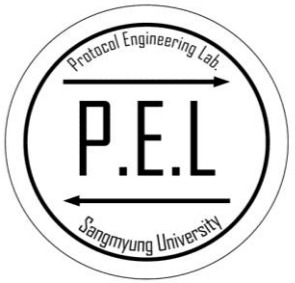


NETWORK SECURITY ESSENTIALS

- SSL/TLS -

Boo-Hyung Lee
(boohyung@pel.smuc.ac.kr)
Protocol **E**ngineering **L**ab., **Sangmyung** University

2015. 01. 30

Content

- 웹 보안
- SSL(Secure Socket Layer)
- SSL/TLS

웹 보안

- **웹(Web)이란?**

  - 인터넷과 TCP/IP 인트라넷상에서 운영되는 기본적인 클라이언트/서버 응용프로그램
  - 다양한 형태의 데이터와 정보에 접근할 수 있도록 해주는 인터넷 서비스
  - WWW(World Wide Web), W3이라고도 불림

- **웹 보안의 필요성**

  - 웹의 특성과 일치

    1) 웹 기반의 정보서비스나 전자상거래 활성화로 웹의 중요성이 점차 증가

    2) 보다 나은 서비스, 다양한 서비스를 위한 웹 서버의 기능확장으로 보안적인 취약점, HTTP 프로토콜의 Stateless 특성으로 인한 구조적인 보안 취약점 내재

웹 보안의 필요성

- **웹 트래픽 보안 방법**

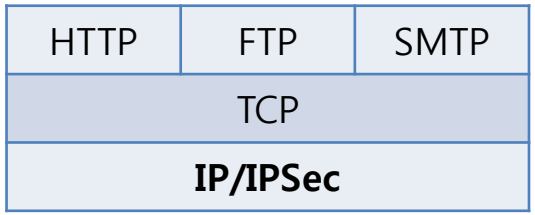
  - 네트워크 계층에서의 보안

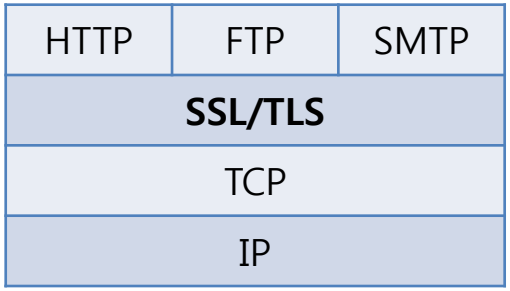
    → 전송계층의 바로 아래에서 보안을 구현 예) IPSec

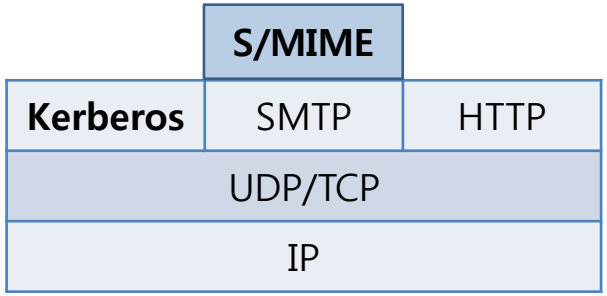
  - 응용 계층과 전송 계층 사이에서의 보안

    → 전송계층의 프로토콜인 TCP 바로 위에서 보안을 구현 예) SSL/TLS

  - 응용프로그램 별로 보안을 제공하는 방법







SSL(Secure Socket Layer)

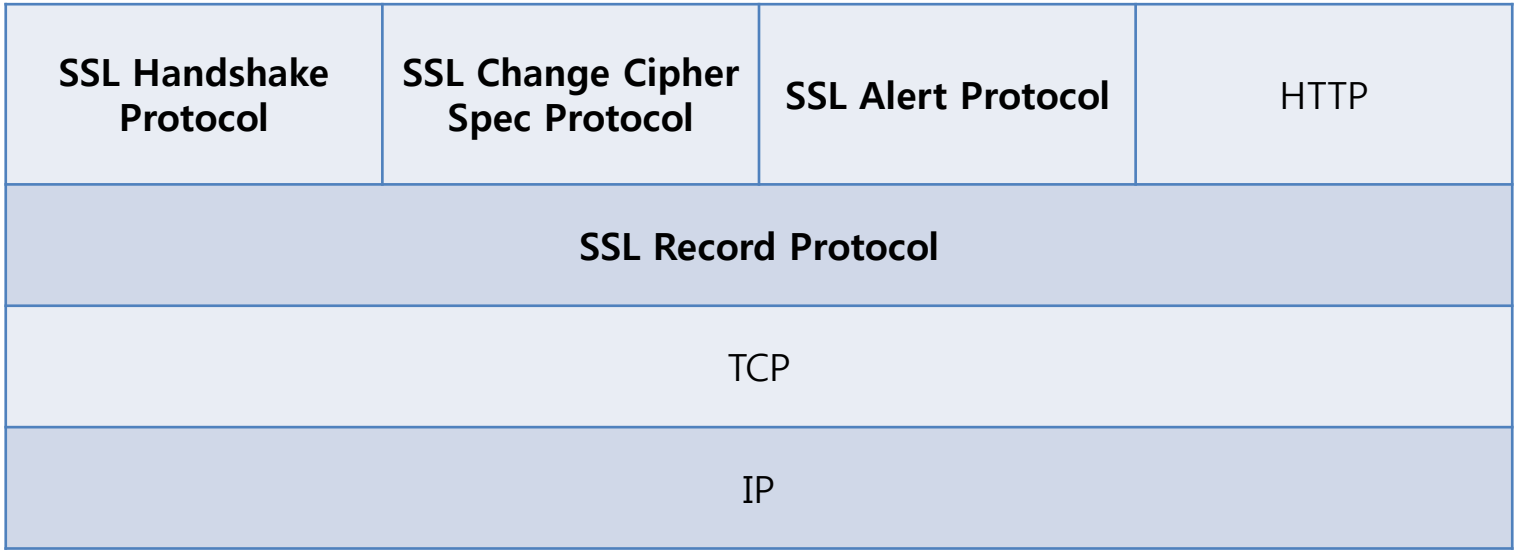
- **SSL의 정의**

  - 넷스케이프사에 의해 1994년에 개발된 웹 보안 프로토콜
  - 1999년에 RFC 2246을 통해 TLS으로 표준화됨
  - 응용계층의 프로토콜 종류에 상관없이 사용할 수 있음
  - 인증과 암호화, 무결성을 보장

SSL(Secure Socket Layer)

- **SSL의 구조**

- SSL은 두 계층의 프로토콜로 이루어짐
- 신뢰성 높은 서비스 제공을 위해 TCP 사용



SSL(Secure Socket Layer)

- **SSL 세션 파라미터**

- SSL Handshake Protocol을 이용하여 세션 키, 암호 알고리즘, 인증서 등의 파라미터를 협상
- 교환된 파라미터를 SSL Record Protocol과정에서 사용
- **파라미터**

1) 세션 식별자 : Server가 선택하는 임의 바이트로 사용 중이거나 재사용 가능한 세션을 인식
2) 인증서 : 통신하는 상호간의 X.509v3 인증서
3) 압축 정보 : 암호화 전 압축에 사용할 알고리즘
4) 암호화 정보 : 암호화와 해시에 사용될 알고리즘과 MD크기 등의 보안 특성
5) Master Secret : Client와 Server간의 일회용 48바이트 비밀 정보
6) 재사용 : 새로운 연결을 시도할 경우, 기존의 세션을 사용할 것인지의 여부

☞ Master Secret : 사전 Master Secret와 서버/클라이언트 난수를 통해 서버와 클라이언트가 각각 계산하여, 대칭 키와 메시지 인증 코드 키, 초기화 벡터를 만드는데 쓰임

☞ 사전 Master Secret : 클라이언트가 만든 난수; Master Secret을 만들기 위한 종자 역할을 하며, 서버의 공개 키로 암호화하여 서버에게 보냄

SSL(Secure Socket Layer)

- **SSL 연결 파라미터**

- 새로운 연결마다 생성, 갱신되는 임시 정보

- 파라미터

1) Server와 Client 난수 : 연결을 위해 Sever와 Client에서 생성하는 난수

2) Server MAC Secret : Server가 전송하는 데이터의 MAC 계산에 사용되는 비밀키

3) Client MAC Secret : Client가 전송하는 데이터의 MAC 계산에 사용되는 비밀키

4) Server 키 : Server가 전송하는 데이터의 암호화와 복호화에 사용되는 관용 비밀키

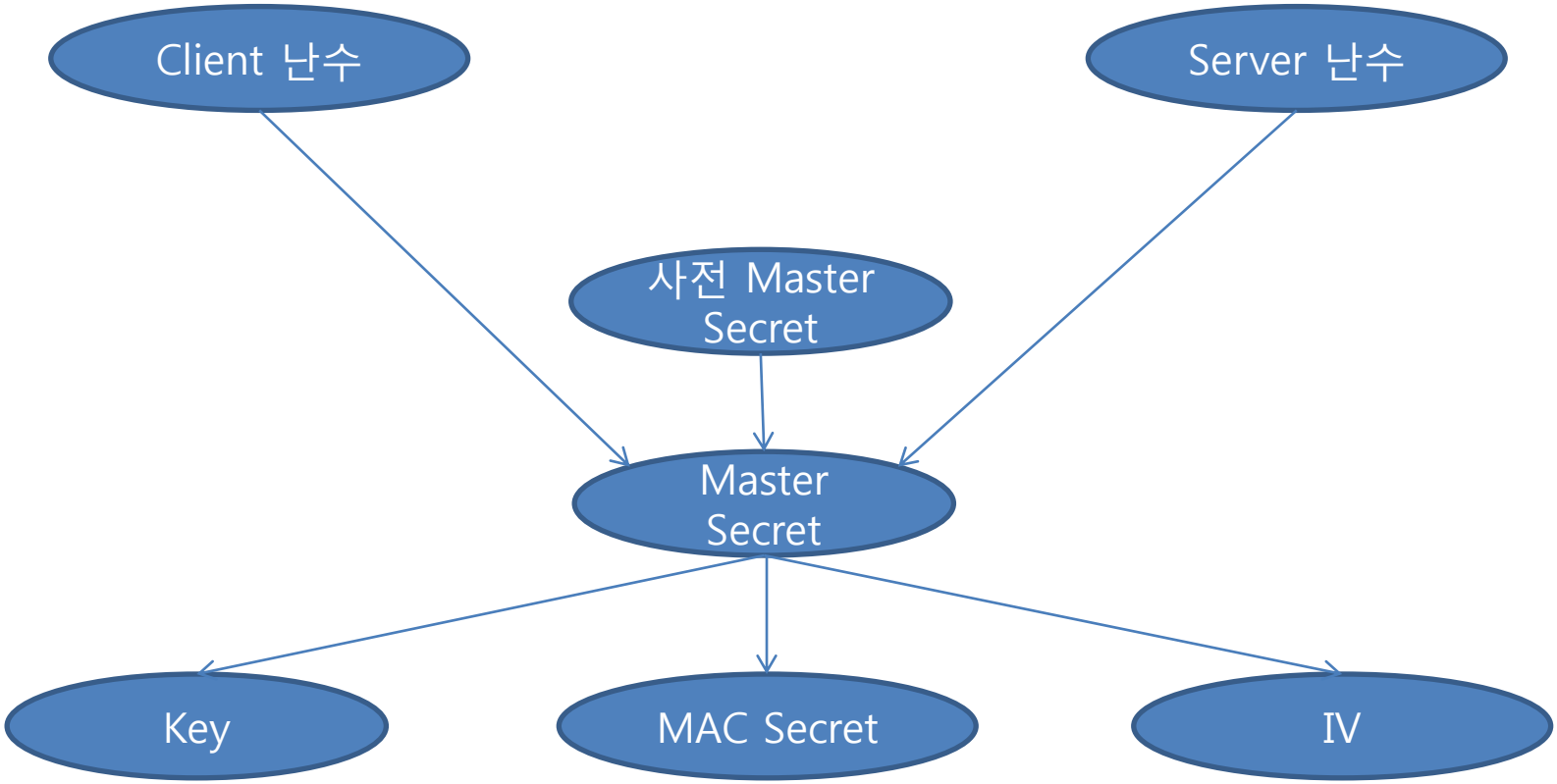
5) Client 키 : Client가 전송하는 데이터의 암호화와 복호화에 사용하는 관용 비밀키

6) Server/Client IV : Server와 Client가 각각 CBC모드의 블록 암호화에 사용되는 초기값

7) 일련번호 : 각 연결의 송신과 수신에 사용되는 일련번호

SSL(Secure Socket Layer)

- **파라미터 생성(1/5)**



SSL(Secure Socket Layer)

- **파라미터 생성(2/5)**

  - Client/Server 난수 : Client/Server가 PRNG를 이용해 만든 난수(32bytes); nonce로 사용
    timestamp(4bytes) + random value(28bytes)
  - 사전 Master Secret : Client가 PRNG를 이용해 만든 난수(48bytes); 일종의 세션키로 사용
    client_version(2bytes) + random value(46bytes)

☞ client_version : 클라이언트가 수용할 수 있는 가장 높은 SSL 버전

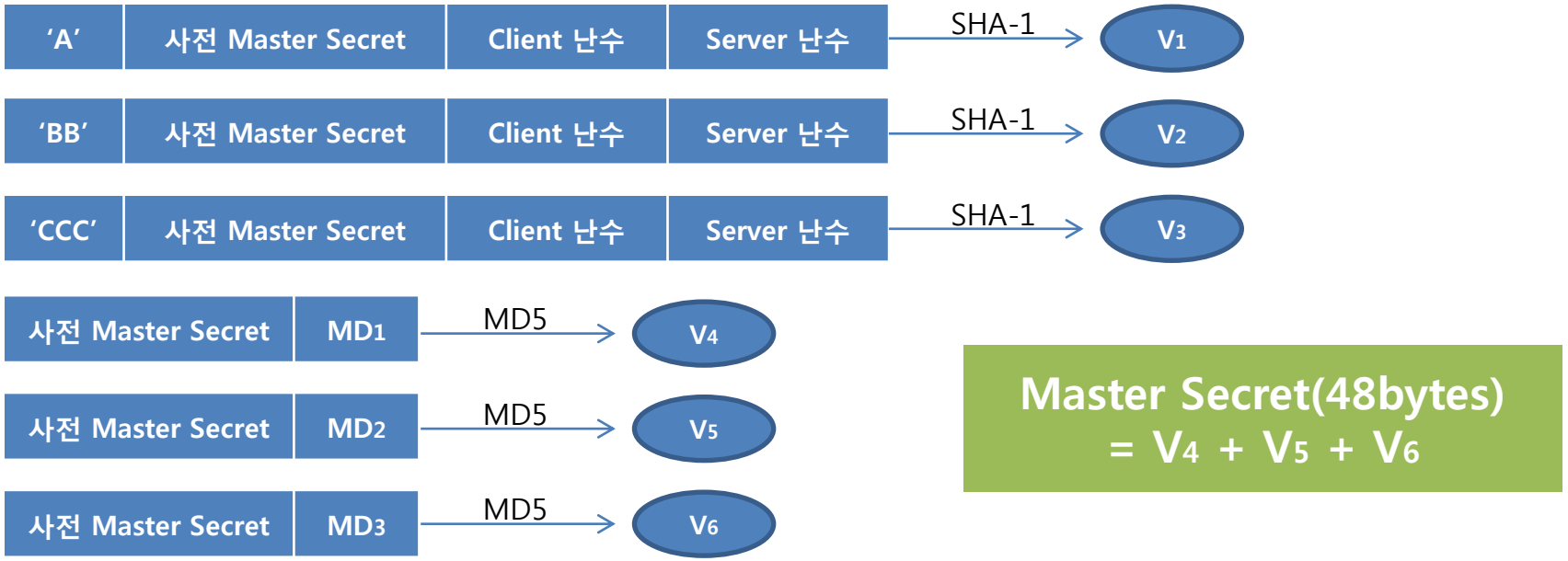
☞ PRNG(Pseudo Random Number Generator) : 의사 난수 생성기

SSL(Secure Socket Layer)

- **파라미터 생성(3/5)**

- Master Secret : 사전 Master Secret과 Client/Server 난수로, Client와 서버가 각각 계산
  Server/Client 키, Server/Client MAC Secret, Server/Client IV를 만드는데 필요한 정보

- 생성과정

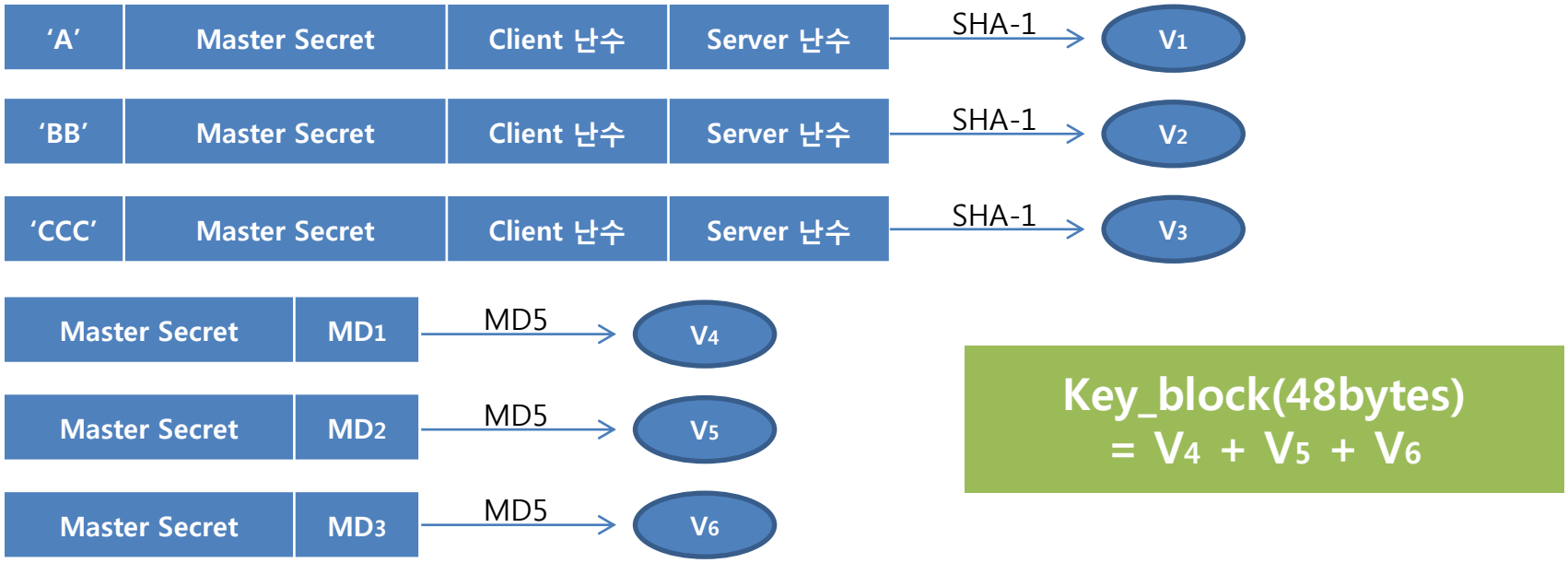


SSL(Secure Socket Layer)

- **파라미터 생성(4/5)**

- Key_block : Master Secret과 Client/Server 난수로, Client와 서버가 각각 계산
  Key_block의 일부로 Server/Client 키와 Server/Client MAC Secret을 만듬

- 생성과정



Key_block(48bytes)
= V4 + V5 + V6

SSL(Secure Socket Layer)

- **파라미터 생성(5/5)**

- Key_block 구조

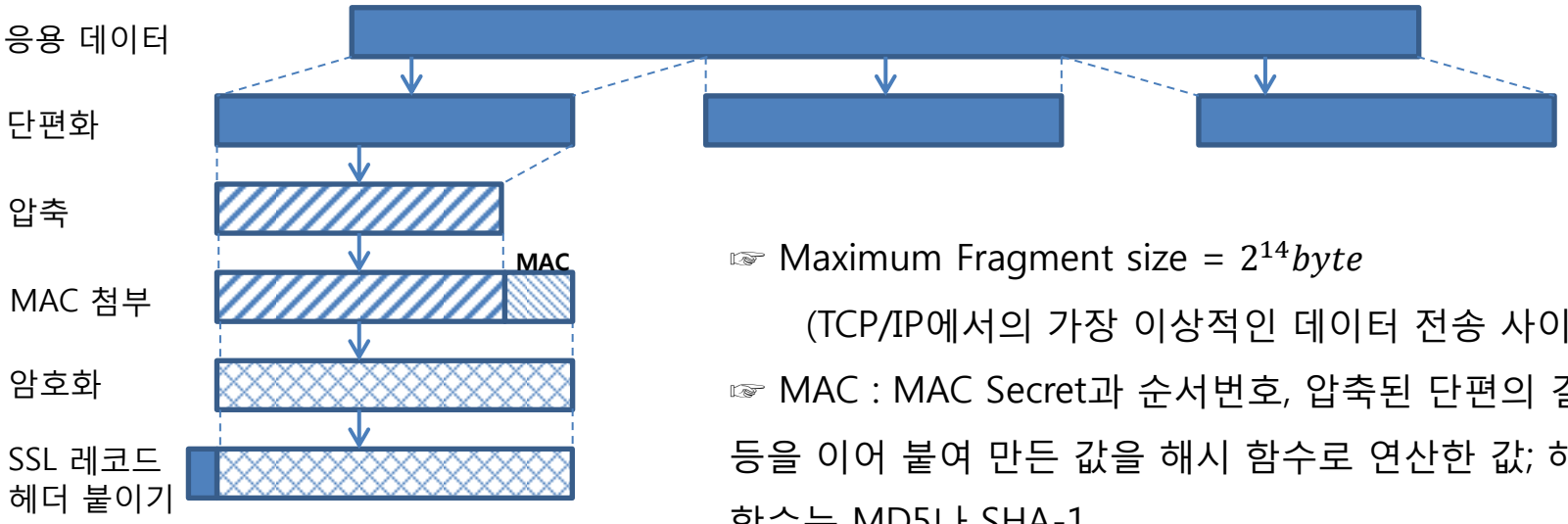
- Client/Server IV

SSL(Secure Socket Layer)

- **SSL Record Protocol**

- Data를 TCP에 안전하게 전달하는 역할
- 기밀성 제공 : 암호화, 메시지 인증 : MAC

- 전반적인 동작과정



☞ Maximum Fragment size = $2^{14} byte$

(TCP/IP에서의 가장 이상적인 데이터 전송 사이즈)

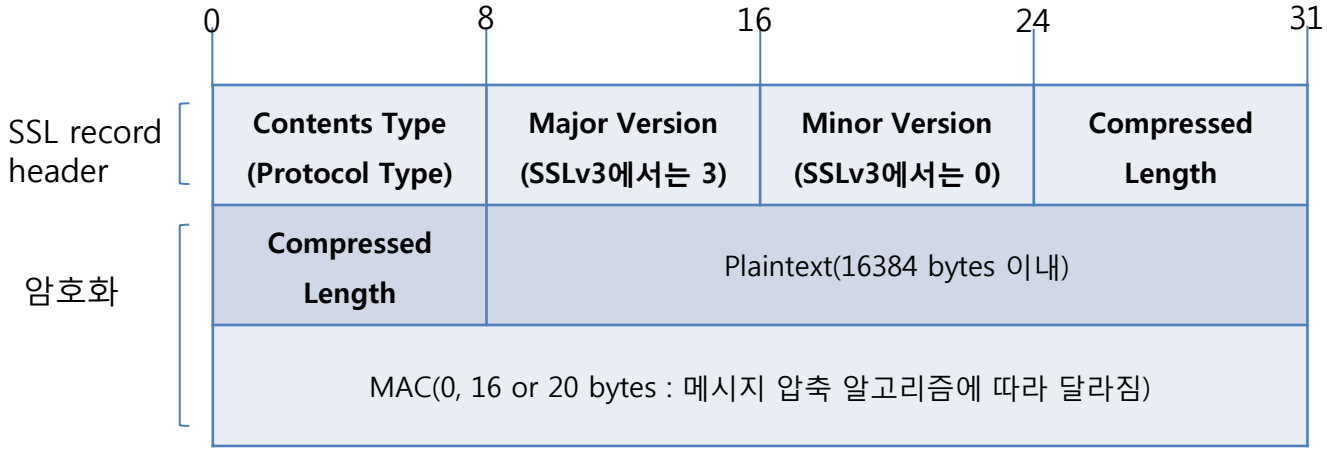
☞ MAC : MAC Secret과 순서번호, 압축된 단편의 길이 등을 이어 붙여 만든 값을 해시 함수로 연산한 값; 해시 함수는 MD5나 SHA-1

SSL(Secure Socket Layer)

- **SSL Record 형식**

- SSL Record Header는 프로토콜 타입, 버전, 압축된 길이로 총 40비트로 이루어져있음
- Protocol Type 필드 값에 따른 값

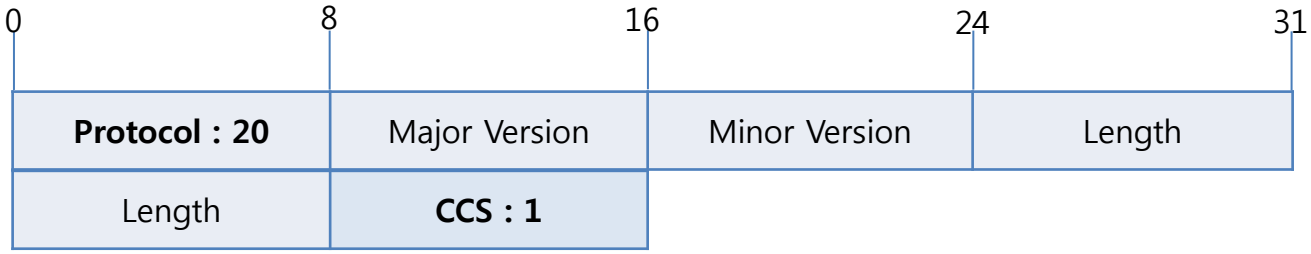
1) Change Cipher Spec Protocol : 20
2) Alert Protocol : 21
3) Handshake Protocol : 22
4) Application Data Protocol : 23



SSL(Secure Socket Layer)

- **SSL Change Cipher Spec Protocol**

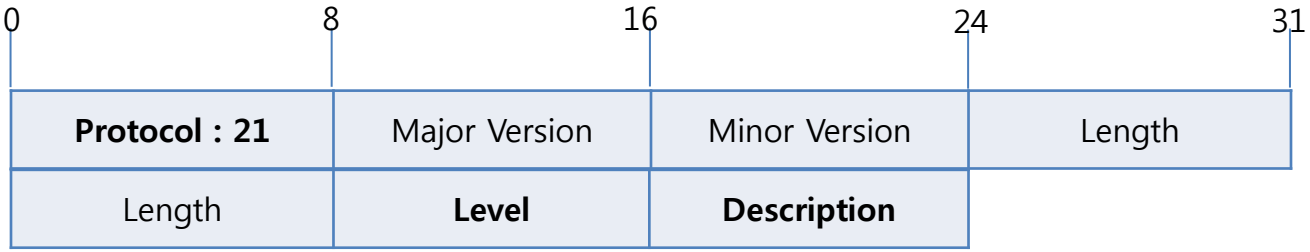
- SSL-지정 프로토콜 중 하나
- Handshake 프로토콜에 의해 협상된 압축, MAC, 암호화에 쓰이는 방식이 이후부터 적용됨을 수신자에게 알리는 역할
- 1 byte이며, 값 1을 갖는 한 개의 메시지로 구성됨



SSL(Secure Socket Layer)

- **SSL Alert Protocol**

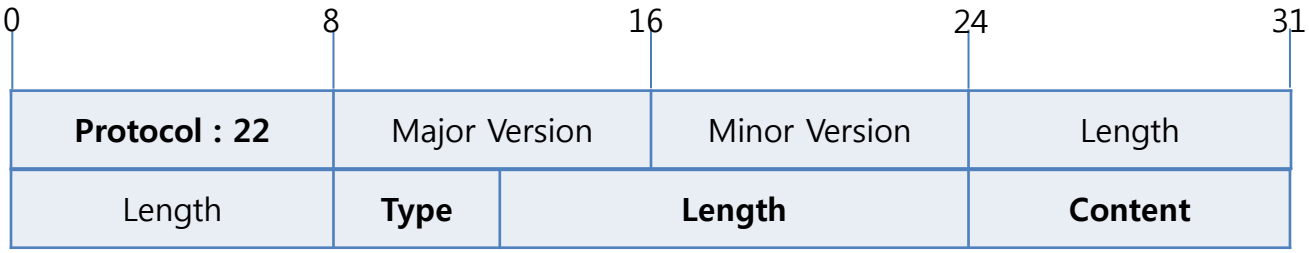
- 암호 오류, 압축 오류, 메시지 인증 오류, 인증 실패 등의 에러 발생을 수신자에게 알리는 역할
- 메시지는 2 byte로 구성
  1) 첫번째 바이트(Level) : warning(1)또는 fatal(2) 값을 가짐
  2) 두번째 바이트(Description) : 세부적인 에러코드
     핸드셰이크, 암호명세 변경, 레코드 프로토콜 수행 중 발생하는 오류메시지를 표현
- SSL과 경고메시지는 압축되고 암호화



SSL(Secure Socket Layer)

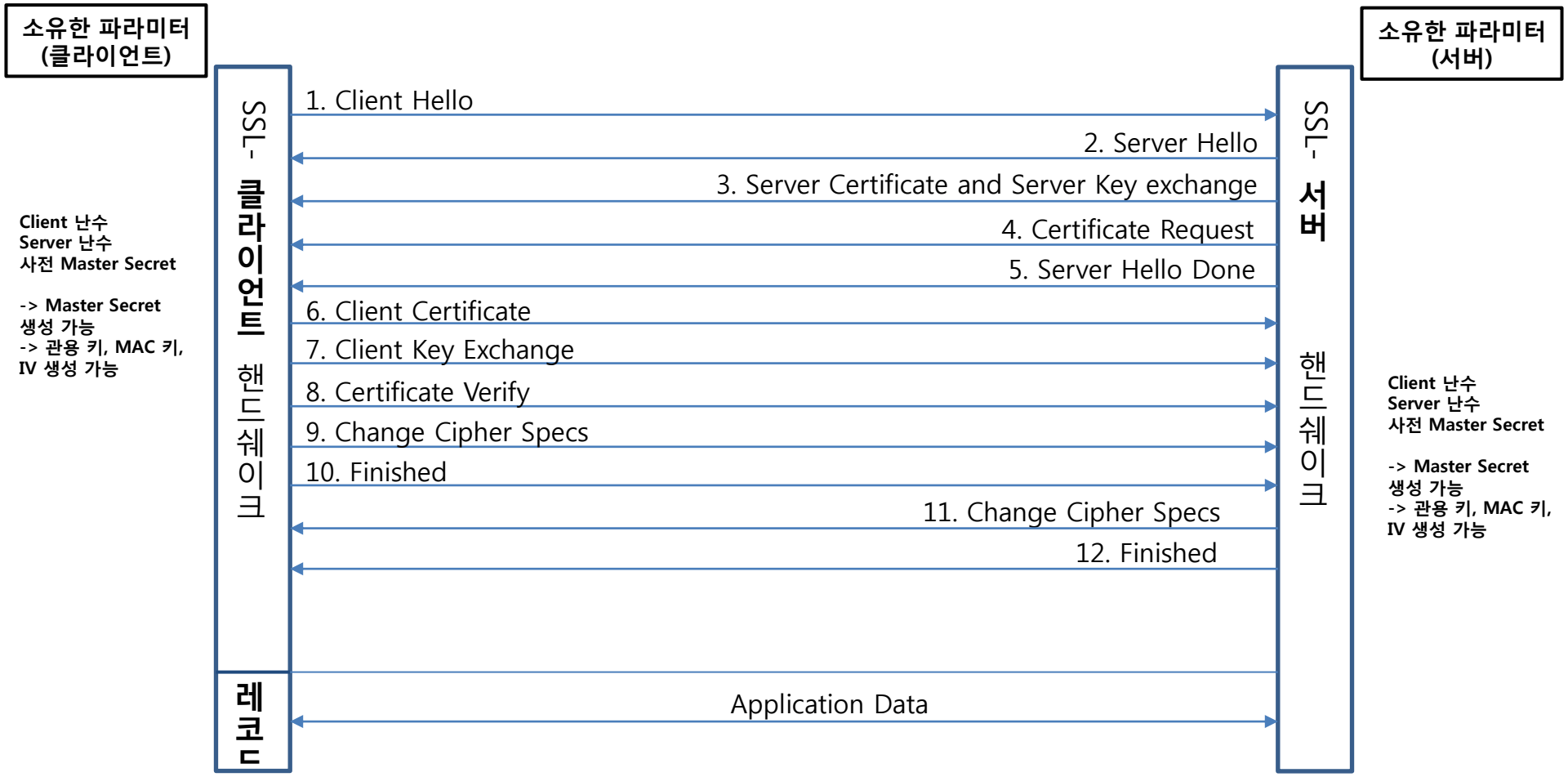
- **SSL Handshake Protocol**

  - 모든 응용 데이터를 전송하기 이전에 사용; 사용할 파라미터의 협상 역할
  - 클라이언트와 서버가 암호 통신에 사용할 공유 키와 MAC 알고리즘을 결정, 인증서를 이용한 인증
  - 클라이언트와 서버 사이의 연속된 여러 메시지 교환으로 구성



SSL(Secure Socket Layer)

- **SSL Handshake Protocol 과정**



SSL(Secure Socket Layer)

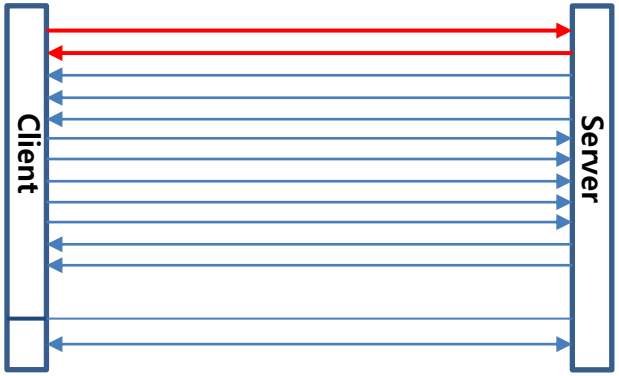
- **SSL Handshake Protocol 과정 설명**

- 단계 1(그림에서 1, 2번 과정) : 일종의 초기화 과정

1) Client Hello : 클라이언트가 수용할 수 있는 SSL 버전, Client 난수, 세션 ID, 지원가능한 Cipher Suite 리스트와 압축방법 리스트 등의 정보를 서버에 전송

2) Server Hello : Client Hello 메시지에 대한 응답으로 Server Hello 메시지를 전송; Server Hello 메시지 안에는 사용할 SSL 버전, Server 난수, 세션 ID, 클라이언트가 보낸 Cipher Suite 리스트 중에서 선택한 하나의 Cipher Suite, 클라이언트가 보낸 압축방법 리스트에서 선택한 압축 방법 등의 정보가 들어있음

☞ 세션 ID : 가변 길이의 세션 식별자



SSL(Secure Socket Layer)

- **SSL Handshake Protocol 과정 설명**

- 단계 2(그림에서 3, 4, 5번 과정)

3) Server Certificate and Server Key Exchange : Server의 인증서와 세션 키(사전 Master Secret)를 암호화할 때 이용될 키 교환용 공개 키를 Client에게 전송

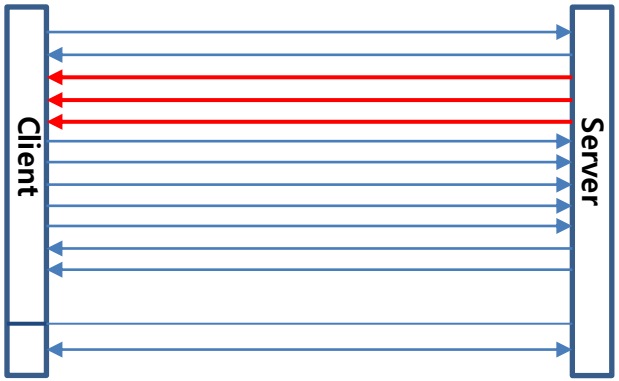
만약, Server의 인증서가 없을 경우 Key Exchange 과정 이후에 서명1을 통해 자신을 인증할 수 있음

→ 이제 Client는 Master Secret 생성 가능 : 관용 키, MAC 키, IV 생성 가능

4) Certificate Request : Client의 인증서를 요청함

5) Server Hello Done : Server의 Hello 절차가 완료되었음을 알림; 이제부터 클라이언트의 응답시작

☞ 서명1 : Client/Server 난수와 키 쌍 생성에 필요한 매개변수에 해시를 취하고, 서버의 개인키로 암호화를 하여 생성



SSL(Secure Socket Layer)

- **SSL Handshake Protocol 과정 설명**

- 단계 3(그림에서 6, 7, 8번 과정)

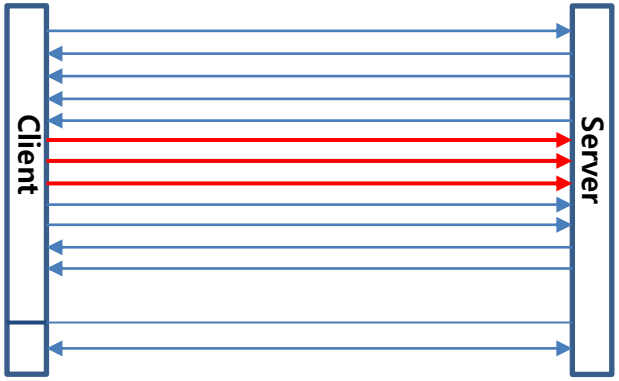
6) Client Certificate : Client의 인증서를 전송

7) Client Key Exchange : 세션 키(사전 Master Secret)를 Server의 공개 키로 암호화하여 전송

→ 이제 Server는 Master Secret 생성 가능 : 관용 키, MAC 키, IV 생성 가능

8) Certificate Verify : Client가 자신의 인증서 유효성을 스스로 검증; 인증서에 서명2포함

☞ 서명2 : Master Secret과 미리 정의된 상수에 해시를 취하고, 클라이언트의 개인키로 암호화



SSL(Secure Socket Layer)

- **SSL Handshake Protocol 과정 설명**

- 단계 4(그림에서 9, 10, 11, 12번 과정) : 암호 알고리즘 정보를 서로 교환하고 프로토콜을 종료

9) Change Cipher Spec : Server에게 Data 암호화에 사용될 알고리즘 정보 전달

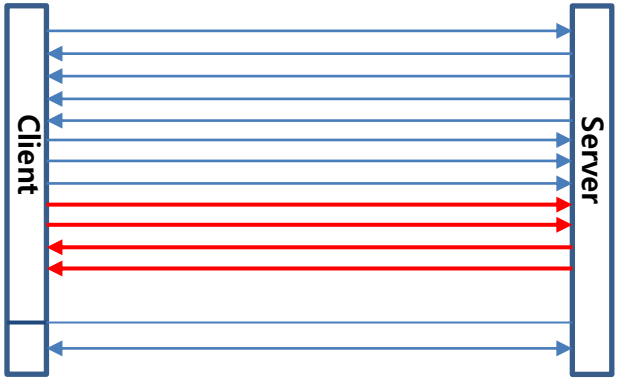
Client : "본격적인 레코드 통신에서 이 알고리즘을 사용할까?"

10) Finished : Server에게 완료 메시지 전송

11) Change Cipher Spec : Client에게 Data 암호화에 사용될 알고리즘 정보 전달

Server : "이제부터 이 알고리즘을 사용하자!"

12) Finished : Client에게 완료 메시지 전송



SSL/TLS

- **TLS(Transport Layer Security)**

  - 클라이언트와 서버 사이에 인증 및 암호화 통신을 위해 전송계층 상에서 사용되는 보안용 프로토콜
  - SSLv3에 기반하여 만들어진 인터넷 표준

- **SSL과의 차이점**

  - SSL 레코드 형식 : 헤더에서 Version 필드의 값이 Major Version은 3이고, Minor Version이 1
  - 메시지 인증 코드 : HMAC 사용
  - Master Secret의 계산식이 다름 : PRF(pseudo-random function)
  - 암호화 알고리즘, 키 교환 : Fortezza를 제외하고 SSLv3에서 사용한 모든 키 교환 기술 사용가능