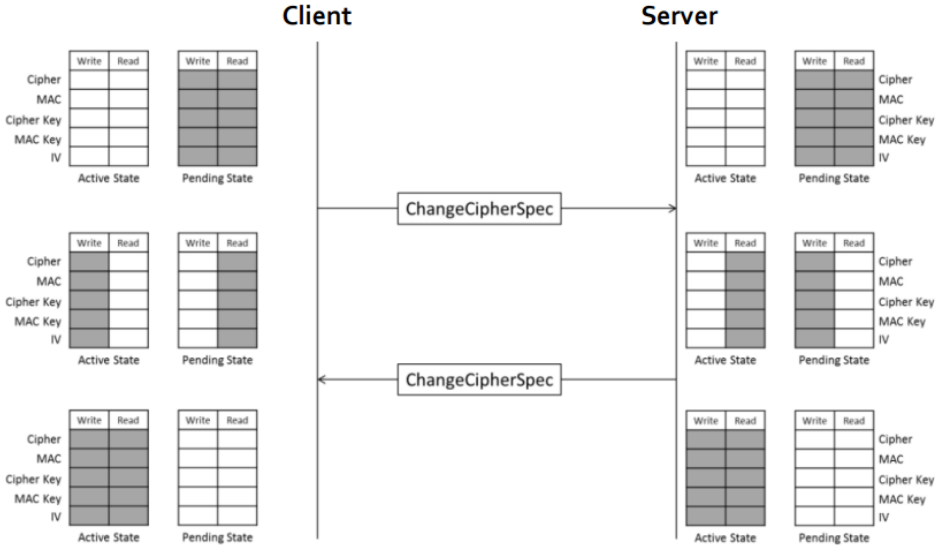
TLS: ChangeCipherSpec 프토토콜

■ **ChangeCipherSpec 프로토콜 과정에는 교환된 CipherSpec 에 대하여 확인하는 과정을 거침**

□ 클라이언트와 서버간

- 암호화/인증/압축 알고리즘 확인

□ 암호화/인증 키 확인



TLS: Alert 프로토콜

■ **시스템이 비정상적으로 동작하였을 때 에러처리를 하기 위하여**

□ 에러/경고 메시지를 Alert 프로토 콜에서 정의하여 사용함

메시지 이름	설명
CloseNotify	더 이상 메시지를 보내지 않을 것이라고 명시하는 메시지이다. 오류 레벨은 경고에 해당된다.
UnexpectedMessage	적합하지 않은 메시지를 수신 받았다고 명시하는 것이며, 오류 레벨은 실패에 해당된다.
BadRecordMAC	올바르지 않은 MAC 값을 수신 받았다는 메시지이며, 오류 레벨은 실패에 해당된다.
DecompressionFailure	압축된 메시지에서 압축을 풀 수 없다는 것을 의미하며, 오류 레벨은 실패에 해당된다.
HandshakeFailure	Handshake 프로토콜 과정이 실패했다는 것을 의미하며, 오류 레벨은 실패에 해당된다.

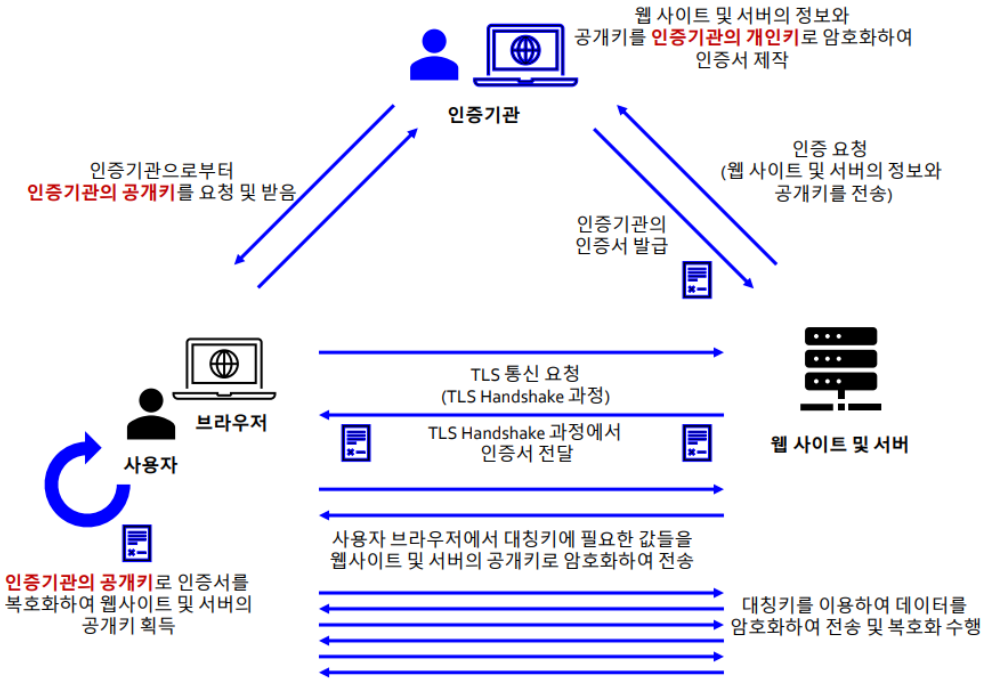
TLS: Record 프로토콜

■ **Record 계층 프로토콜을 사용하여 모든 메시지를 캡슐화(encapsulation) 함**

□ 즉, 상위계층인 응용계층, ChageCipherSpec, Alert, Handshake 프로토콜 메시지를

- **공통된 형식으로 구조화 시키는 역할을 함**

■ **TLS 통신 흐름**

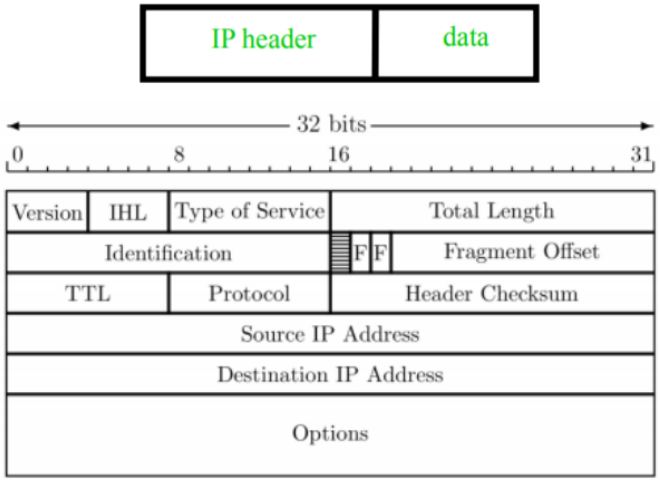


IPSEC

IPSec

- **IPSec (Internet Protocol Security)은 SSL과는 달리 네트워크 계층에서 보안을 제공하기 위해서 IETF (Internet Engineering Task Force)에 의해 표준화 된 프로토콜**
  - IPSec is to add cryptographic protection to IP Header

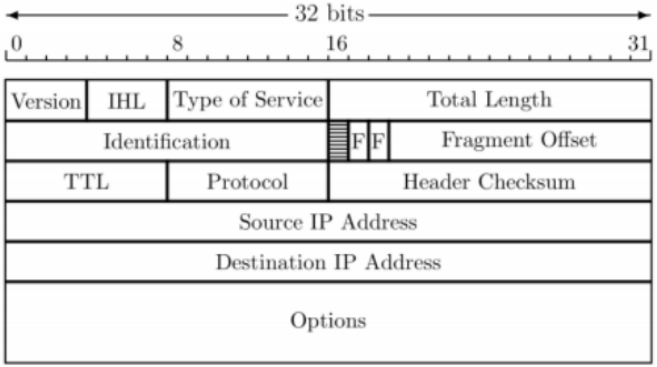
- **Why IPSec?**
  - client/server 애플리케이션들 중 애플리케이션 레이어 에서 보호되지 않는 것들이 존재함
  - client/server 애플리케이션들 중 UDP를 사용하는 것들 이 존재함



IPSec

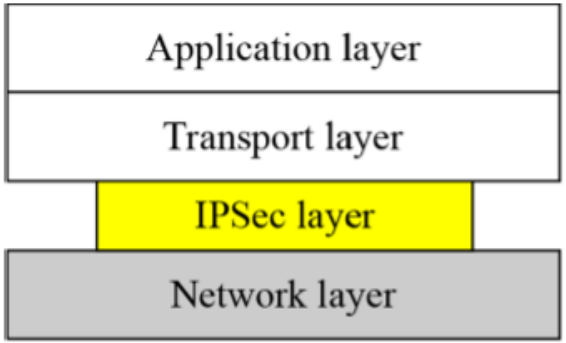
- **Version: IPv4 혹은 IPv6를 나타냄**
  - Version = 4 for IPv4
- **Header length: 헤더의 길이를 나타냄 (4byte 단위)**
- **Total length: 헤더와 데이터의 총 바이트 수**
  - Maximum size is 65,535 bytes.
- **Identification, flags, fragment offset: 패킷 분할 및 재조합에 사용됨**
- **Time to live (TTL): Hop의 카운트를 나타냄.**

Protocol
- ICMP = 1
- TCP = 6
- UDP = 17
- Encapsulating Security Payload (ESP) = 50
- Authentication Header (AH) = 51

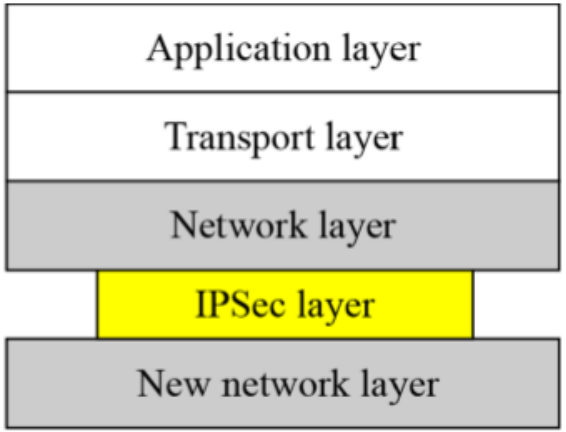


IPSec

■ **Transport mode vs. tunnel mode**



Transport Mode

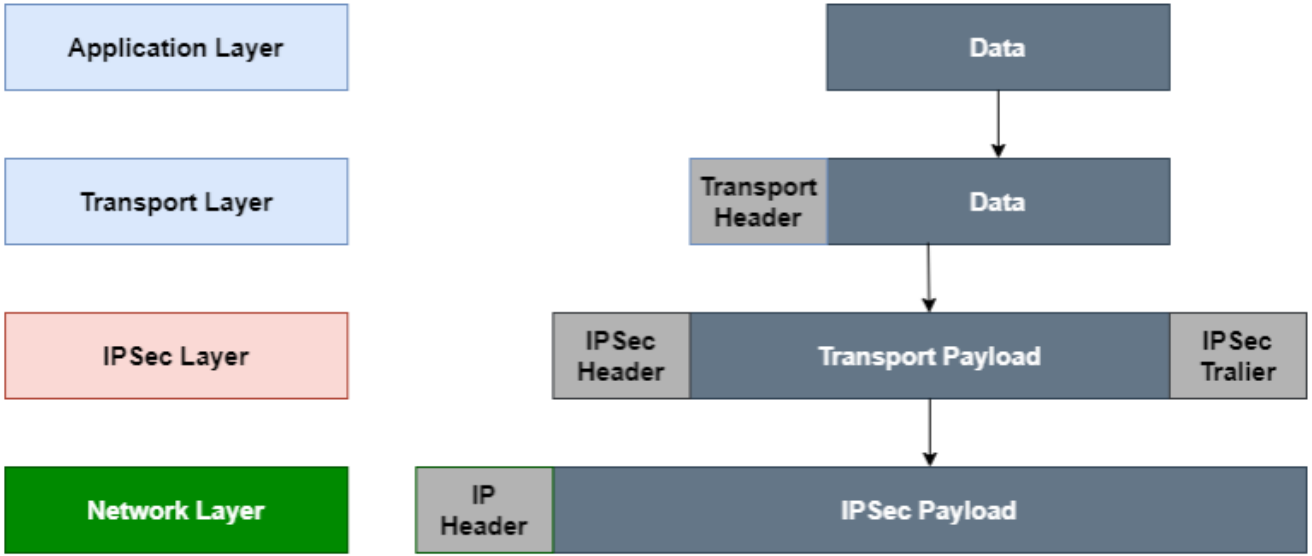


Tunnel Mode

IPSec

■ 전송모드 (Transport 모드)

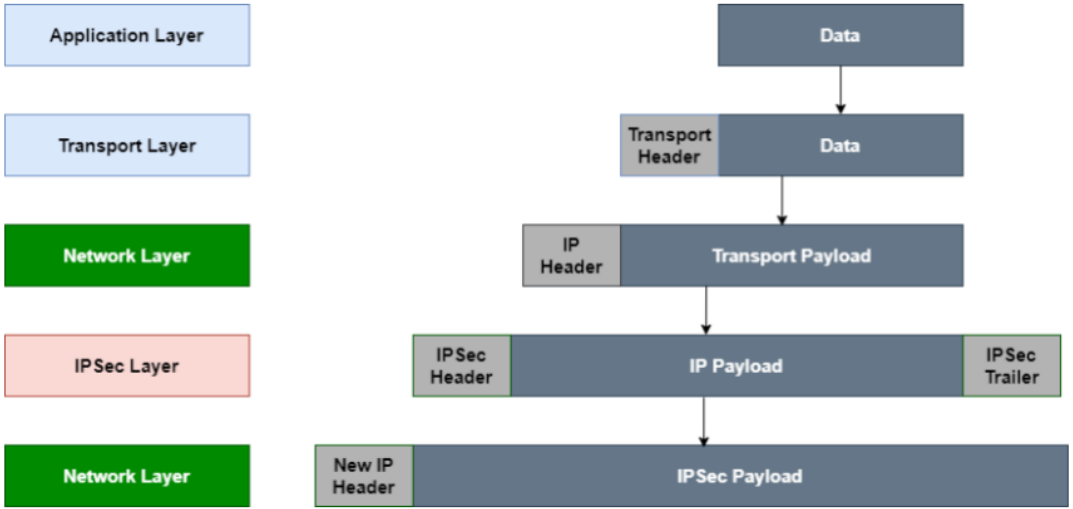
- □ 전송모드는 IPSec의 기본 모드로 사용하며, 종단간 (end-to-end) 보안을 제공하는 방식
- □ 전송 모드가 사용되면 IP 데이터 영역만 보안 처리함으로써, IP 데이터 영역 을 보호하는 방식임
  - 즉, 전송 모드에서는 IP 데이터 영역만이 보호되며, IP 헤더는 보호되지 않음



■ 터널모드 (Tunnel 모드)

□ 터널 모드는 IP 패킷 전체를 보호하는 방식임

- 즉, 전송 모드는 IP 데이터 영역만 보호하는 방식이었지만
  - 터널 모드는 IP 데이터 영역뿐만 아니라 IP 헤더 영역도 보호하는 방식임.
- 터널 모드에서는 원래의 IP 헤더 영역을 암호화 하기 때 문에 IP망 전송을 위한 새로운 IP 헤더를 더함
  → 트래픽 흐름 보호가 가능함



[Source: https://reakwon.tistory.com/108]

IPSec : 보안 프로토콜 (AH and ESP)

■ IPSec에서는 두 개의 보안 프로토콜을 정의함

□ AH (Authentication Header) 프로토콜

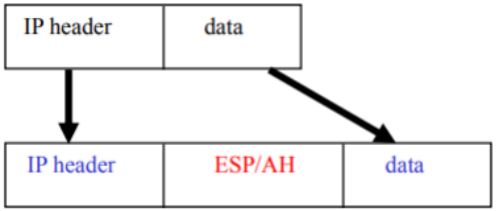
- AH 프로토콜은 IP 패킷에 대한 객체 인증과 IP 데이터 영역에 대한 무결성 을 제공하기 위한 프로토콜임.
  - 기밀성은 제공하지 않음.

□ ESP (Encapsulating Security Payload) 프로토콜

- IPSec 에서는 대체 프로토콜인 ESP 프로토콜을 정의하였으며,
  - ESP 프로토콜에서 는 객체 인증, 데이터 무결성 뿐만 아니라 데이터 기밀성도 제공함.

IPSec : IPSec의 전송모드 운용

■ IPSec 전송모드의 패킷 구조

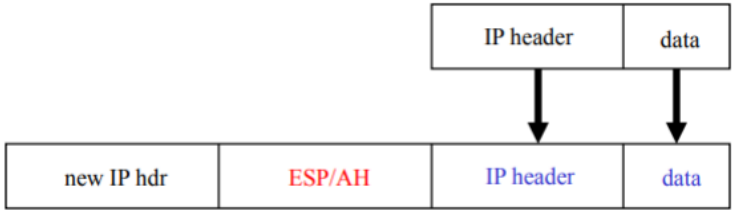


□ 효율적으로 처리가 가능함.

- 프로토콜에 대한 헤더정보 (ESP or AH) 만 추가하면 됨
- Original Header 가 남아있음
- 도청 공격자 (Passive attackers)는 트래픽 흐름을 분석할 수 있음

IPSec : IPSec의 터널모드 운용

■ IPSec 터널모드의 패킷 구조



- □ Original IP 패킷인 IPSec에 포함됨
- □ ESP 프로토콜을 사용한다면, 도청공격자는 Original IP header를 볼 수 없음
  - 도청 공격자는 트래픽 흐름 분석을 할 수 없음

IPSec : AH 프로토콜

■ AH 프로토콜

□ Next header : AH 프로토콜 다음에 오는 페이로드 (payload) 영역의 프로토콜의 형식을 번호로 표시함

□ Payload length : AH 프로토콜 자체 길이

□ Security Parameters Index (SPI)

- 암호 알고리즘, 인증 알고리즘 등을 위한 파라미터

□ Authentication Data or Integrity CheckValue (ICV)

- IP 패킷에 대한 무결성 검증을 위한 값을 저장함 - TTL과 같이 변하는 값들은 무결성 알고리즘의 입력값이 아님.

□ AH 프로토콜의 특징

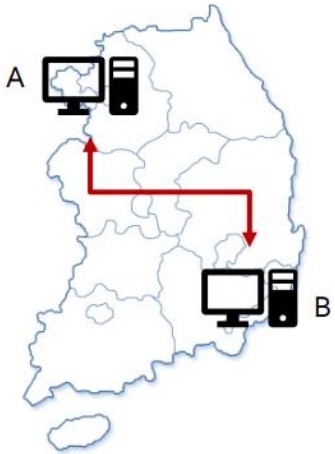
- 전송 모드
  - AH 프로토콜은 IP 패킷에서 변하지 않는 부분들에 대한 무결성을 제공함.
  - AH 프로토콜은 다음부분들에 대한 무결성을 제공하지 못함.
    - IP Header: flags, frag offset,TTL, header checksum
    - AH Header: Authentication Data
  - IP Header의 일부를 변경함
- 터널 모드
  - AH 프로토콜은 IP 패킷의 모든 부분에 무결성을 제공함
  - 새로운 IP 헤더가 만들어짐

VPN

VPN

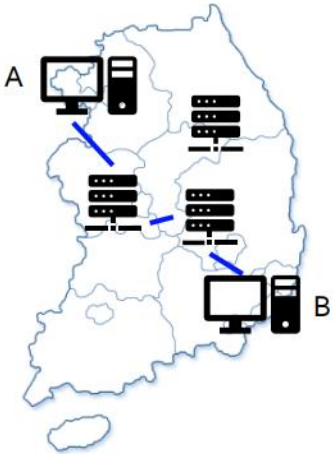
■ VPN (Virtual Private Network): 가상사설망

□ A 와 B 의 보안통신을 위해 전용선을 구축한 경우



□ 기존망을 활용하여,

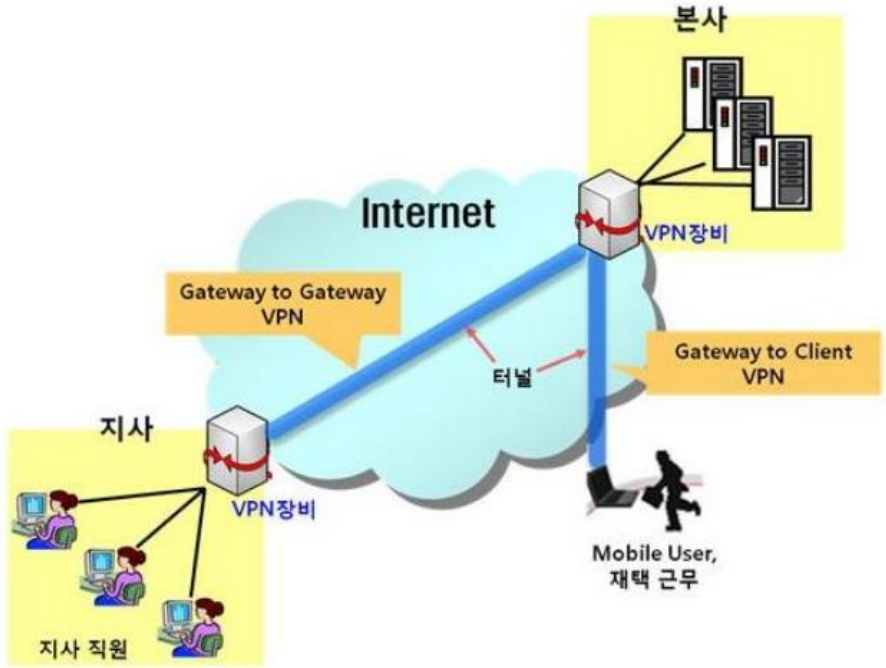
- A와 B의 보안통신을 구축하려면 가상 사설망이 필요



VPN

■ **VPN 구현방식은 Gateway to Gateway (G to G VPN) 와 Gateway to Client(G to C VPN) 방식**

□ 두 가지로 분류할 수 있음



■ VPN: G to G VPN

□ G to G VPN 방식은 본사 및 각 지사 별로 VPN장비를 구축하여 VPN 장비 간에 터널을 제공하는 방식임
- IPSec의 ESP (with 터널모드)를 통해 구현됨

□ VPN 서비스를 이용하는 지사 직원 및 본사 서버에는 VPN과 관련된 어떠한 애플리케이션이나 설정은 필요 없음
- 즉, 본사 및 지사의 VPN장비 간 인터넷 망에는 암호화된 패킷이 전송되고,
  - VPN장비를 거치면서 복호화된 패킷이 본사 서버 및 지사직원에게 전달됨
- 인터넷 망에는 자동으로 눈에 띄지 않게 암호화된 통신이 전송되는 것이 G to GVPN의 개념

■ VPN : G to CVPN

□ G to C VPN 방식은 VPN 장비와 VPN 사용자의 PC간에 터널을 제공하는 방식
- 모듈이 설치되어 있는 계층에 따라 구분되며,
  - 네트워크 계층에 서 VPN기능을 제공하는 IPSec 터널모드(IPSec VPN)과
    - 전송 계층에서 VPN 기능을 제공하는 SSL(SSLVPN)으로 나뉘어짐.

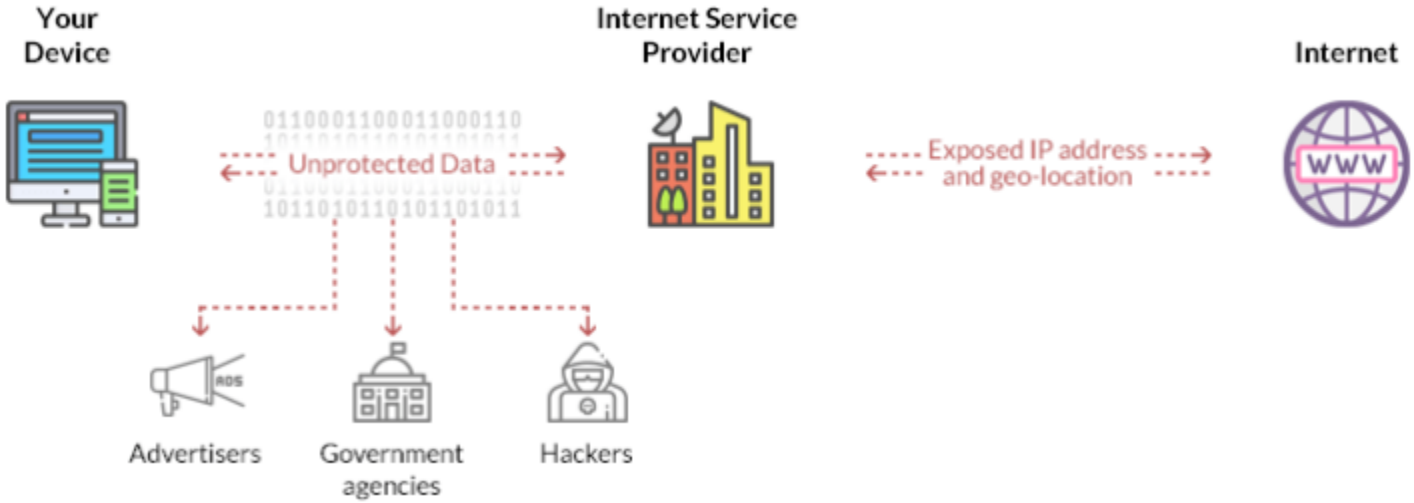
□ IPSec VPN의 경우 별도로 설치되어야 하는 네트워크 드라이버가 반드시 필요하며,
- IPSec관련 설정을 개별로 수행해야 하는 번거로움이 있음
- 또한, 실 환경상에서는 IPSec을 이용하기 위해 설치된 네트워크 드라이버가 PC에 기 설치된 타 드라이버 및 프로그램 간의 충돌로 인한 블루스크린 등의 PC이슈가 발생하는 단점이 존재

VPN : Privacy 관점

■ WithoutVirtual Private Network

□ 클라이언트가 TLS를 사용하더라도, IP에 대한 보호를 제공하지 못하므로

- ISP 및 제 3자가 Traffic 플로우 분석을 할 수 있음.
  - 즉, ISP 혹은 네트워크를 도청가능한 공격자는 사용자가 어디에 접속 했었는지 알 수 있음



Q & A

THANK YOU