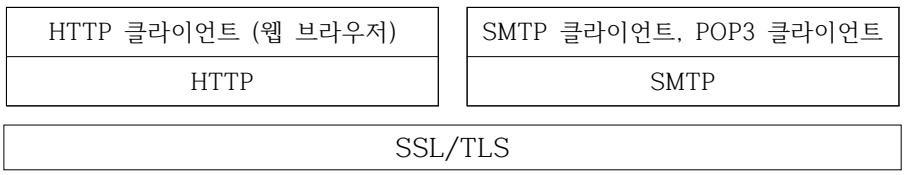
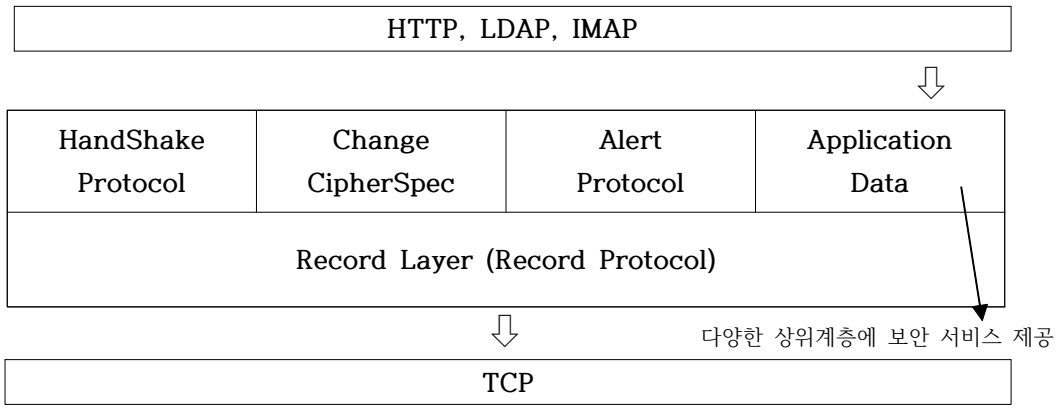
SSL 프로토콜 (Secure Socket Layer)

1) 웹 브라우저를 위한 보안 프로토콜
2) 이후에 버전 3.0에 대한 수정, 보완 과정을 거쳐 TLS(Transparent Layer Security) 라는 이름으로 표준화
3) SSL/TLS로 통신을 수행할 때의 URL은 https://로 시작
4) 전송 계층 상에서 클라이언트, 서버에 대한 인증 및 데이터 암호화 수행
5) SSL/TLS 상에 HTTP를 올리는 것 -> 프로토콜의 이중 구조 -> HTTP 통신 도청 방지



6) 인터넷 프로토콜 TCP 계층과 LDAP, IMAP과 같은 응용 계층 사이에서 동작



<제어 프로토콜 (Control Protocol)> **-> 2개의 계층으로 이루어진 프로토콜**

① HandShake Protocol
- Client 와 Server 간에 상호 인증
- 암호 알고리즘, 암호 키, MAC 알고리즘 등의 보안 속성을 협상, 세션 키 생성
- 레코드 단위에서 동작되는 행동
- SSL/TLS 의 레코드에 적용할 알고리즘과 키 교환
- 상호 송수신을 위한 암호화 스펙이 핸드쉐이크 프로토콜에 의해 공유

② Change CipherSpec Protocol
- 협상된 암호 규격과 암호 키를 이용하여, 추후의 레코드 계층 메시지를 보호할 것 명시
- 암호화 알고리즘과 보안 정책을 송수신 측 간에 조율하기 위해 사용
- 이 이후부터 협상된 압축, MAC, 암호화 방식 등이 적용됨을 상대방에게 알림

③ Alert Protocol
- 다양한 에러 메시지를 전달
- 2바이트로 구성, 첫 번째 byte에는 warning 또는 fatal이 들어가고, 두 번째 byte에는 handshake, change cipher spec, record protocol 수행 중 발생하는 오류 메시지 들어감

<레코드 프로토콜 (Record Protocol)>

④ Record Layer

- 합의된 암호 알고리즘을 이용한 암호화
- 합의된 MAC을 이용한 무결성 기능
- 메시지 분할, 합의된 압축 알고리즘을 이용한 메시지 압축 기능
- 상층에 위치하는 4개의 프로토콜에 기밀성, 무결성 같은 보안 서비스 제공

4) 기능

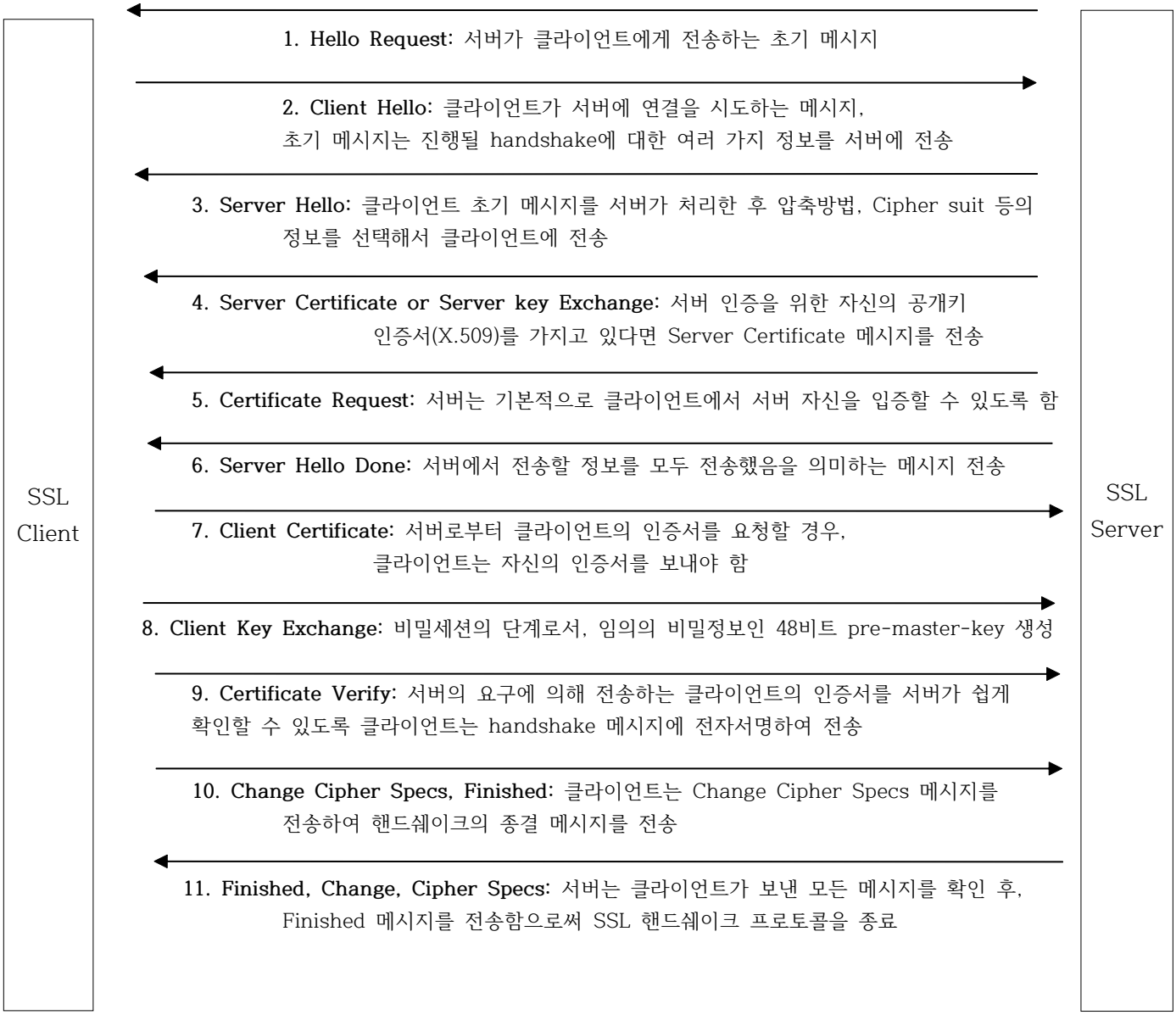
① 사이트 인증: 상대 사이트에 대한 신뢰성 있는 인증 제공

② 데이터 기밀성: 인터넷을 통해 전달되는 데이터 보호

③ 메시지 무결성: 웹 브라우저에서 웹 서버까지 전달되는 동안 메시지가 변경되지 않도록 보장

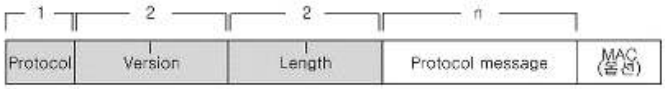
5) SSL HandShake Protocol (★★★)

- 핸드쉐이크는 서버와 클라이언트 사이에 인증과 암호화 과정 및 MAC 알고리즘을 교환하여 SSL 레코드에서 보내진 데이터들을 보호하기 위하여 암호화 key들을 사용



<SSL 레코드 프로토콜 데이터 포맷>

② Record Protocol 메시지 포맷



각 필드에 대한 길이와 포함하는 내용은 다음과 같다.

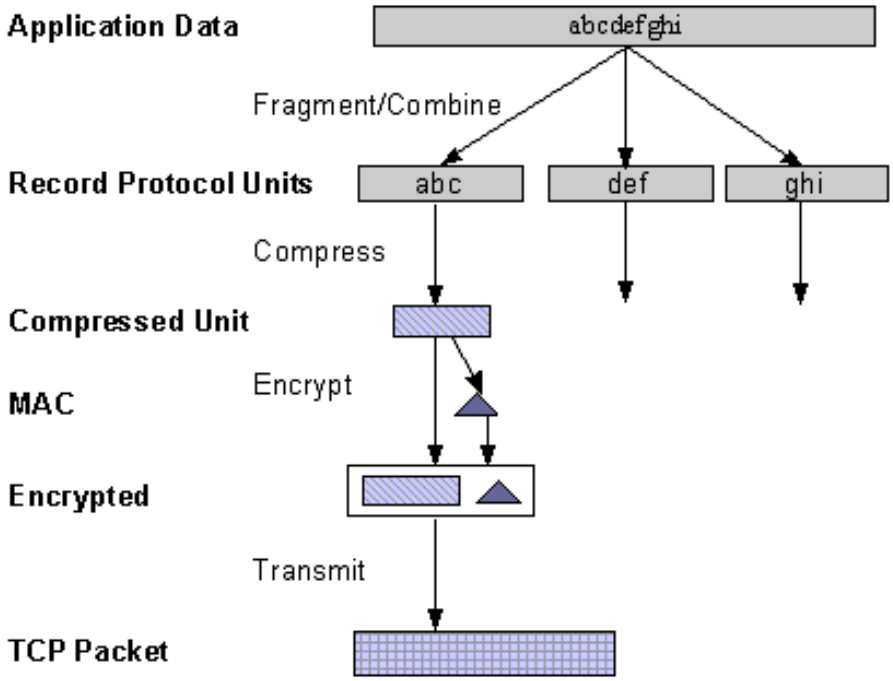
필드	길이(바이트)	설명
Protocol	1	Record Layer 프로토콜이 감싸고 있는 프로토콜이 무엇인지 표시 합니다. 즉 Record Layer 프로토콜 안의 내용이 어떤 프로토콜의 것인지를 표시 합니다.
Version	2	SSL의 버전을 표시 합니다. 주로 3.00이 사용 됩니다.
Length	2	Record Layer 프로토콜이 감싸고 있는 프로토콜의 내용의 길이를 표시 합니다. 이 길이는 2^14 의 값, 즉 16384바이트를 넘을 순 없습니다.
Protocol Message	16384바이트 이내의 길이	Record Layer 프로토콜이 감싸고 있는 프로토콜의 내용입니다.
MAC (옵션)	메시지 압축알고리즘에 따라 달라짐	Protocol Message 내용의 MAC 값입니다. 메시지 인증 기능을 사용할 경우 사용 됩니다. 따라서 이 필드의 사용은 옵션입니다.

Protocol 필드 값에 따른 상위 프로토콜은 다음과 같다.

필드 값	프로토콜
20	ChangeCipherSpec Protocol
21	Alert Protocol
22	Handshake Protocol
23	어플리케이션 프로토콜

<SSL 레코드 프로토콜 동작 방식>

- 데이터의 압축을 수행하여 안전한 TCP 패킷으로 변환하고, 데이터 암호화 및 무결성을 위한 메시지 인증을 수행



1. 전송할 메시지(Application Data)를 일정한 크기의 레코드 프로토콜 유닛으로 단편화 (Fragmentation은 2^14 바이트로 된다.)

2. 사전에 협의한 규칙에 맞게 SSL 핸드쉐이크 프로토콜을 통해 메시지를 압축하고, 전자서명을 붙임

Note: 현재 모든 주요 SSL 구현은 압축을 지원하지 않고 있다.

3. 압축된 유닛마다 해시를 이용한 MAC을 생성하고 핸드쉐이크가 끝난 후, 클라이언트와 서버는 MAC을 포함한 메시지를 암호화하여 통신

4. 마지막으로 SSL Record Header를 암호문에 붙인다. content-type, major version, minor version, compressed length 필드를 갖고 있음

5. 데이터의 압축을 수행하여 안전한 TCP 패킷으로 변환하고, 데이터 암호화 및 무결성을 위한 메시지 인증을 수행, 이를 TCP로 전달

Securing HTTP Communication

- SSL은 보통 웹 브라우저와 웹 서버 간의 HTTP 통신을 안전하게 만드는 데에 사용되며, 이 경우 non-secure한 HTTP를 사용하지 않는 것이 아니라, SSL 위에서 일반적인 plain HTTP를 구현하는 것임.
- HTTPS를 이용할 때, HTTP와 다른 서버 포트(default 443)를 사용하지 않는 것이 아니라, HTTPS라는 URL scheme을 사용하는 것