Network Security Essential

- 5장 전송레벨 보안 -

명 세인(sein@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

웹 보안

- 웹 (WWW: World Wide Web)
  - 개요
    - 인터넷과 TCP/IP 인트라넷 상에서 운영하는 기본적인 서버/클라이언트를 구현하는 응용 프로그램
    - 웹과 서비스를 구현함에 따라 보안서비스 적용에 여러 문제점이 있음
  - 웹 보안의 필요성
    - 웹의 역사가 짧지만 내부 구성과 적용되는 새로운 기술들은 복잡하여 발견되지 않은 취약점이 많음
    - 웹 서버가 취약해지고 서비스 서버를 통한 관리 서버침투가 발생할 수 있음
    - 일반 사용자는 보안정보를 알 필요 없이 충분한 보안성을 제공

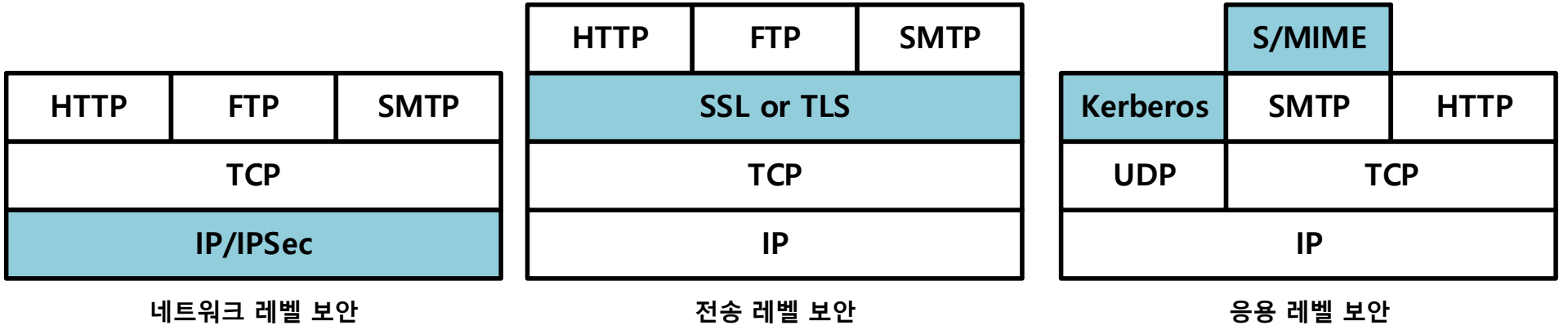
웹 보안

- 웹 보안
  - 웹 보안 위협 정리 표

	위협	피해 사항	대응 방법
Integrity	• 사용자 데이터 변경 • 트로이 목마 브라우저 • 메모리의 변경 • 전송 중 메시지 트래픽 변경	• 정보의 손실 • 하드웨어(기계) 침해 • 다른 위협에 대한 취약성	• 암호학적 Check Sum
Confidentiality	• Net 도청 • 서버 정보 갈취 • 클라이언트 데이터 갈취 • 네트워크 구성에 대한 정보 습득 • 서버와 통신중인 클라이언트 정보 습득	• Loss of information • Loss of privacy	• 암호화, • 웹 프록시
DoS	• 사용자 스레드(Thread) 중단 • 과도한 가짜 위협 전송 • 메모리, 디스크 역영 차지 • DNS 공격을 통한 고립	• Disruptive • Annoying • 사용자 작업 방해	• 예방이 어려움
Authentication	• 신분위장 • 데이터 위조	• 사용자 식별 오류 • 가짜정보를 진짜로 오인	• 암호학적 기술

웹 보안

- 웹 보안
  - 웹 트래픽 보안
    - 웹이 제공하는 서비스 유형
    - 웹이 사용하는 메커니즘 (암호화, 인증, 접근제어 등)
    - 웹의 응용 범위 또는 TCP/IP 프로토콜 계층에서 웹이 있는 상대적인 위치
      - S/MIME: Secure/Multipurpose Internet Mail Extensions



목 차

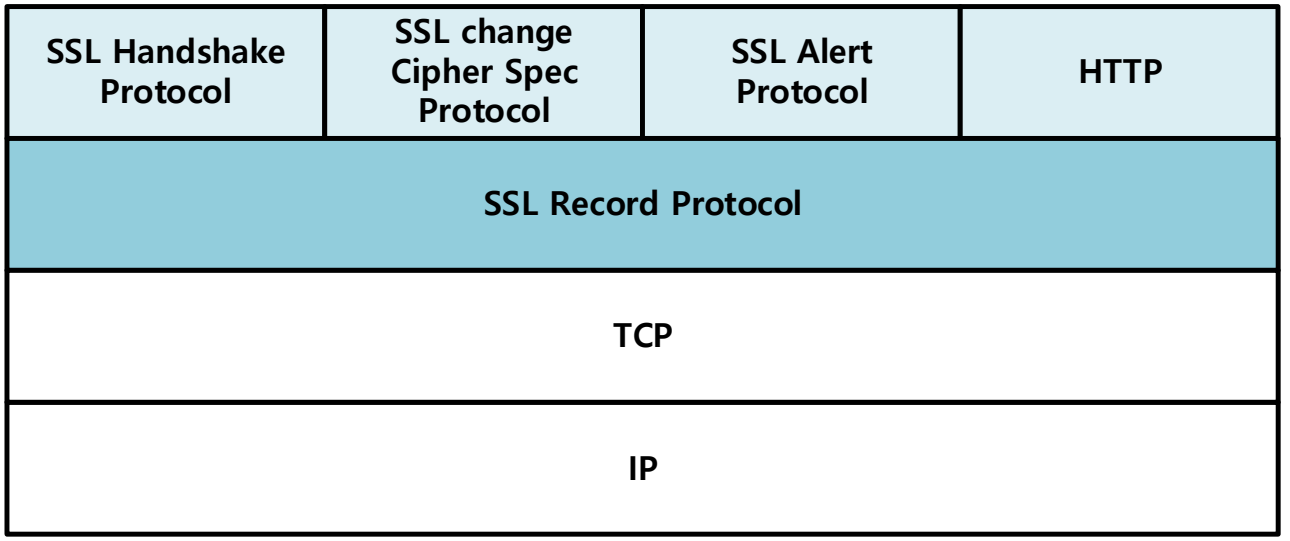
- 웹 보안
- 안전소켓 계층 보안 (SSL)
- 전송계층 보안 (TLS)
- HTTPS
- SSH

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 개요
    - 넷 스케이프 사에서 1994년 최초 프로토콜 스펙 정의, 웹 서버와 브라우저간의 안전한 통신을 위함
    - 신뢰할 수 있는 TCP 위에 세션 개념으로서 SSL 사용
      - 상위 계층 프로토콜에 기본적인 보안 서비스를 제공 (HTTP)
      - 기존의 TCP/IP에 없는 보안 개념을 추가
    - SSL을 기반으로 TLS를 설계

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 개요
    - 프로토콜 구성
      - SSL Record Protocol
      - 3가지 상위계층 프로토콜을 SSL의 일부로서 정의
        - Handshake protocol (Hello)
        - Change Cipher Spec Protocol (암호명세 변경)
        - Alert Protocol (경고)



안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL의 개념
    - 세션(Session)
      - 클라이언트와 서버 사이의 핸드셰이크 Protocol을 통한 연결
      - 다수의 연결이 공유하는 암호적 매개변수를 적의(각 연결에 필요한 보안 협상을 위해 사용)
      - 세션 파라미터 정리 표

파라미터	의미
Session Identifier	서버가 선택하는 임의의 바이트 열
Peer Certificate	대등 X.509v3 인증서 (NULL가능)
Compression Method	암호화 하기 전 압축에사용되는 알고리즘
Cipher Spec	MAC계산에 사용되는 암호 또는 해시 알고리즘과 해시크기 등을 정의
Master Secret	클라이언트와 서버가 공휴하는 48Byte 비밀 값
Is Resumable	재시작 여부, 새 연결을 시작하기 위해 세션을 사용할 수 있는지에 대한 Flag

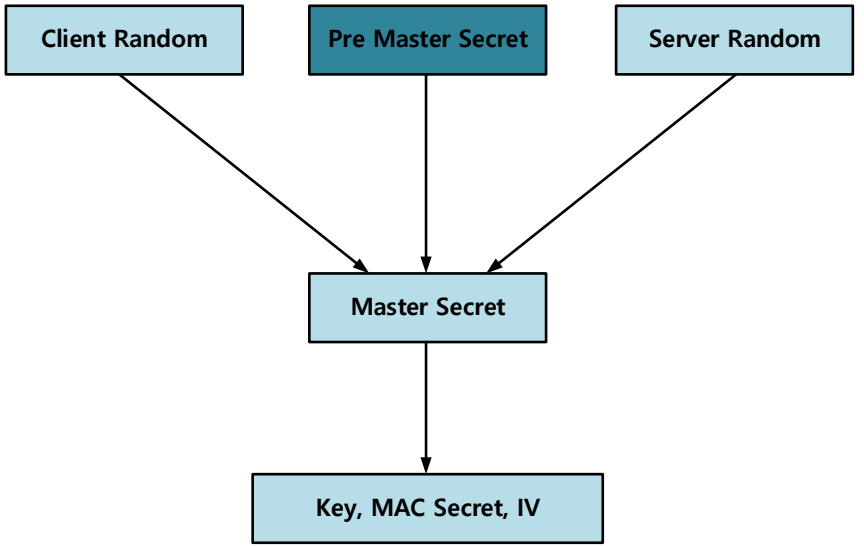
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL의 개념
    - 연결 (Connection)
      - Peer-to-Peer의 관계
      - 모든 연결은 일시적이며 하나의 세션과 연관(Associated)됨
      - 연결 파라미터 정리 표

파라미터	의미
Server/Client Random	각 연결에 사용하는 서버/클라이언트가 선택하는 바이트 열
Server write MAC Secret	서버가 보낸 데이터로 MAC계산시 사용하는 비밀키
Client write MAC Secret	클라이언트가 보낸 데이터로 MAC계산시 사용하는 비밀키
Server write Key	서버가 데이터를 암호화, 클라이언트가 복호화할 때 사용하는 대칭 키
Client write Key	클라이언트가 데이터를 암호화, 서버가 복호화할 때 사용하는 대칭 키
Initialization Vectors	CBC 모드로 블록 암호를 사용할 때 사용되는 초기 값
Sequence Numbers	송/수신하는 메시지에 대한 순서번호 (MAX: 2^{64}-1)

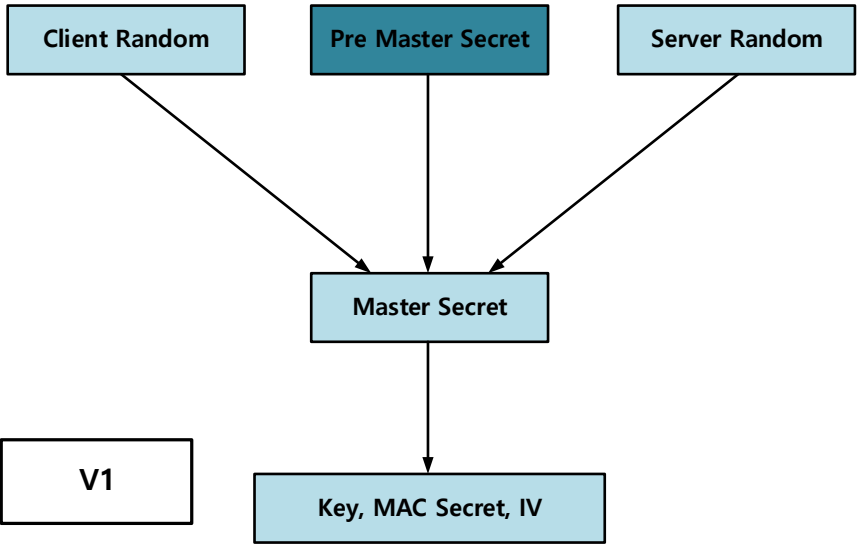
안전 소켓 계층 보안

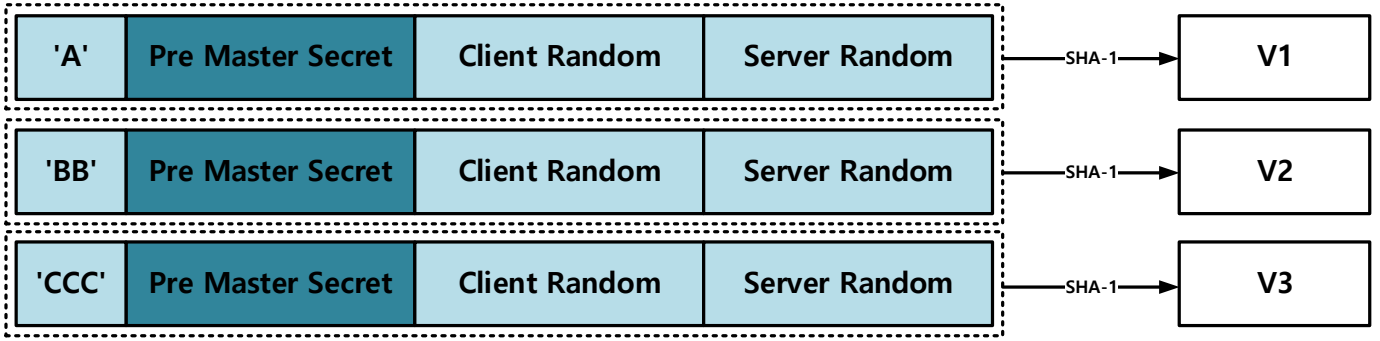
- SSL (Secure Socket Layer)
  - SSL의 개념
    - 파라미터의 생성
      - Client/Server 난수: PRF를 이용해 만든 32bytes 난수
        - Nonce로 사용: timestamp(4bytes) + Random value(28bytes)
      - Premaster Secret: Client가 PRF를 이용해 만든 난수(48bytes)
        - 세션키로 사용: client_version(2bytes) + random value(46bytes)

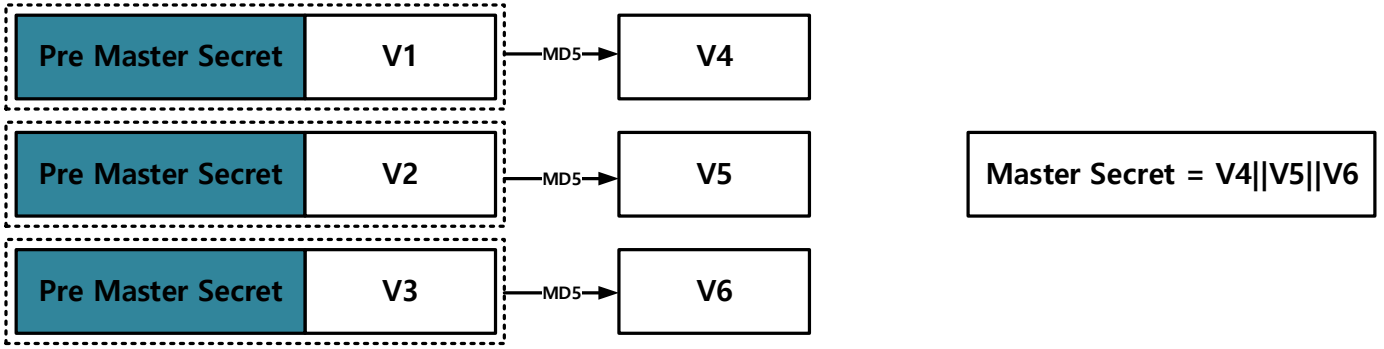


안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL의 개념
    - 파라미터의 생성
      - Master Secret의 생성 그림



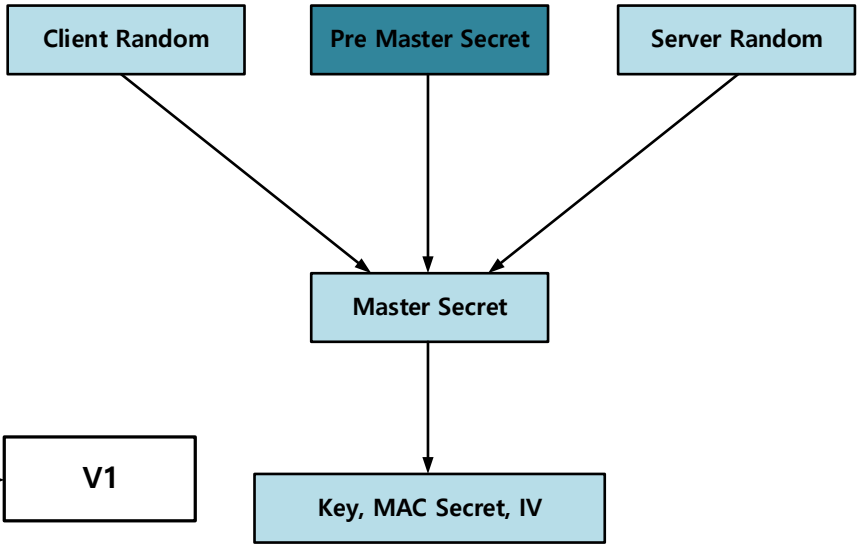


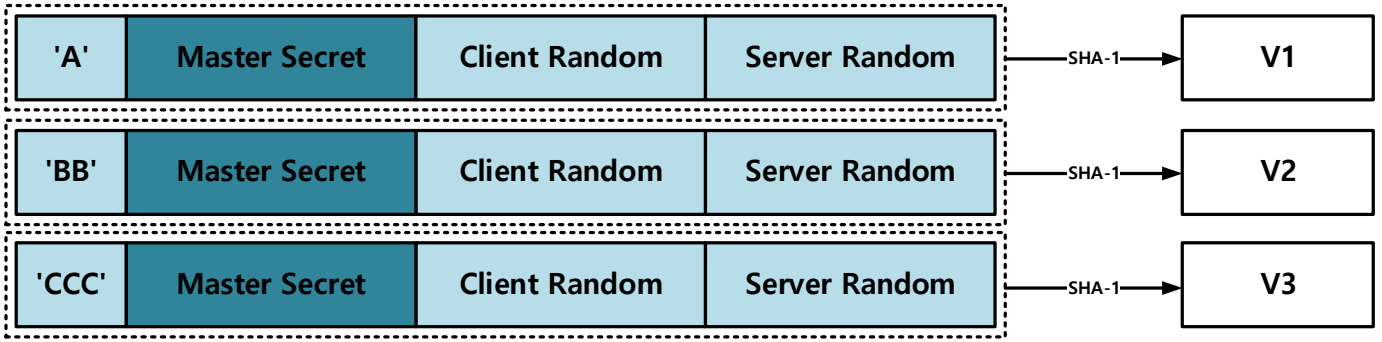


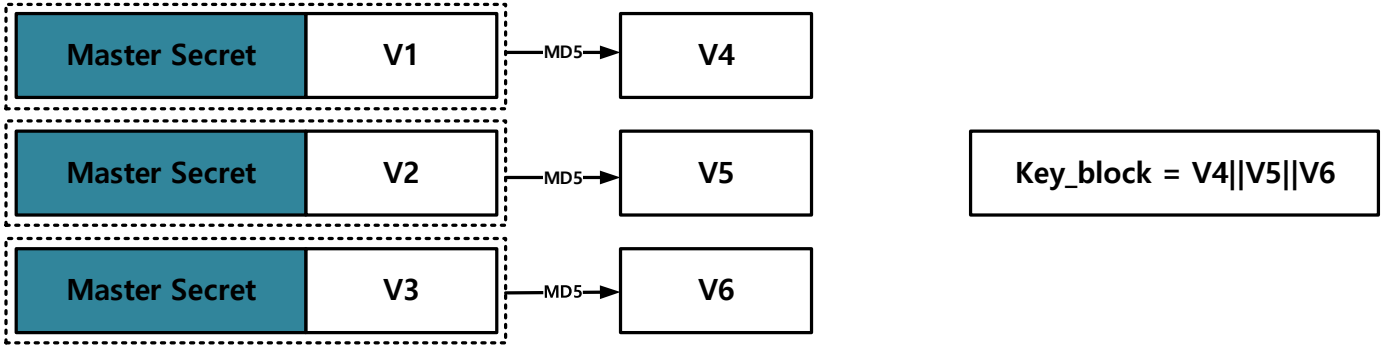
Master Secret = V4||V5||V6

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL의 개념
    - 파라미터의 생성
      - Key block 생성 그림



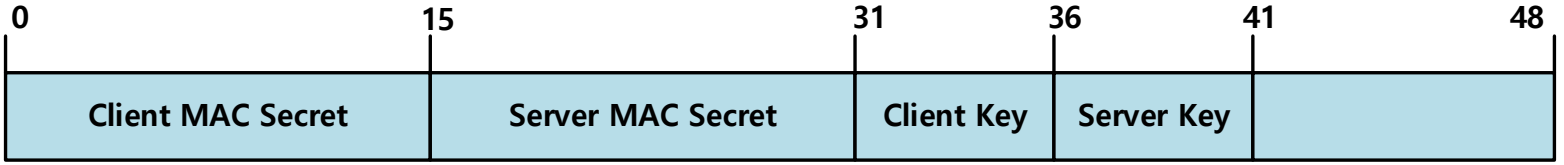




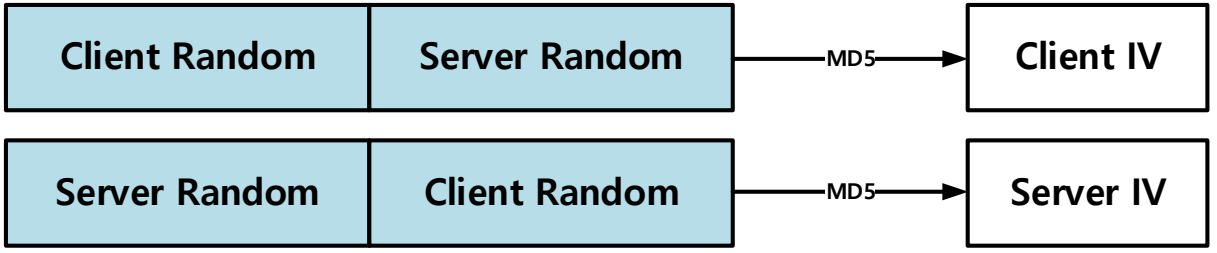
Key_block = V4||V5||V6

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL의 개념
    - 파라미터의 생성
      - Key block 구조

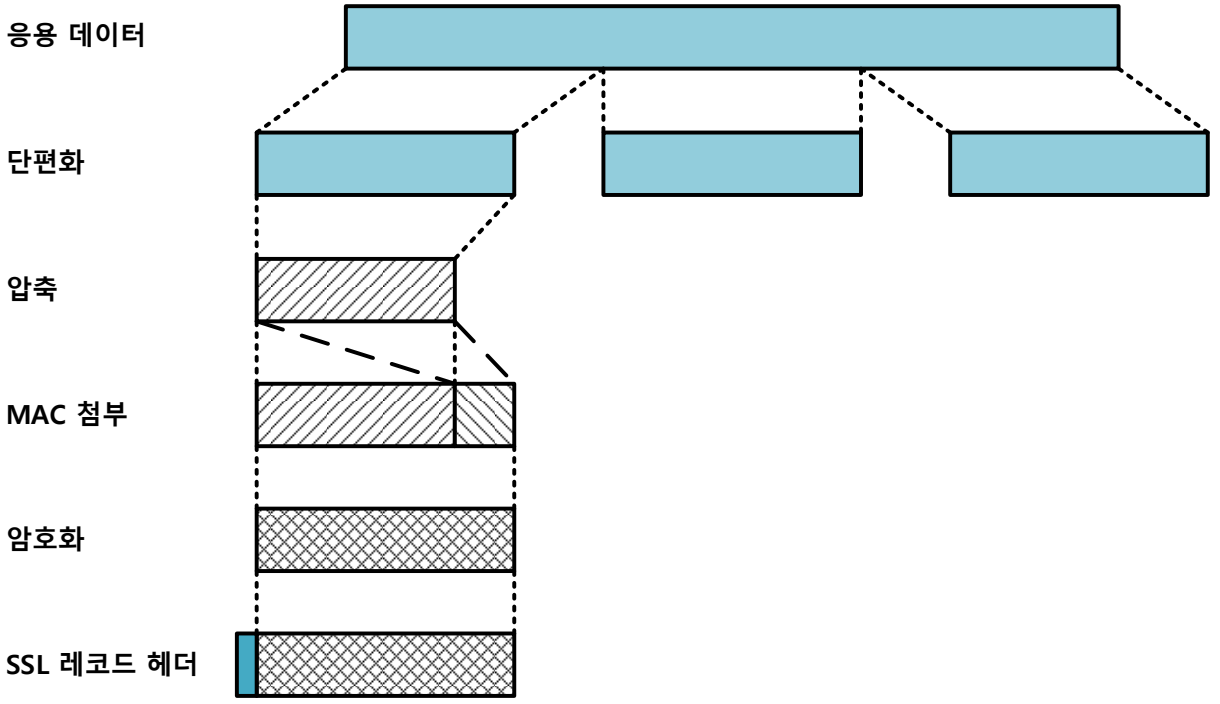


      - Client / Server IV 계산



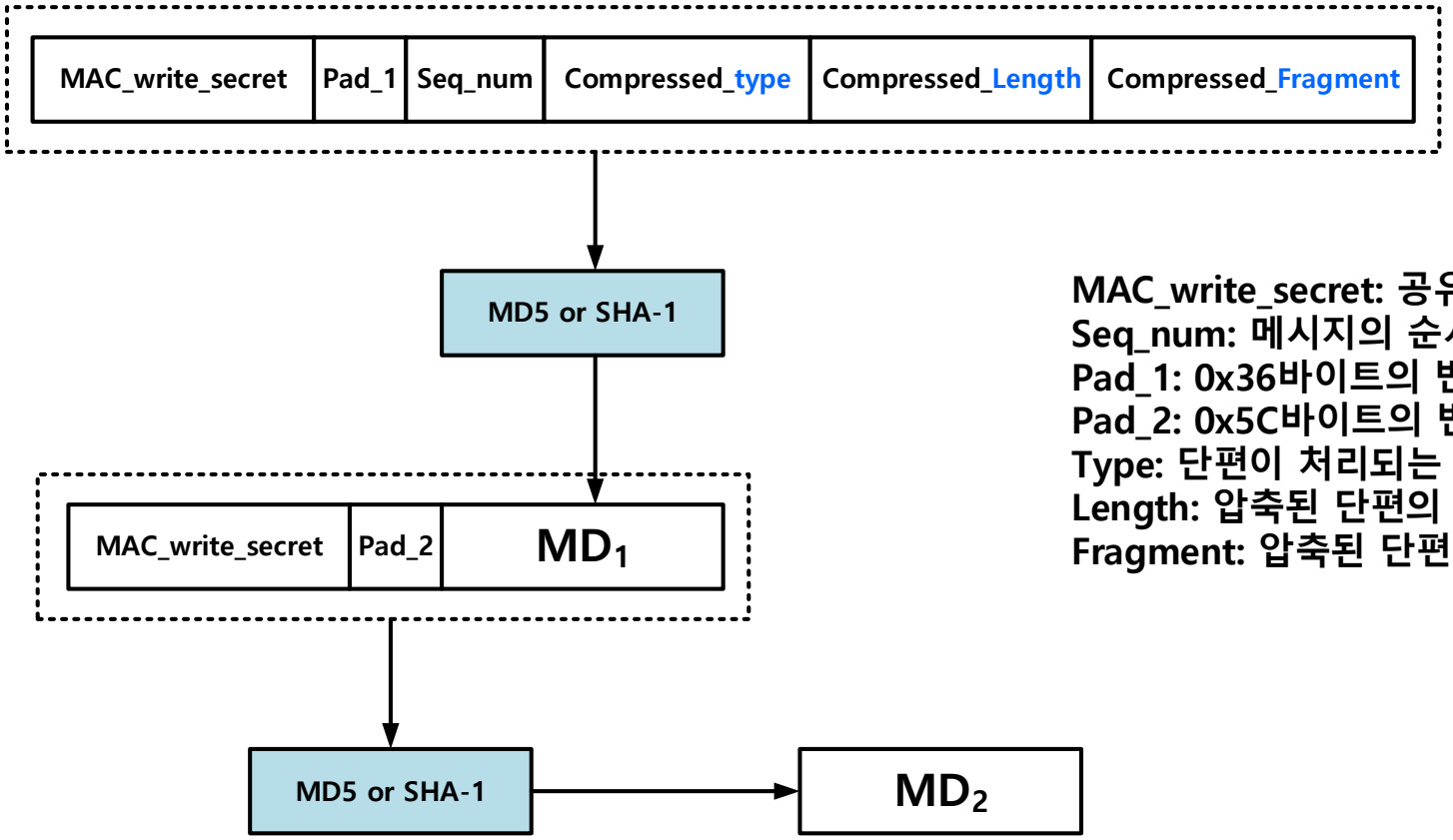
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL 레코드 프로토콜 동작
    - 기능
      - 상위 계층 메시지를 TCP에 전달
      - 기밀성, 메시지 무결성(MAC)제공
    - 전체 동작 그림



안전 소켓 계층 보안

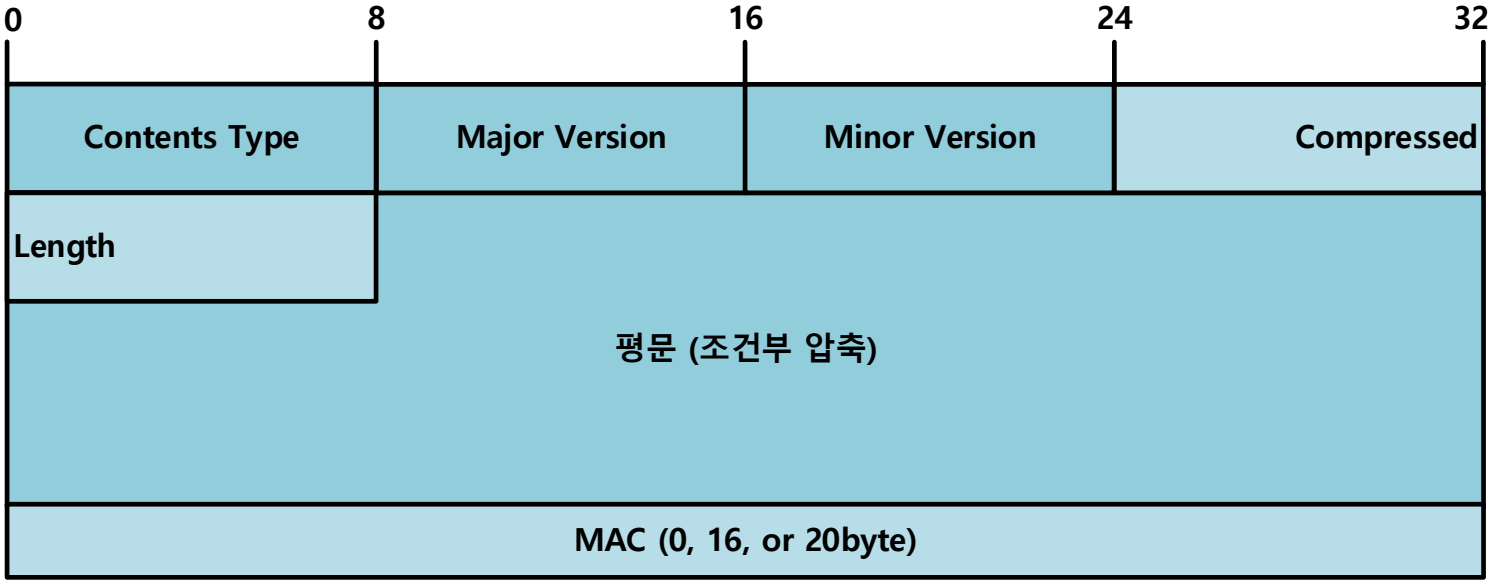
- SSL (Secure Socket Layer)
  - SSL 레코드 프로토콜 동작
    - MAC 계산 동작 그림



MAC_write_secret: 공유 비밀키
Seq_num: 메시지의 순서번호
Pad_1: 0x36바이트의 반복 (MD5는 48번, SHA-1은 40번)
Pad_2: 0x5C바이트의 반복 (MD5는 48번, SHA-1은 40번)
Type: 단편이 처리되는 상위-계층 프로토콜
Length: 압축된 단편의 길이
Fragment: 압축된 단편(평문의 단편)

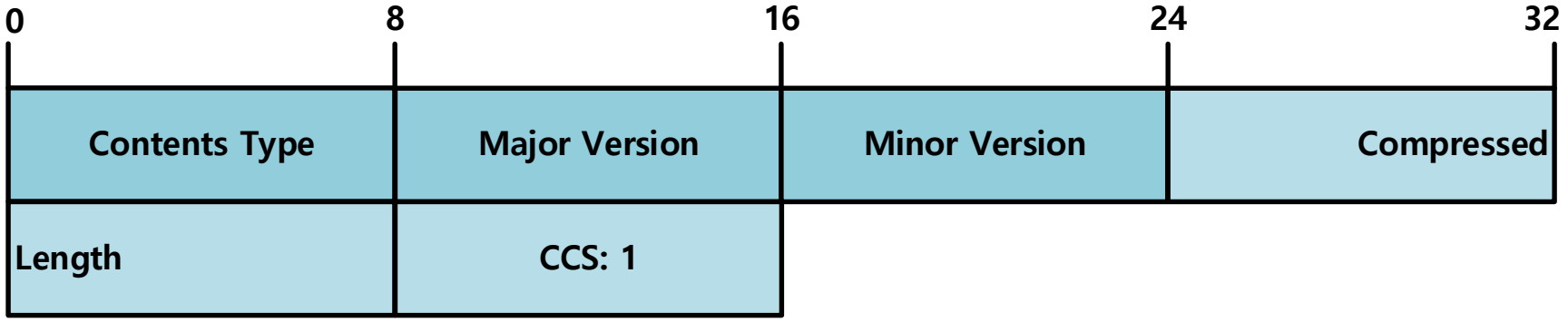
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL 레코드 프로토콜 동작
    - Record 형식
      - Contents Type: 상위 Protocol을 식별
        - 20: 암호명세 변경 프로토콜
        - 21: 경고 프로토콜
        - 22: 핸드쉐이크 프로토콜
        - 23: 응용 계층



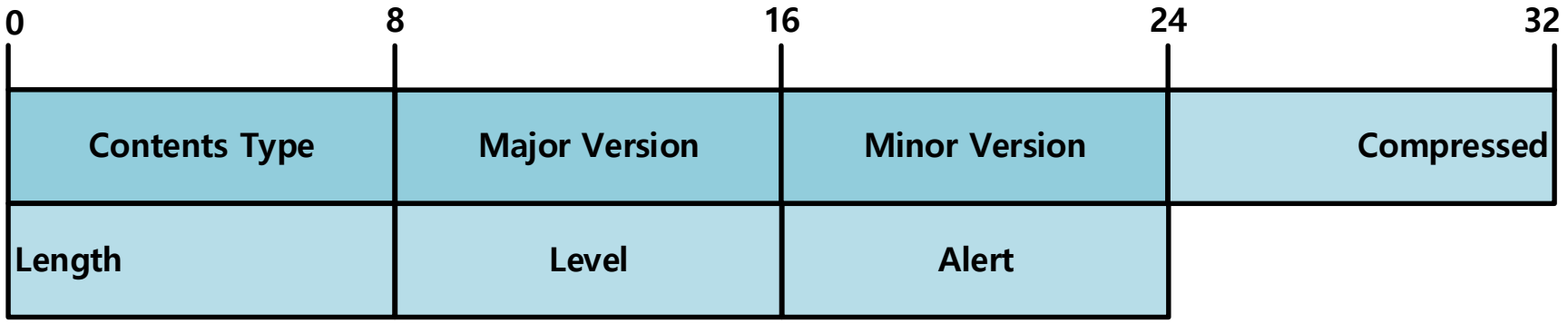
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 암호 명세 변경 프로토콜 (Change Cipher Spec protocol)
    - 프로토콜 역할
      - 1byte로 구성된 메시지로 구성
      - Handshake 프로토콜로 협상된 내용이 이후의 통신에 적용됨을 알림



안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 경고 프로토콜 (Alert Protocol)
    - 2byte로 구성된 프로토콜, 경고(Warning)와 심각(Fatal)을 알림
      - 심각: SSL은 즉시 연결을 단절, 현재 세션에서 새로운 연결 불가능, 현재 세션의 다른 연결은 지속
    - Level 바이트에서 경고와 심각을 구별
    - Alert 바이트는 특정 경고를 나타내는 코드



안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 경고 프로토콜 (Alert Protocol)
    - 항상 심각(Always Fatal)경고의 종류 표

상황	의미
Unexpected_Message	적합하지 않은 메시지의 수신
Bad_Record_MAC	부정확한 MAC 수신
Decompressed_Failure	압축해제함수에 적합하지 않은 입력(압축풀기 불가 또는 최대 허용길이보다 큰 경우 등)
Handshake_Failure	사용할 수 있는 옵션이 주어졌지만 송신자와 협상 불가
Illegal_Parameter	핸드셰이크 메시지 안의 한 필드가 범위 밖이거나 다른 필드와 맞지 않음

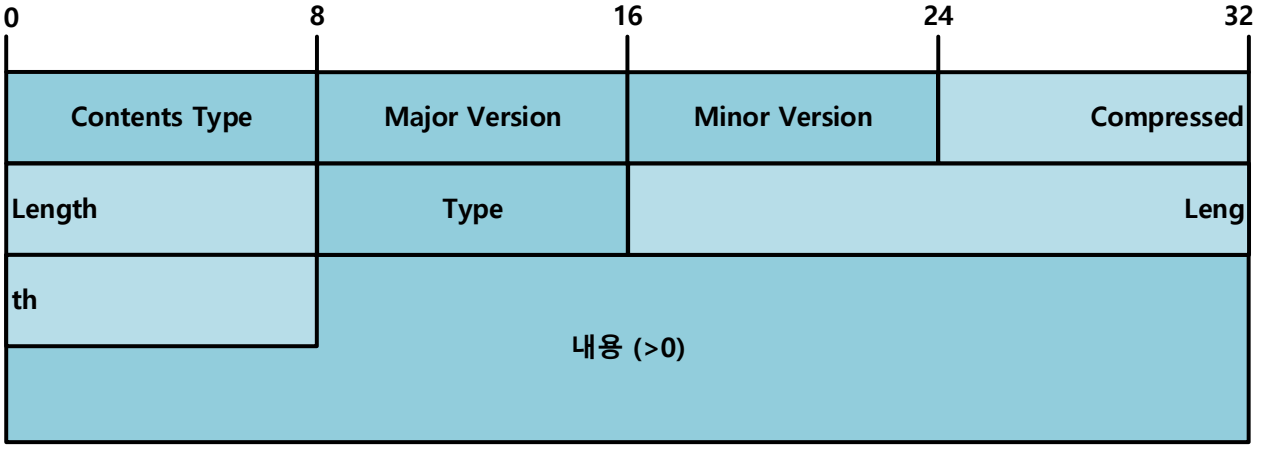
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 경고 프로토콜 (Alert Protocol)
    - 일반적인 경고의 종류 표

상황	의미
Close_Notify	송신자가 이 연결에서 더 이상 메시지를 보내지 않을 것을 수신자에게 알림 (각 개체는 연결 종료 전에 이 경고를 보내야 함)
No_Certificate	적절한 인증서가 없는 경우 인증서 요청에 대한 응답
Bad_Certificate	수신된 인증서에 문제가 있음 (검증하지 않은 서명 등)
Unsupported_Certificate	수신된 인증서의 유형을 지원하지 않음
인증서 상태	Revoked, Expired, Unknown등의 인증서 문제가 발생

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 데이터를 전송하기 전에 수행
      - 서버와 클라이언트가 서로를 인증하고 암호, MAC알고리즘 레코드 데이터 보호에 사용될 암호화 키를 협상
    - 클라이언트와 서버가 교환하는 여러 메시지로 구성
      - 메시지 형식
        - Type: 10개의 메시지를 식별
        - Length: 메시지의 길이를 바이트로 나타냄
        - Content: 메시지와 연관된 매개변수



안전 소켓 계층 보안

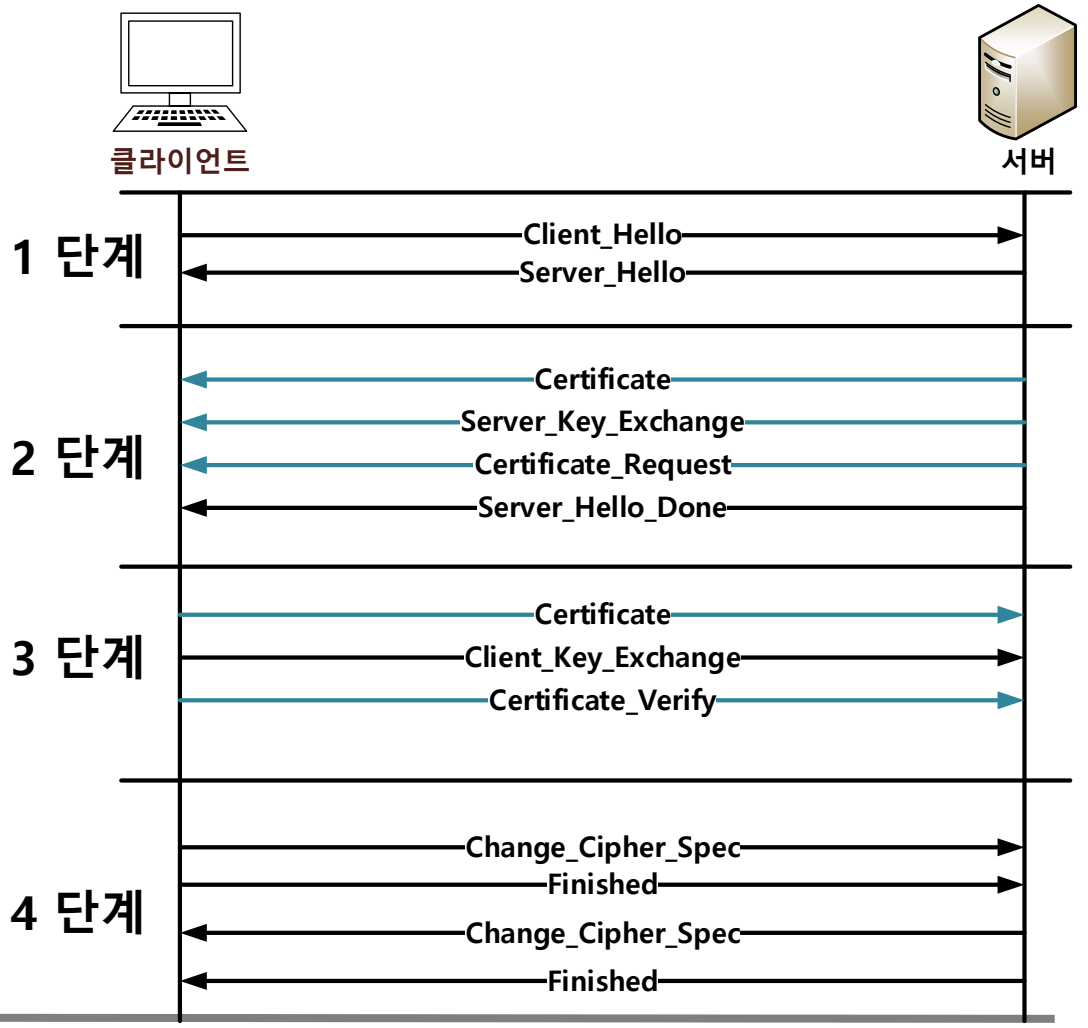
- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 클라이언트와 서버가 교환하는 여러 메시지로 구성
      - 메시지 유형정리 표

메시지 유형	매개 변수
Hello_Request	Null
Client_Hello	Version, Random, Session ID, Cipher suite, Compression Method
Server_Hello	Version, Random, Session ID, Cipher suite, Compression Method
Certificate	연속된 X.509v3 인증서
Server_Key_Exchange	Parameters, Signature
Certificate_Request	Type, Authorities
Server_Done	Null
Certificate_Verify	Signature
Client_Key_Exchange	Parameters, Signature
Finished	Hash Value

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 프로토콜 동작

단계	동작
1	보안 기능 수집: 버전, 세션 ID, 암호 조합, 압축 방법, 조기 난수
2	(서버가 필요하면: 인증서, 키교환을 보내고 인증서를 요청) Hello단계의 끝을 알림
3	(인증서가 요청되면 응답, 확인을 보낼수 도 있음) 클라이언트가 키 교환을 보냄
4	암호 조합을 교환하고 협상 프로토콜을 종료



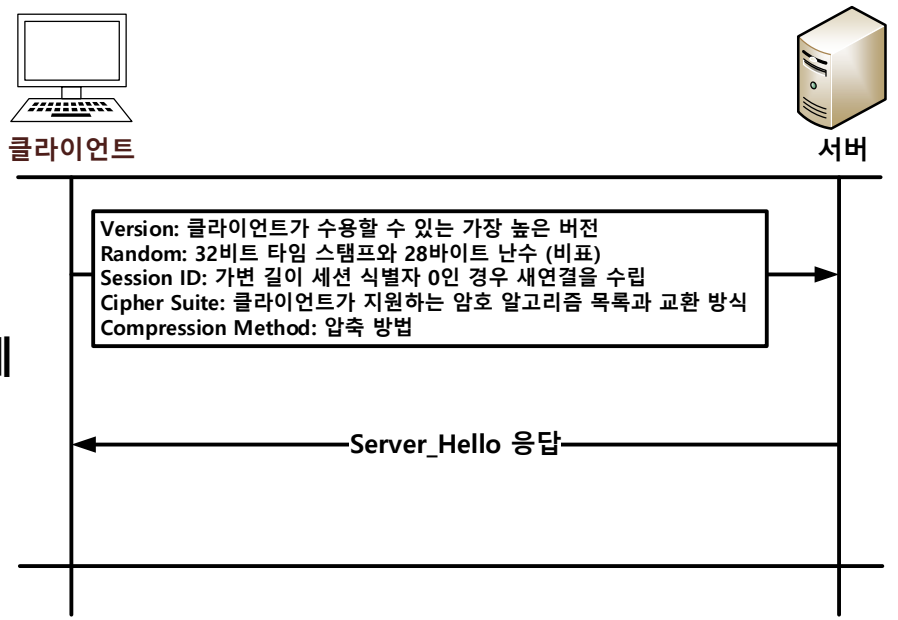
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 1단계: 보안 기능 설정

이름	역할
Session ID	값이 0이 아니면 동일한 암호값을 사용, 0이면 새로운 세션을 위한 값을 설정
Cipher Suite	5가지의 키 교환 방법(RSA, Diffie-Hellman 등) 을 정의하고 암호 명세로 사용할 암호 알고리즘을 나열

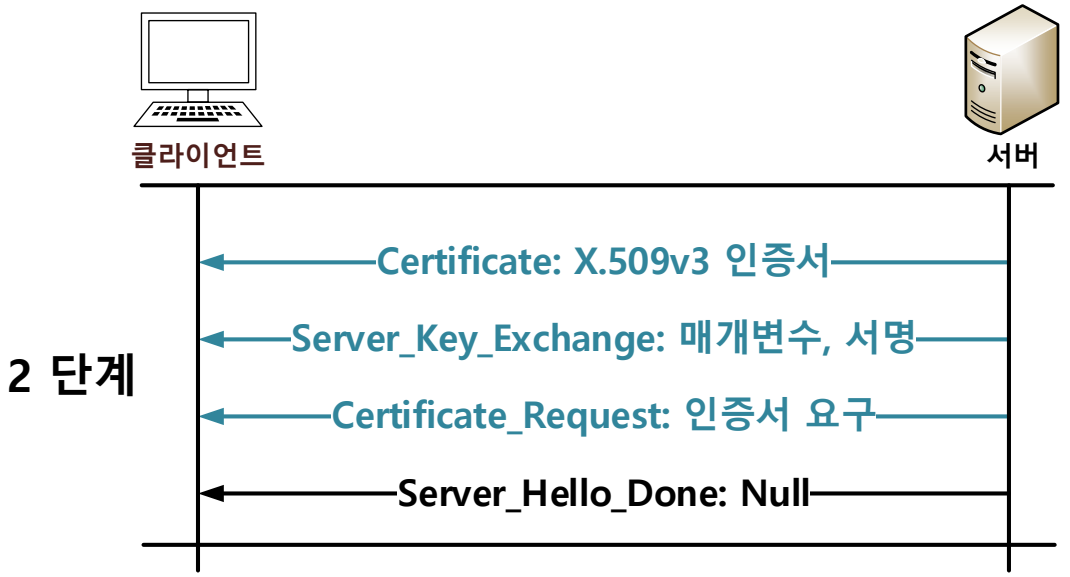
암호 명세	의미
Cipher Algorithm	RC4, DES 등
MAC Algorithm	MD5 또는 SHA-1
Cipher Type	스트림 또는 블록
Is Exportable	참 또는 거짓
Hash Size	0, 16(MD5), 또는 20(SHA-1) Bytes
Key Material	Write 키 생성에 사용할 데이터를 포함하는 바이트 열
IV Size	CBC 암호화에 사용하는 초깃값의 크기

1 단계



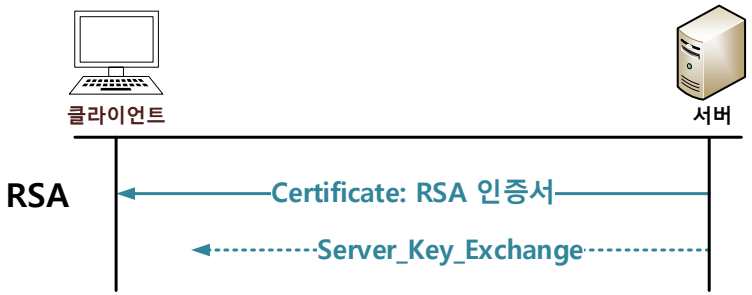
안전 소켓 계층 보안

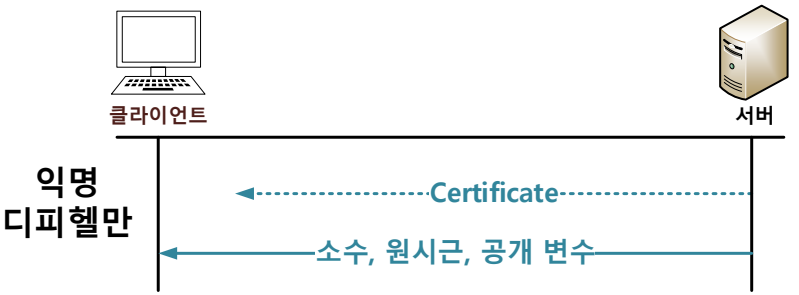
- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 2단계: 서버 인증과 키 교환
      - 키 교환의 유형
        - RSA
        - 익명 디피헬만
        - 임시 디피헬만
        - 고정 디피헬만



안전 소켓 계층 보안

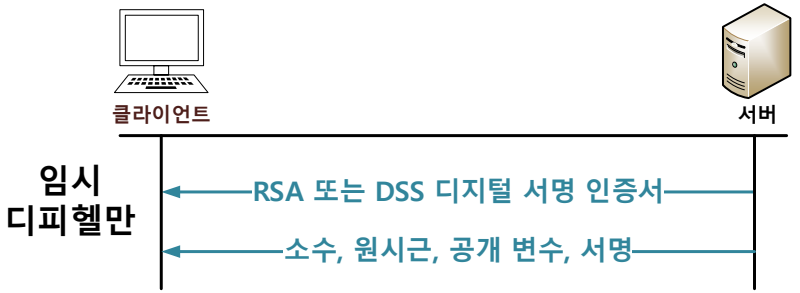
- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 2단계: 키 교환의 유형
      - RSA
        - 클라이언트는 Pre-master secret을 서버의 공개키로 암호화하여 전송
        - Pre-master secret: client_version(2bytes)+난수(4bytes)
      - 익명 디피헬만
        - p, a, g^x를 평문 전송
        - Premaster secret: g^{cs} mod p

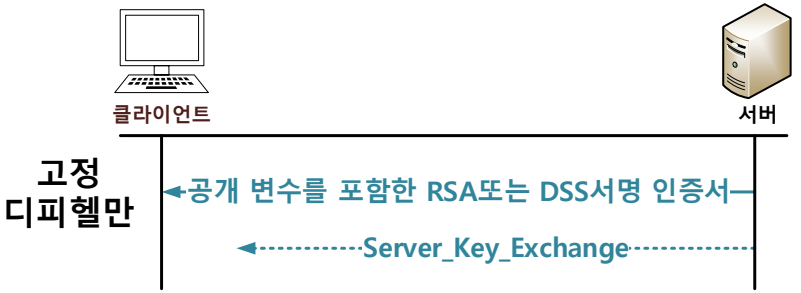




안전 소켓 계층 보안

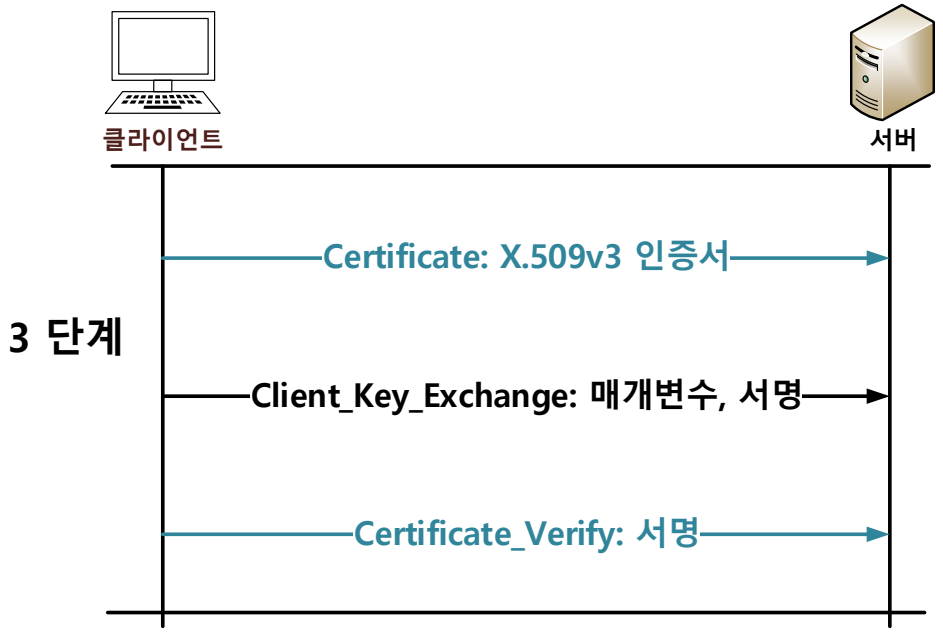
- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 2단계: 키 교환의 유형
      - 임시 디피헬만
        - 개인키로 서명된 p, a, g^x를 전송
        - Pre-master secret: g^{cs} mod p
      - 고정 디피헬만
        - 각 개체는 고정 DH 매개변수를 생성할 수 있는 매커니즘이 존재
        - 인증서의 내용에 g^x를 포함, 인증기관의 개인키로 서명하여 인증
        - RSA 또는 DSS 인증서 활용
        - Pre-master secret: g^{cs} mod p





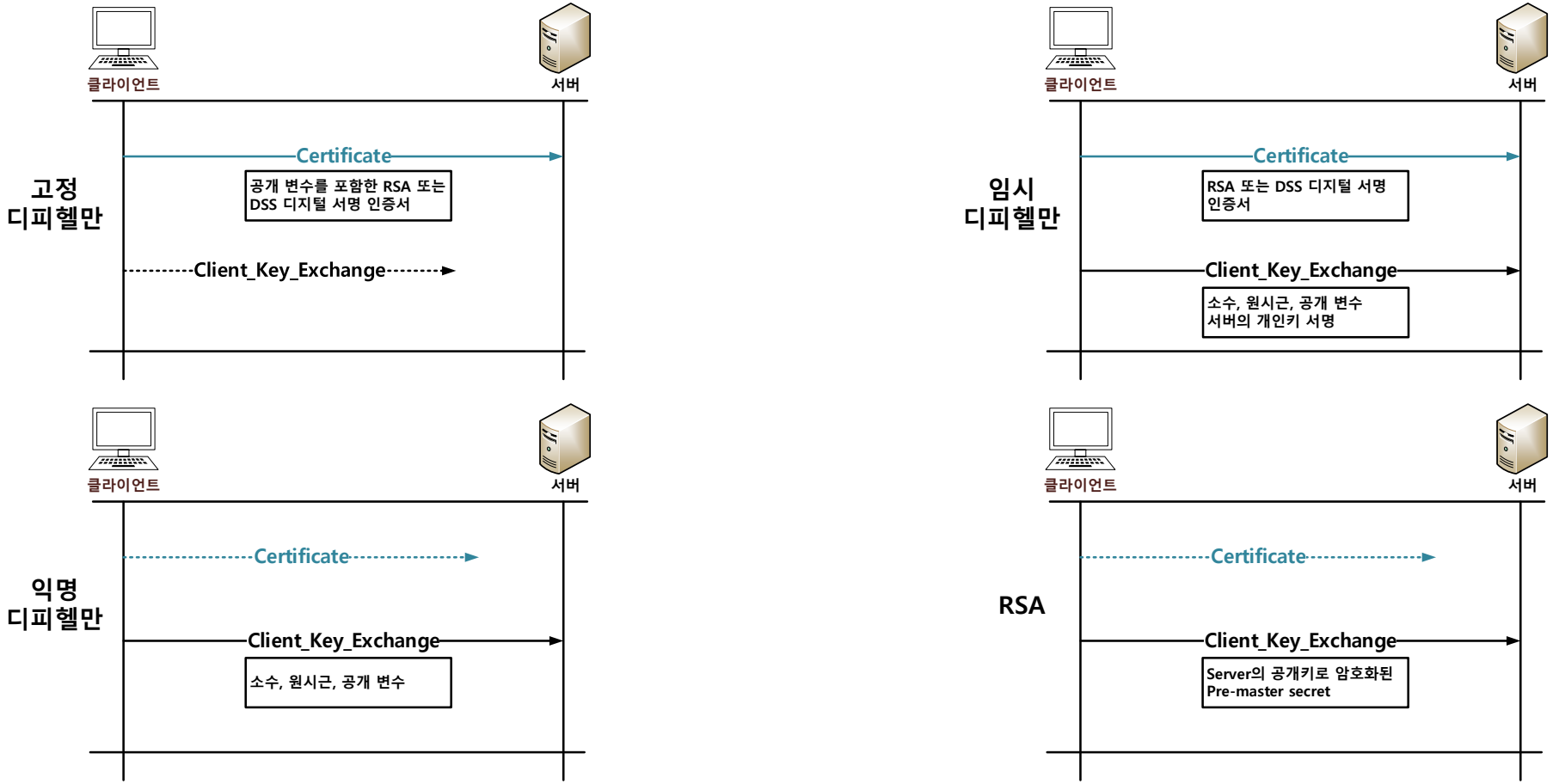
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 3단계: 클라이언트 인증과 키 교환
      - 클라이언트가 자신의 인증서의 유효함을 스스로 검증



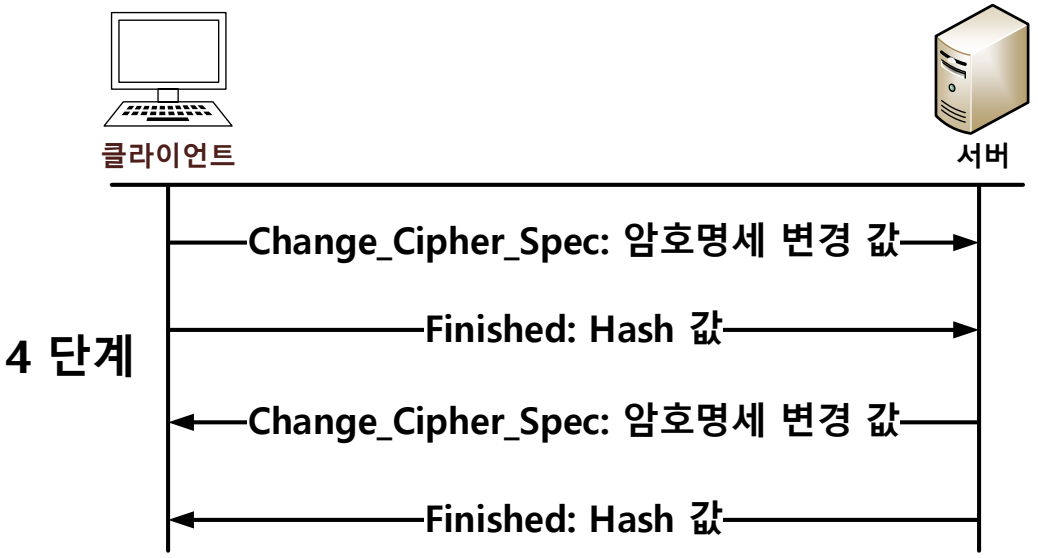
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 3단계: 클라이언트 인증과 키 교환의 4가지 유형



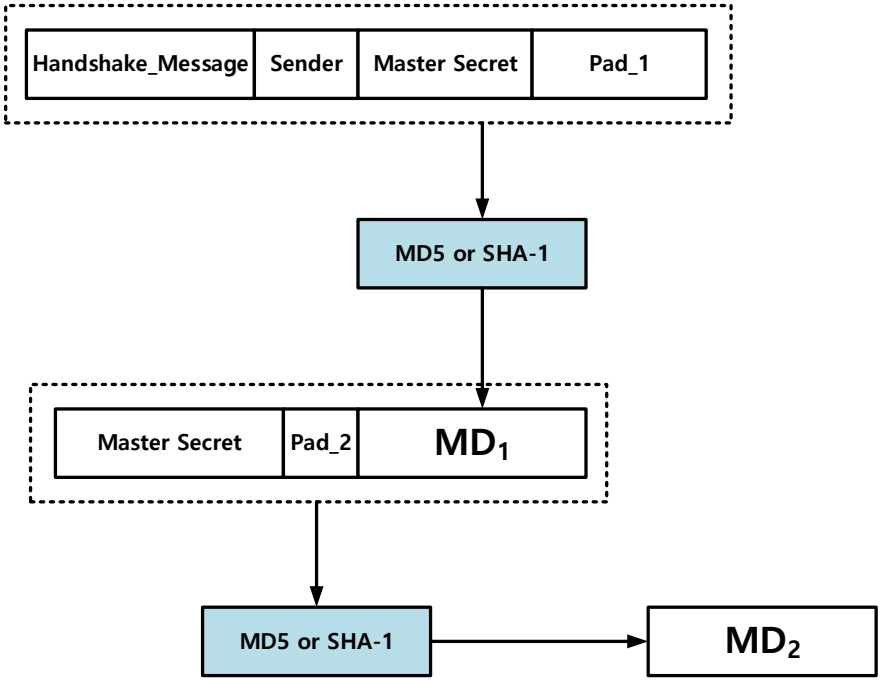
안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 4단계: 종료
      - 안전한 연결 종료
      - Finished 메시지는 키 교환과 인증 과정이 성공적임을 확인 (해시값)



안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - 협상 프로토콜 (Handshake Protocol)
    - 4단계: 종료
      - Finished 메시지의 해시 계산



Sender: Client-0x434C4E54, Server-0x53525652
Pad_1: 0x36바이트의 반복 (MD5는 48번, SHA-1은 40번)
Pad_2: 0x5C바이트의 반복 (MD5는 48번, SHA-1은 40번)

안전 소켓 계층 보안

- SSL (Secure Socket Layer)
  - SSL과 전송계층 보안
    - 전송계층 보안은 SSL을 인터넷 표준 버전이 되도록 IETF 표준
      - RFC 2246으로 제안된 인터넷 표준
    - 주요 차이점
      - MAC 계산 알고리즘과 범위
      - Master Secret 계산
      - Finished 메시지 계산에 의사 난수 함수(PRF: Pseudo Random Function)의 사용

목 차

- 웹 보안
- 안전소켓 계층 보안 (SSL)
- 전송계층 보안 (TLS)
- HTTPS
- SSH

전송계층 보안

- TLS (Transport Layer Security)
  - TLS 레코드 형식
    - SSL의 레코드 형식과 동일하지만 버전필드에서 주 버전은3, 부 버전은 1로 표기
  - 메시지 인증코드
    - TLS에서는 HMAC을 사용
      - $HMAC_k(M) = H[(K^+ \text{ ^ } opad) \,||\, H[K^+ \text{ ^ } ipad \,||\, M]]$

H: 해시 함수 (MD5 or SHA-1)
M: 입력 메시지
K^+: 비밀키 0을 패딩하여 해시코드 블록 길이로 만든 값
(MD5 와 SHA-1의 블록 길이: 512bits)
ipad: 0x36바이트의 64번 반복
opad: 0x5C바이트의 64번 반복

전송계층 보안

- TLS (Transport Layer Security)
  - 의사 랜덤 함수 (PRF: Pseudo Random Function)
    - 원하는 길이만큼의 데이터 출력을 위해 HMAC_hash()연산을 반복 수행할 수 있음

PRF_hash(secret, seed) = HMAC_hash(secret, A(1) || seed) ||
HMAC_hash(secret, A(2) || seed) ||
HMAC_hash(secret, A(3) || seed) || . . .

A(0) = seed
A(i) = HMAC_hash(secret, A(i-1))

전송계층 보안

- TLS (Transport Layer Security)
  - 경고 코드
    - No_Certificate만 제외하고 SSLv3에 정의된 경고코드를 지원하며, 다양한 상황에 대한 경고를 위해 경고코드가 추가됨
  - 암호 도구 (Cipher Suites)
    - 키 교환: Fortezza를 제외한 SSLv3 키 교환 지원
    - 대칭 암호 알고리즘: Fortezza를 제외한 대칭 암호 알고리즘 지원
  - 인증 확인과 종료 메시지
    - TLS에서 Finish메시지 생성계산이 SSL과 다름

      PRF(master_secret, finished_label, MD5(handshake_messages) || SHA-1(handshake_messages))

전송계층 보안

- TLS (Transport Layer Security)
  - 암호 계산
    - pre_master_secret 계산은 SSLv3와 동일
    - TLS의 master_secret계산 방법은 다름

      Master_secret = PRF(pre_master_secret, "master secret",
      ClientHello.random || ServerHello.random)

    - 키 블록 재료(MAC 비밀키, 세션 암호 키, IV)의 계산은 충분한 출력이 생성될 때 까지 수행

      key_block = PRF(master_secret, "key expansion",
      SecurityParameters.server_random ||
      SecurityParameters.client_random)

  - 유연한 패딩 가능(암호 블록 길이의 배수)

전송계층 보안

- TLS (Transport Layer Security)
  - 단점
    - TCP를 사용해야 함
      - 메모리 소비
    - PKI 를 사용해야 함
      - PKI 지원하지 않는 클라이언트 존재

목 차

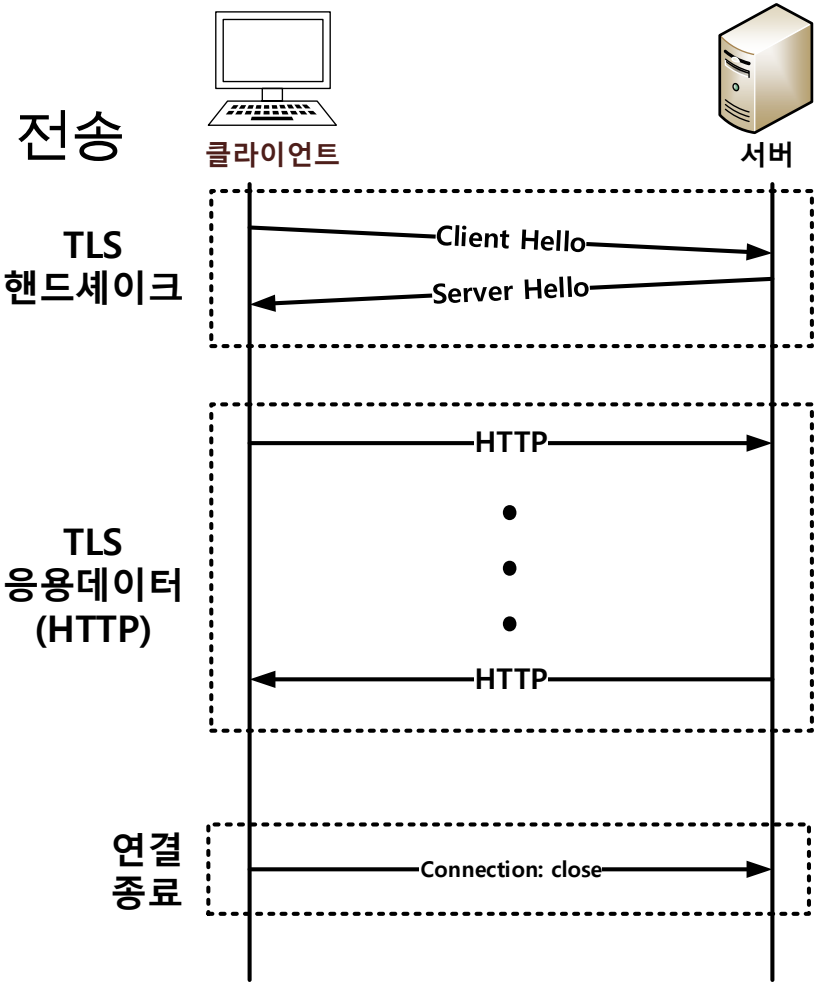
- 웹 보안
- 안전소켓 계층 보안 (SSL)
- 전송계층 보안 (TLS)
- HTTPS
- SSH

HTTPS

- HTTPS (HTTP Over TLS/SSL)
  - 개요
    - 웹 브라우저와 서버간 안전한 통신을 구현하기 위함 SSL과 HTTP의 결합
      - HTTPS 기능은 현재 모든 웹 브라우저에 내장
        - 웹 서버에 따라 사용
      - HTTPS의 포트는 443 (HTTP는 80)
      - URL이 http:// -> https:// 로 시작
  - HTTPS의 암호화 기능
    - 요청 문서 URL
    - 문서의 내용
    - 브라우저 양식 내용
    - 브라우저가 송신한 쿠키, 서버가 송신한 쿠키
    - HTTP 헤더의 내용

HTTPS

- HTTPS (HTTP Over TLS/SSL)
  - 동작
    - 연결 개시
      - TLS 핸드셰이크 완료후 HTTP첫 요청 전송
      - 모든 HTTP데이터는 TLS 응용 데이터
    - 연결 종료
      - HTTP 레코드에 Connection: close 값으로 연결 종료 표시
      - TLS수준 연결종료는 close_notify 경보 전달



HTTPS

- HTTPS (HTTP Over TLS/SSL)
  - 동작
    - 비 정상 종료
      - 하위 TCP연결이 TLS 수준연결 종료보다 먼저 발생
        - Close_notify경보와 Connection:close지시자 없이 종료 또는 종료 절차가 완전하지 않은 경우
      - 서버 프로그램 오류나 제 3자의 공격이 원인이 됨
        - TLS는 보안 경보를 통해 에러와 위험 등을 감지/대처

목 차

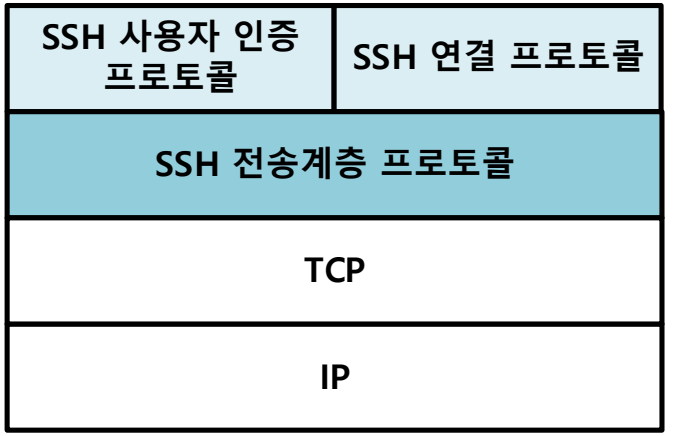
- 웹 보안
- 안전소켓 계층 보안 (SSL)
- 전송계층 보안 (TLS)
- HTTPS
- SSH (Secure SHell)

SSH

- SSH (Secure SHell)
  - 개요
    - 네트워크에서 다른 컴퓨터에 로그온, 원격 명령 등 안전한 데이터 전송을 구현한 통신 프로토콜
    - 암호화 되지 않는 기존의 Telnet등을 대체하기 위해 설계
    - 클라이언트/서버 구조의 TCP 보안 채널 (포트:22)

SSH

- SSH (Secure SHell)
  - 개요
    - 전송 계층 프로토콜 (Transport Layer Protocol)
      - 전방향 기밀(Forward Secrecy)를 만족하는 서버의 인증, 데이터 기밀성/무결성 제공 (옵션으로 압축 제공)
    - 사용자 인증 프로토콜(User Authentication Protocol)
      - 서버에게 사용자를 인증
    - 연결 프로토콜(Connection Protocol)
      - 하나의 기본 SSH 연결을 사용하여 여러 개의 논리적 통신 채널(암호화된)을 다중화

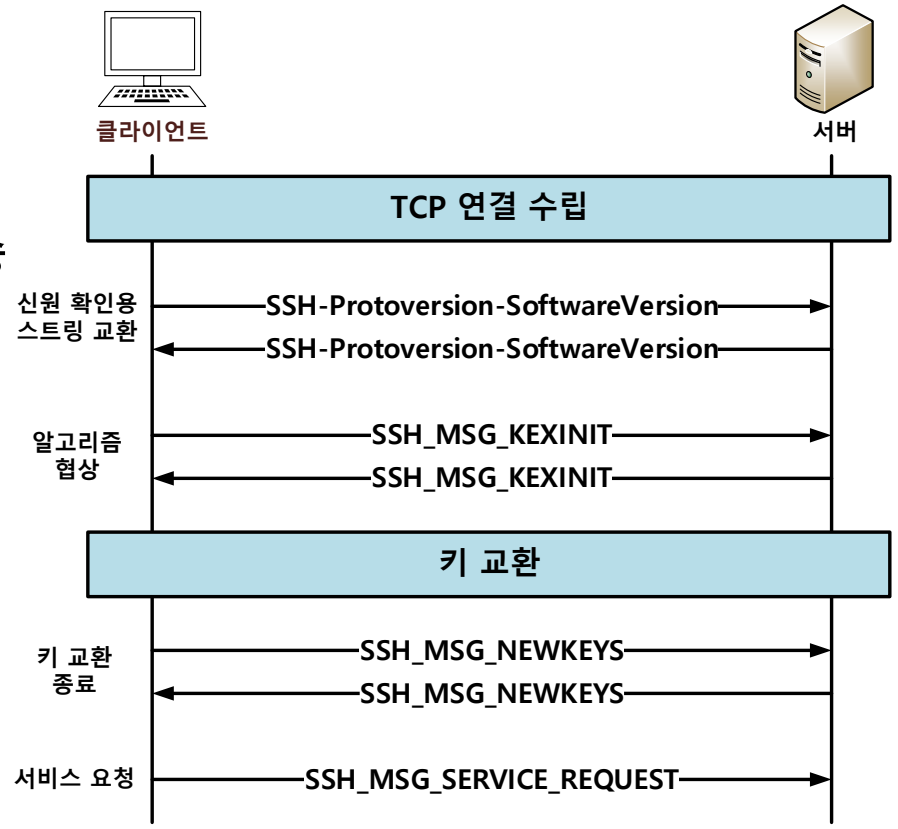


SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 서버인증과 암호화를 통해 기밀성/무결성 보장
      - 암호화에 사용할 알고리즘 협상, 키 교환, 암호화/복호화 담당
    - 서버 인증
      - 클라이언트가 각 호스트 이름과 대응되는 호스트-공개키 쌍 데이터베이스를 관리
        - 중앙 집중, 공증이 필요치 않음 <-> 키 관리에 부담
      - 호스트-키 쌍을 인증기관(CA)을 통해 인증
        - CA의 루트 키만 알고 신뢰하는 모든 CA들이 인증하는 호스트 키를 검증
        - 각 호스트 키는 인가 이전에 미리 중앙기관에 의해 적절한 인증이 필요

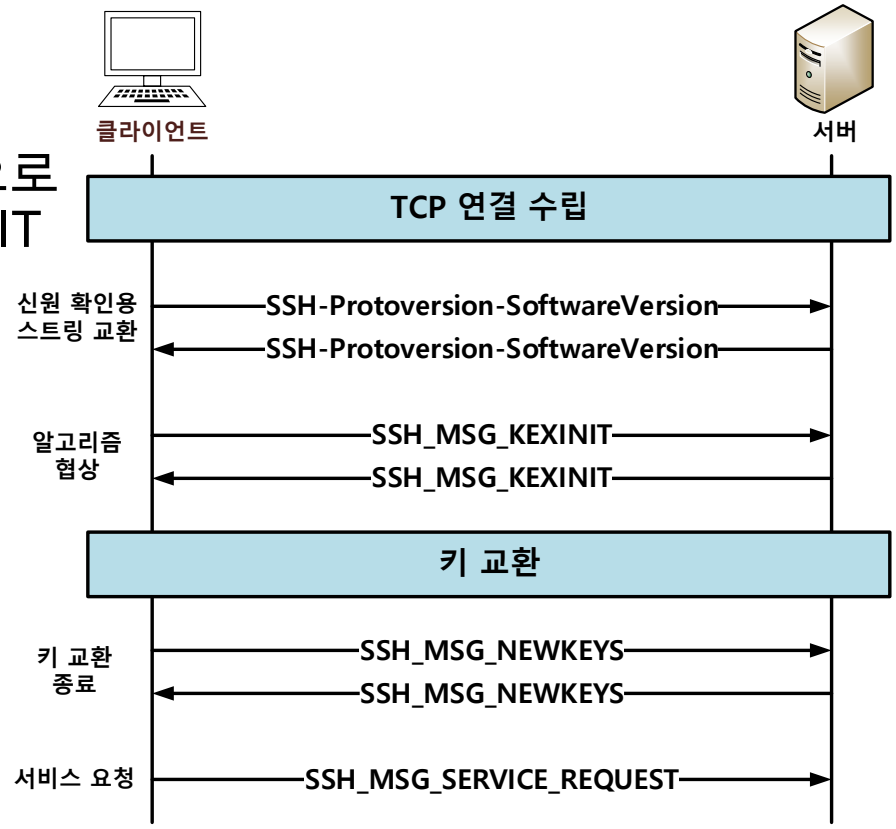
SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작 초기
      - TCP연결 수립
      - 신원 확인용 스트링 교환
        - 클라이언트/서버 식별 문자열 전송



SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작
      - 알고리즘 협상
        - 지원 가능 알고리즘을 선호도 순으로 정렬한 목록을 SSH_MSG_KEXINIT 에 포함하여 전송
          - 키 교환, 암호, MAC, 압축 목록

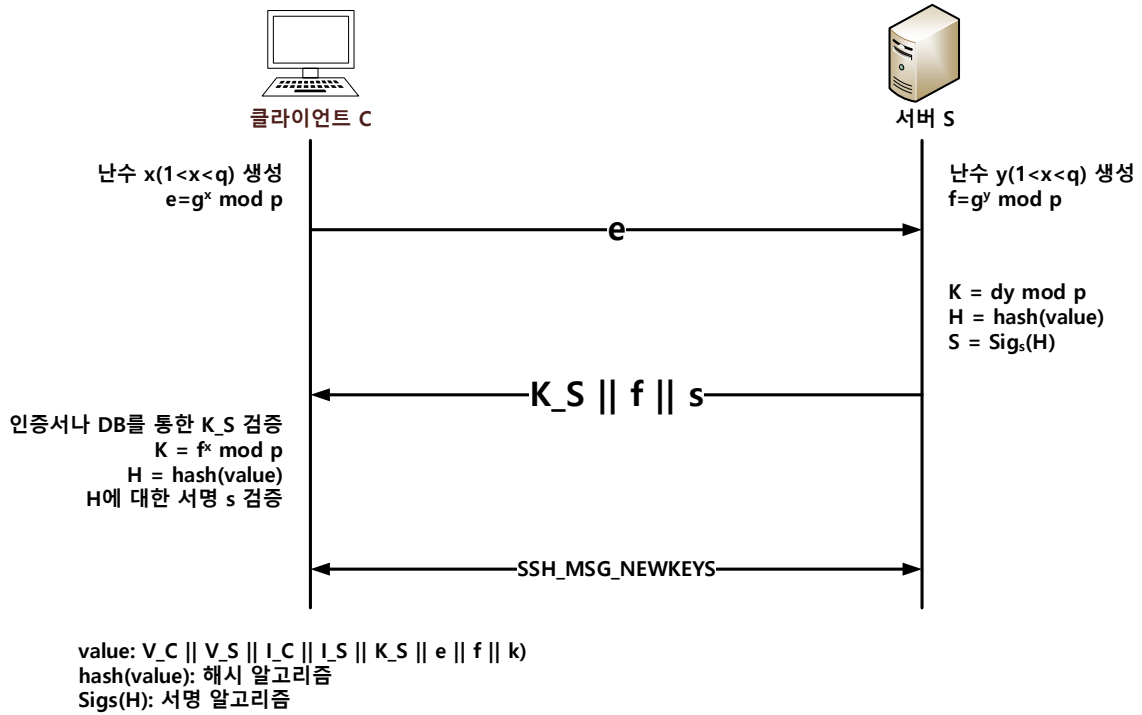


SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작
      - 알고리즘 협상
        - 암호알고리즘
          - CBC모드의 DES(3DES), AES, Blowfish, Twofish, Serpent, Arcfour, Cast 등의 알고리즘을 다양한 키 길이로 지원
        - MAC 알고리즘
          - HMAC사용을 위한 SHA-1, MD5등의 알고리즘을 다양한 키 길이로 지원
        - 압축 알고리즘
          - None: 압축 없음
          - zlib: RFC 1950과 RFC 1951에서 정의

SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작
      - 키 교환(Diffie-Hellman)
        - 키 교환후 클라이언트 서버는 K를 공유



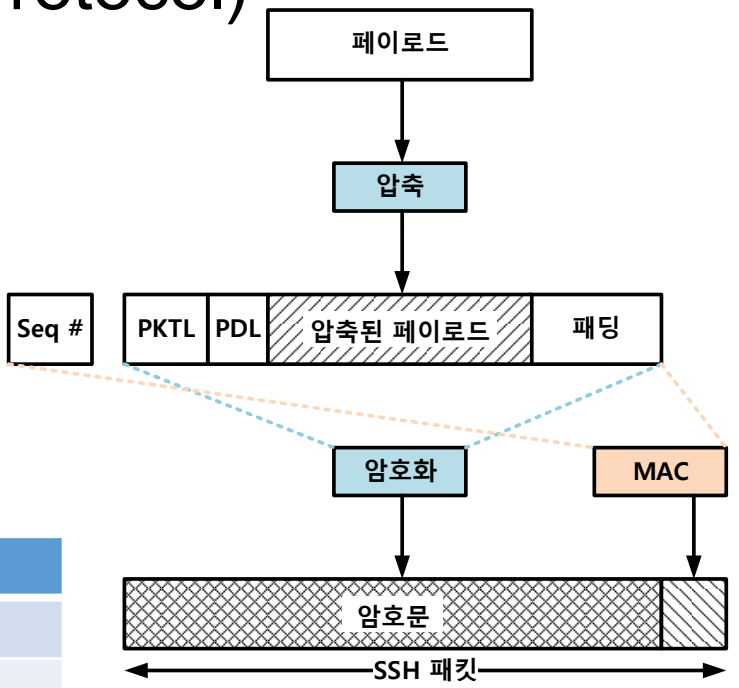
인자	의미	인자획득 시간
p	안전한 큰 소수	알고리즘 협상
g	서브 그룹의 생성자	알고리즘 협상
q	서브 그룹 위수	알고리즘 협상
V_S	서버 식별 문자열	신원확인 단계
V_C	클라이언트 식별 문자열	신원확인 단계
I_C	클라이언트SSH_MSG_KEYINIT	알고리즘 협상
I_S	서버 SSH_MSG_KEYINIT	알고리즘 협상
K_S	서버의 공개키	

SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작
      - 키 생성(Diffie-Hellman)
        - 공유된 마스터 키 K, 키교환 시 계산된 해시값 H, 세션 식별자를 이용해 암호화와 MAC에 사용할 키 생성 (키 생성후 키 교환이 없으면 세션 식별자는 H)
        - C -> V IV: HASH(K || H || “A” || session_id)
        - V -> C IV: HASH(K || H || “B” || session_id)
        - C -> V 암호화 키: HASH(K || H || “C” || session_id)
        - V -> C 암호화 키: HASH(K || H || “D” || session_id)
        - C -> V 무결성 키: HASH(K || H || “E” || session_id)
        - V -> C 무결성 키: HASH(K || H || “F” || session_id)

SSH

- SSH (Secure SHell)
  - 전송계층 프로토콜 (Transport Layer Protocol)
    - 프로토콜 동작
      - 서비스 요청
        - 프로토콜 수립 요청과 수립 이후의 패킷



필드	의미
Packet Length	pkl과 MAC를 제외한 패킷의 바이트 길이
Padding Length	랜덤 패딩 필드의 길이
Payload	패킷의 유용한 정보, 알고리즘 협상이 완료된 뒤 압축, 압축 협상이 수립되면 이후의 패킷에서 이 필드를 압축
Random Padding	암호화 알고리즘이 협상 되면 이 필드를 추가, MAC을 제외한 패킷 길이를 암호블록 크기의 배수가되게 하거나, 스트림인 경우 8파이트
MAC	메시지 인증 협상이 완료되면 이 필드에 MAC값 삽입,

SSH

- SSH (Secure SHell)
  - 사용자 인증 프로토콜 (User Authentication Protocol)
    - 클라이언트가 서버에게 인증
    - 사용자 인증 프로토콜 동작
      - 비밀번호 인증 (Password Authentication)
        - 사용자 ID-패스워드 인증
      - 공개키 인증 (Public Key Authentication)
        - 공개키 인증서 기법을 사용한 인증
      - 호스트 기반 (Host-based)
        - 호스트가 인증(공개키)되어있고 호스트에서 클라이언트를 인증해주면 서버는 호스트를 믿고 클라이언트에게 서비스 제공

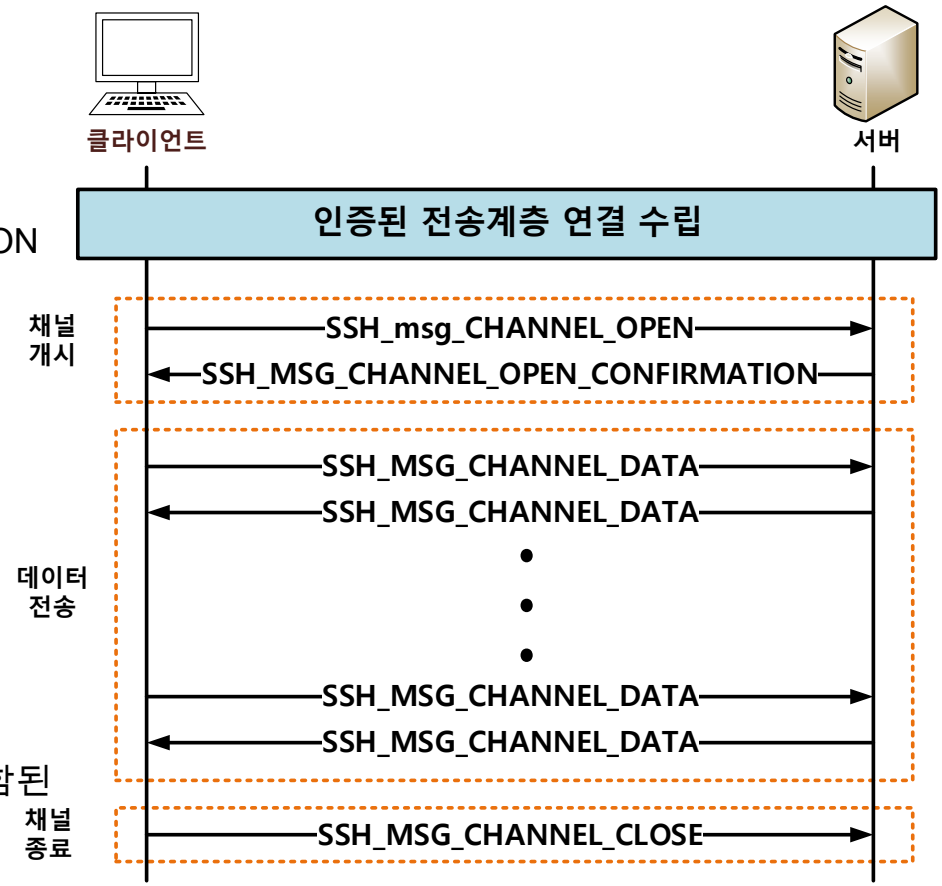
SSH

- SSH (Secure SHell)
  - 연결 프로토콜 (Connection Protocol)
    - SSH의 안전한 인증 연결이 된 전송 계층 프로토콜 상에서 수행
      - 안전한 인증 연결(터널) = 전송 계층이 수립되고 이미 사용자 인증 프로토콜이 수행된 이후
    - 안전한 터널을 다수의 논리 채널로 다중화

SSH

- SSH (Secure SHell)
  - 연결 프로토콜 (Connection Protocol)
    - 채널 메커니즘
      - 채널 개시
        - 채널에 로컬 번호를 할당 후 메시지 전송
        - 원격지에서 채널 개시가 가능하면 SSH_MSG_CHANNEL_OPEN_CONFIRMATION 메시지 반환
        - 채널 개시 실패 시 원인 코드와 SSH_MSG_CHANNEL_OPEN_FAILURE 메시지 반환
      - 데이터 전송
        - SSH_MSG_CHANNEL_DATA 메시지로 전송
        - 채널이 열려있으면 양 방향 전송임
      - 채널 종료
        - 한 쪽이 종료를 원하면 수신자 채널 정보가 포함된 SSH_MSG_CHANNEL_CLOSE 전송

메시지 유형	
Byte	SSH_MSG_CHANNEL_OPEN
String	channel type
uint32	sender channel
uint32	initial window size
uint32	maximum packet size
. . .	채널 유형에 따른 데이터



감사합니다!