

# HTTPS와 ssl 인증서

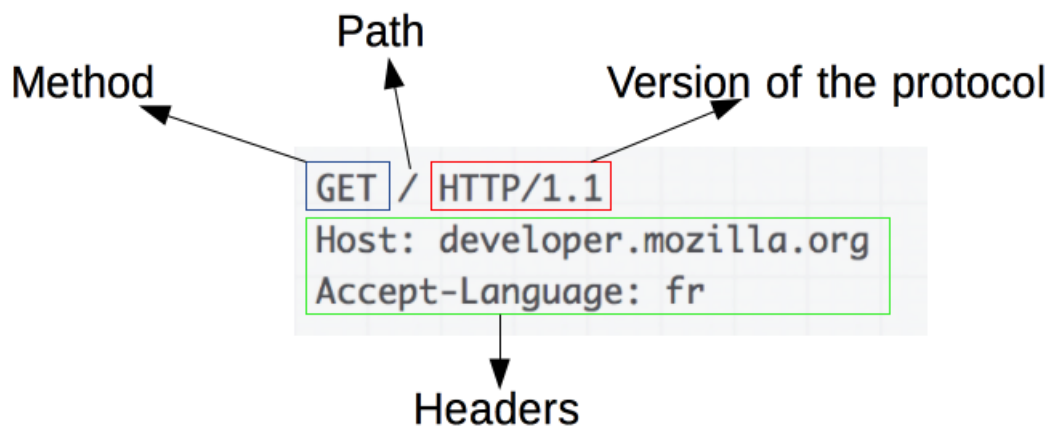
## HTTP ( 포트번호 80 )

: 서버/클라이언트 모델을 따라 데이터를 주고받기위한 프로토콜

### 1. 특징

: 무상태성(=비연결성=stateless)의 성질

### 2. 구조



## HTTPS ( 포트번호 433 )

: HTTP + Security (SSL or TLS)

: HTTP에 데이터 암호화가 추가된 프로토콜

### 1. 특징

1) 기밀성 : 메시지를 가로챌 수 없음 = 읽을 수 없음

2) 무결성 : 메시지가 조작되지 않음 = 수정할 수 없음

2. 확인 : 브라우저 URL 창에 있는 자물쇠 아이콘을 클릭하면 확인할 수 있다

## SSL과 TLS

: 인증서

1. SSL(Secure Socket Layer) : 브라우저와 서버 사이의 암호화된 연결을 수립하는데 쓰는 인증서

2. TLS(Transport Layer Security) : SSL의 향상된 더 안전한 버전

## 암호화

1. 공개키 : 모두에게 공개가능한 키
2. 비밀키 : 나만 가지고 알고 있어야 하는 키
3. 대칭키 (RSA) 암호화 : 키 한개로 암호/복호
4. 비대칭키 (AES) 암호화 : 암호/복호화 키 다름
  - 1) 개인키로 암호 + 공개키로 복호 = 전자서명 (출처만 확인)
  - 2) 공개키로 암호 + 개인키로 복호 = 암호화(나만 볼 수 있음) <= 실질적인 암호화
5. 해시 : 시드 + 본문의 해시값이 같은지 확인
6. HTTPS는 대칭키 / 비대칭키 혼용함

## SSL 인증서 작동 원리

SSL [ 대칭키  
비대칭키 ]

### 대칭키 알고리즘



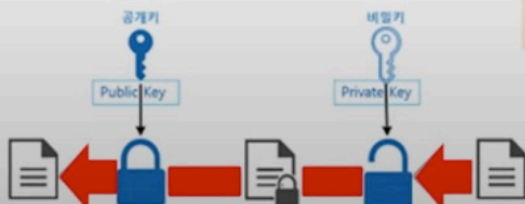
- 발신자와 수신자가 동일한 비밀키 사용
- 메시지 전달 경로가 길어지면 위험
- 비밀키가 유출되면 엄청난 재앙
- 하지만, 속도가 빠름

### 비대칭(공개)키 알고리즘



- 공개키: 누구나 사용할 수 있는 비밀키
  - 비공개(비밀)키: 절대 알려주지 않는 비밀키
- "공개키 다운 받아서 그걸로 암호화 해서 보내셈~"
- 메시지 내용은 암호화, 비밀키를 가진 나만이 볼 수 있다!
  - 비밀키 유출 우려 해결
- 비대칭 암호 통신의 원리

### 만약, 방향이 반대라면??



- "내 비공개키로 암호화 해서 보내니 공개키로 열어 보셈~"
- 어라! 공개키로 열리네!!
  - 비공개키 가진 사람이 암호화 해서 보낸 것이 확실하다!
- 인증서의 원리

↳ RSA 알고리즘 (공개키 암호와 암호해독)

# SSL 동작 → 대칭키 + 비대칭키 혼합 방식

## ■ SSL 기본 원리

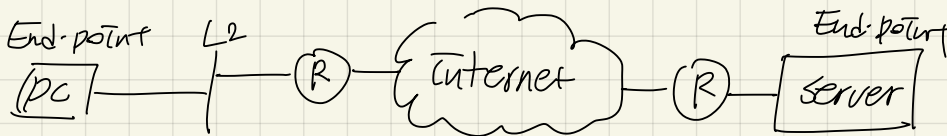
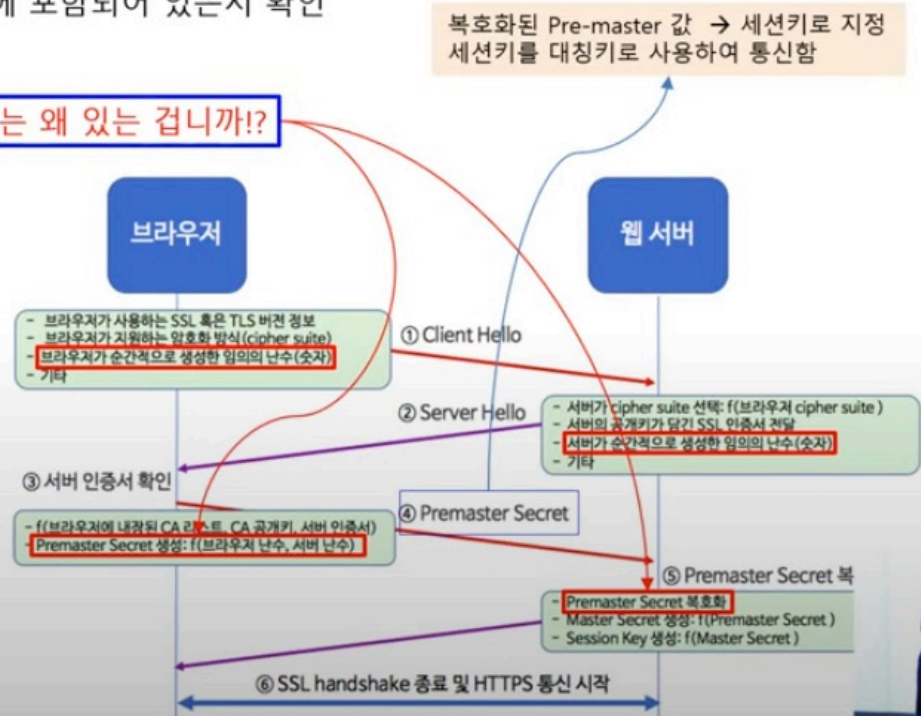
- 브라우저가 서버에 접속 → 서버 자신의 인증서를 제공
- 서버가 제공한 인증서가 CA 리스트에 포함되어 있는지 확인

### - 의문점

- 인증서에 담겨있는 서버의 공개키는 왜 있는 겁니까!?

## ■ 실제 SSL 작동

- 암호화 방식
  - 실제 데이터 → 대칭키 암호화
  - 대칭키의 키 → 서버의 공개키
- 3단계로 구성
  1. 악수(handshake)
  2. 데이터 전송
  3. 세션 종료



① 키 생성

PC Pub + PC Pri Key

Server Pub + Server Pri Key

Pub → 많지만  
Pri → 복제본만

② 키 교환

PC

Server

Server Pub  
File → Key

PC Pub + C

Key → File

1. 공개키 : 모두에게 공개가능한 키
2. 비밀키 : 나만 가지고 알고 있어야 하는 키
3. 대칭키 (RSA) 암호화 : 키 한개로 암호/복호
4. 비대칭키 (AES) 암호화 : 암호/복호화 키 다름
  - 1) 개인키로 암호 + 공개키로 복호 = 전자서명 (출처만 확인)
  - 2) 공개키로 암호 + 개인키로 복호 = 암호화(나만 볼 수 있음) <= 실질적인 암호화
5. 해시 : 시드 + 본문의 해시값이 같은지 확인
6. HTTPS는 대칭키 / 비대칭키 혼용함