HTTPS와 ssl 인증서

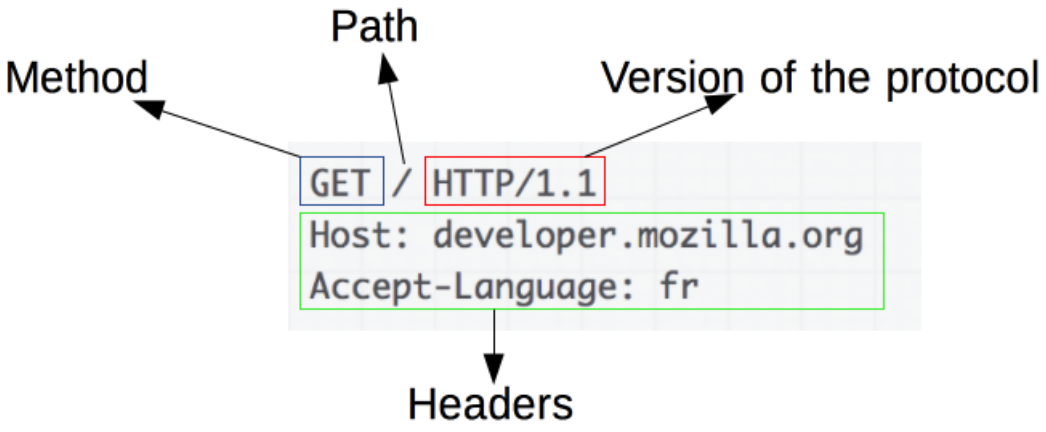
HTTP (포트번호 80)

: 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 : 메세지를 가로챌 수 없음 = 읽을 수 없음

2) 무결성 : 메세지가 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 암/복호

4. 비대칭키 (AES) 암호화 : 암/복호화 키 다름

1) 개인키로 암호 + 공개키로 복호 = 전자서명 (출처만 확인)

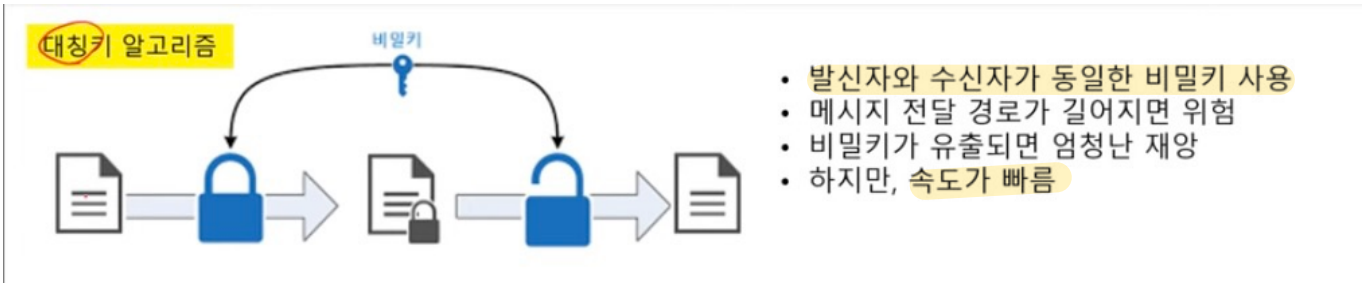
2) 공개키로 암호 + 개인키로 복호 = 암호화(나만 볼 수있음) <= 실질적인 암호화

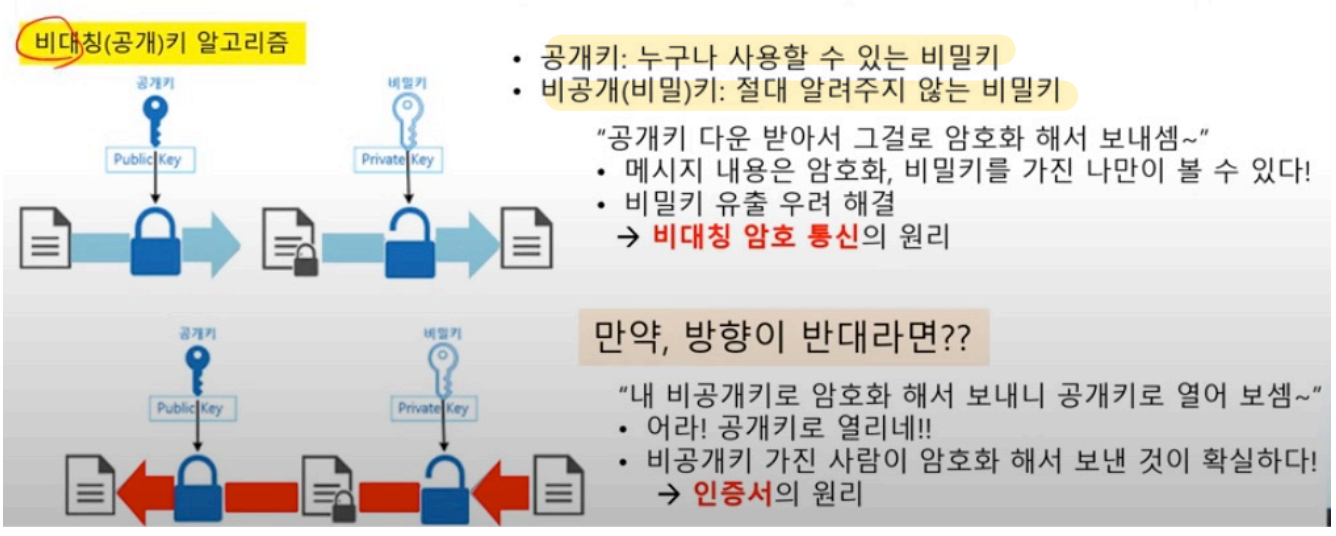
5. 해시 : 시드 + 본문의 해시값이 같은지 확인

6. HTTPS는 대칭키 / 비대칭키 혼용함

SSL 인증서 작동원리

· SSL ┌ 대칭키
      └ 비대칭키





└ RSA 알고리즘 (공개키 암호화 알고리즘)

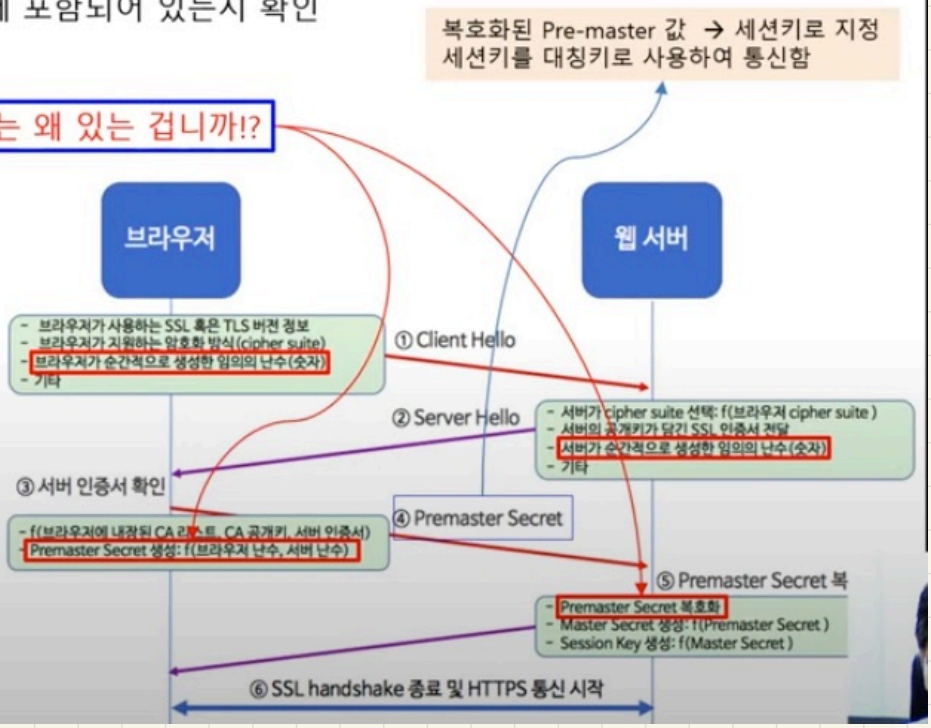
SSL 동작 → 대칭키 + 비대칭키 혼합 방식

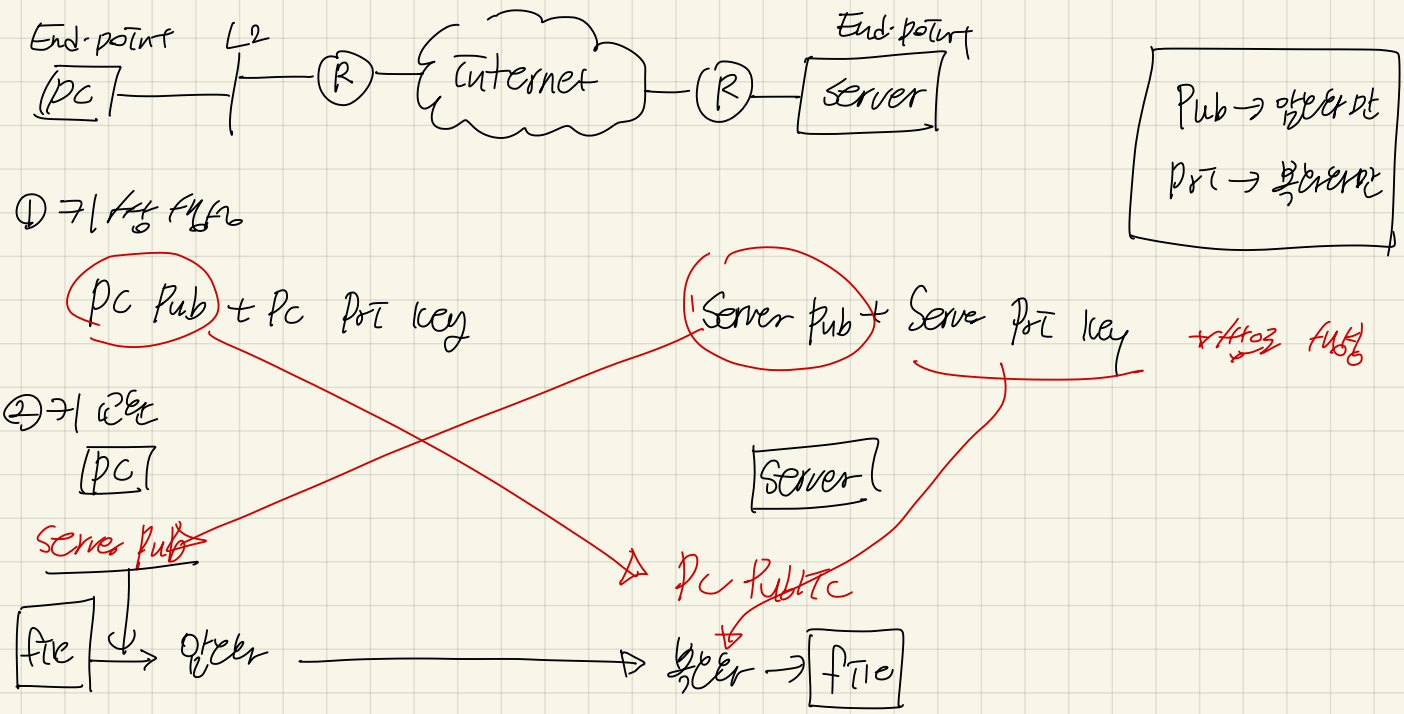
SSL 기본 원리

- 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. 공개키 : 모두에게 공개가능한 키

2. 비밀키 : 나만 가지고 알고 있어야 하는 키

3. 대칭키 (RSA) 암호화 : 키 한개로 암/복호

4. 비대칭키 (AES) 암호화 : 암/복호화 키 다름

1) 개인키로 암호 + 공개키로 복호 = 전자서명 (출처만 확인)

2) 공개키로 암호 + 개인키로 복호 = 암호화(나만 볼 수있음) <= 실질적인 암호화

5. 해시 : 시드 + 본문의 해시값이 같은지 확인

6. HTTPS는 대칭키 / 비대칭키 혼용함