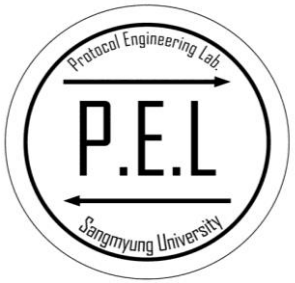


네트워크 보안 에센셜

(6장 WAP)

Sa-Rang Wi (sarang@pel.smuc.ac.kr)
Protocol **E**ngineering **L**ab., **Sangmyung** University

Content

- WAP의 개요
- WAP 프로토콜 구조
- 무선 전송 층 보안(WTLS)
- WAP 종단-대-종단 보안

WAP의 개요

- **모바일 무선 단말기를 사용할 때 생기는 문제점**
  - 장치는 배터리, 메모리, 프로세서등의 제한 가짐
  - 사용자 인터페이스는 제한적이고 화면도 작음
  - 무선 네트워크는 상대적으로 낮은 대역폭, 긴 지연시간을 가지며 유선 네트워크에 비해 예측하기 어려운 가용성과 안전성을 가짐

  - WAP에서는 위와 같은 문제점 해결 가능 !!!!!

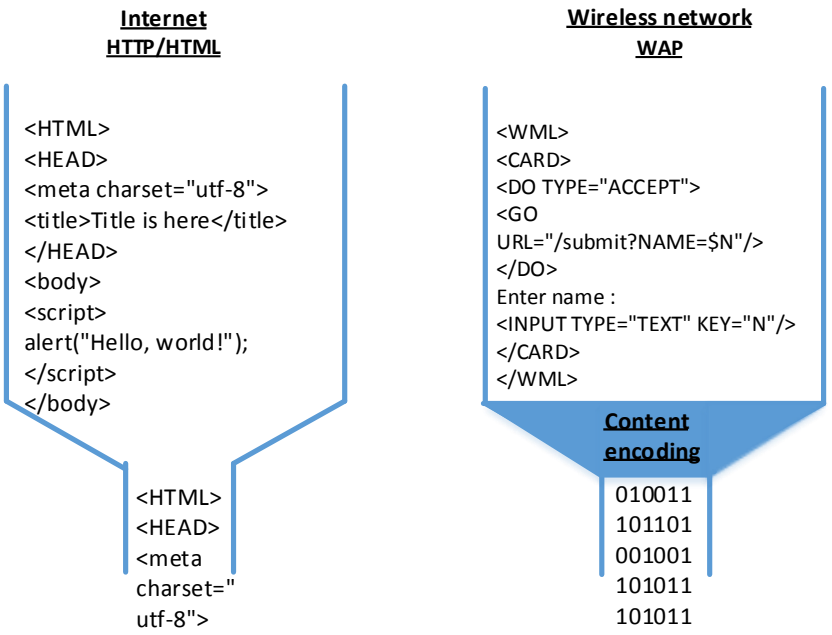
WAP 규격
- WWW 프로그래밍 모델에 기반한 프로그래밍 모델
- XML과 무선 마크업 언어(WML: Wireless Markeup Language)
- 모바일 무선 단말기에 적합한 소형 브라우저 규격
- 경량 통신 프로토콜 스택
- 무선 전화 응용을 위한 프레임워크

WAP의 개요

- **WAP이란?**
  - Wireless Application Protocol(무선응용통신)의 약자
  - 이동전화나 PDA 등 무선장치 상에서 인터넷을 이용할 수 있도록 해주는 통신 프로토콜
  - 1998년 WAP 포럼에서 개발한 통합 표준이고, 현재는 WAP 2.0 버전 사용
  - XML,HTML,HTTP와 같은 기존 인터넷 표준에 호환해서 사용가능
  - .....더 이상 사용하지 않음

WAP의 개요

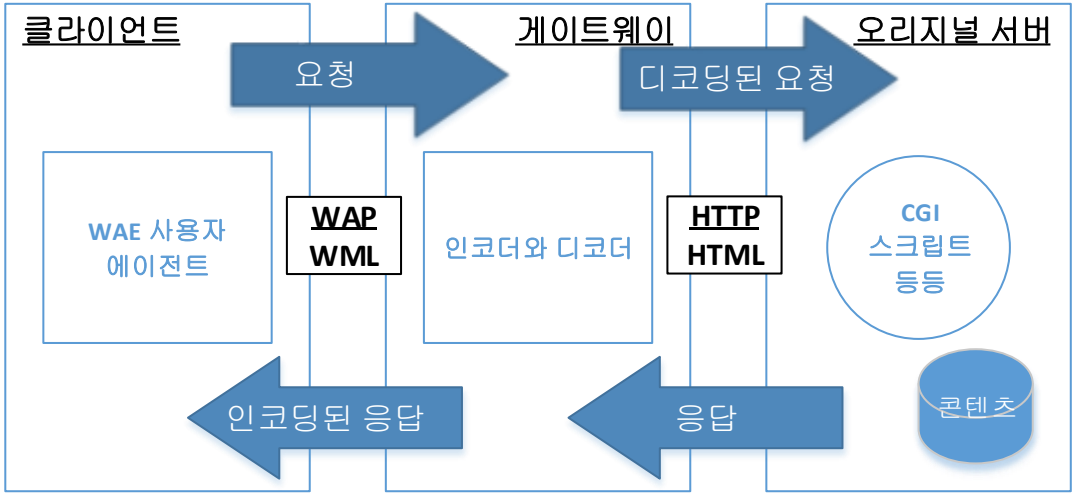
- **WML이란?**
  - Wireless Markup Language(무선마크업언어)의 약자
  - WAP 에서 작동하는 무선 프로토콜 마크업 언어
  - 작은 화면과 제한된 메모리 및 CPU, 좁은 대역폭을 가진 모바일 무선 단말기에 적합한 언어



WAP의 개요

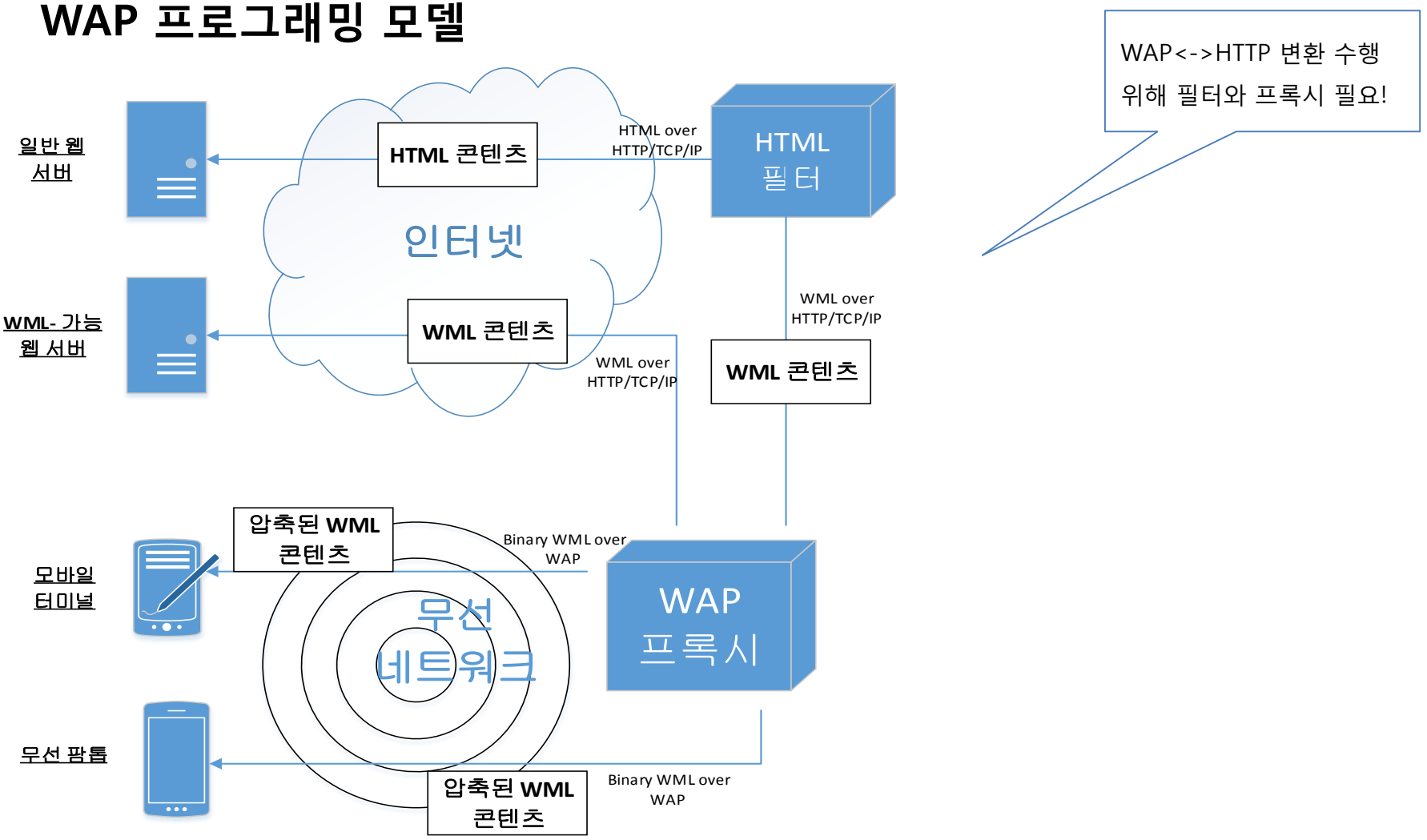
- **WAP 프로그래밍 모델**
  - 클라이언트 : 사용자의 모바일 무선 단말기
  - 게이트웨이 : WAP 과 HTTP 사이의 변환, 인코딩/디코딩 수행
  - 오리지널 서버 : 웹 서버

  - 게이트웨이와 오리지널 서버사이의 트래픽은 HTTP로 전달



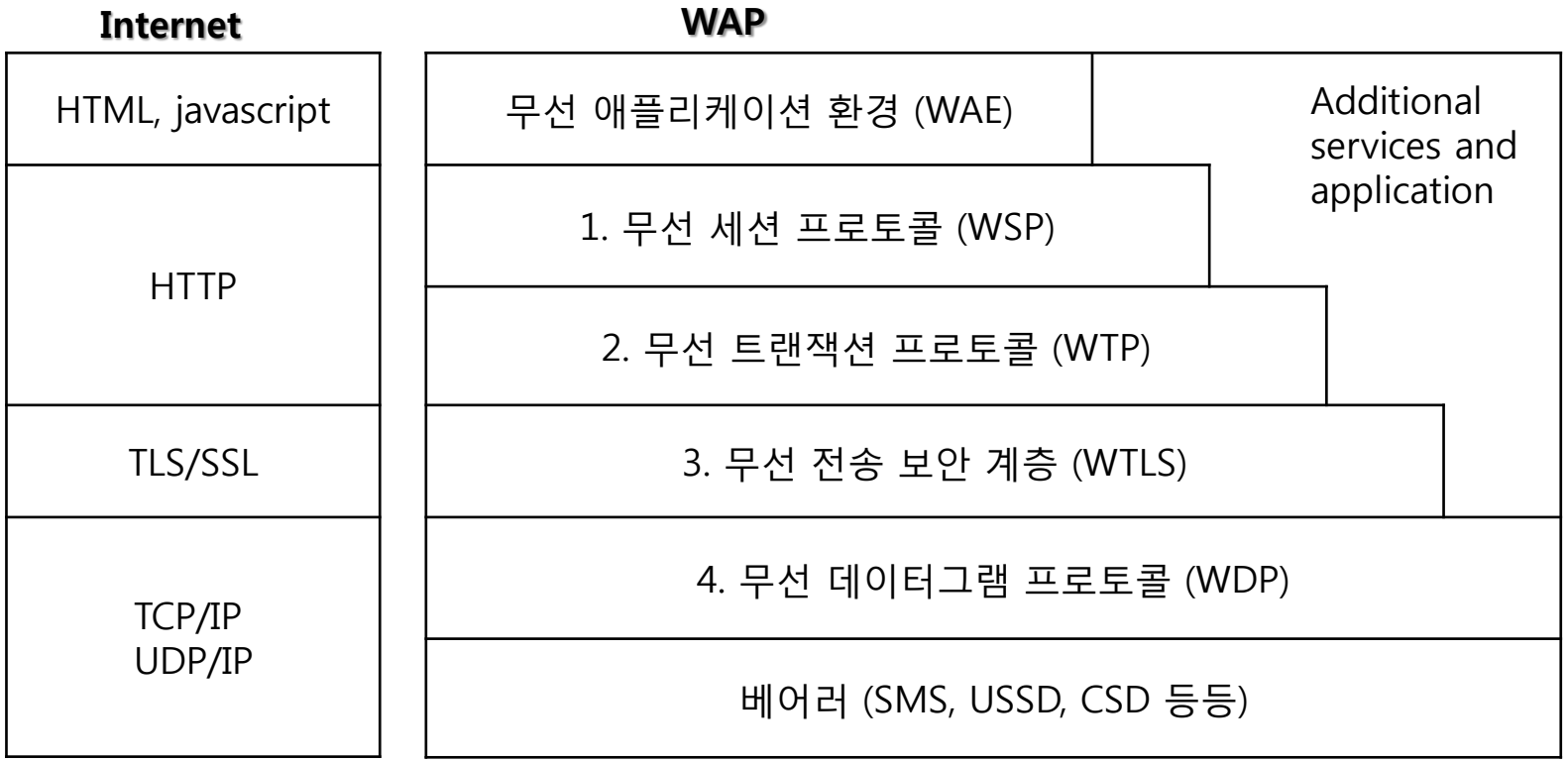
WAP의 개요

- **WAP 프로그래밍 모델**



WAP 프로토콜 구조

- **WAP 프로토콜 스택**



WAP 프로토콜 구조

- **WAP 프로토콜**
  - 무선 애플리케이션 환경(WAE)
    - ✓ Wireless Application Enviroment의 약자
    - ✓ WAP 스택의 상위계층
    - ✓ 어떻게 무선 장치를 해석하고 WML, WML script 를 표현하는지를 정의하는 마이크로 브라우저 환경
  - 무선 세션 프로토콜(WSP)
    - ✓ Wireless Session Protocol의 약자
    - ✓ 역할 : WAP 클라이언트의 WAP 브라우저가 WAP 게이트웨이에 접속할 수 있도록 새로운 접속 세션(Session)을 생성하고, 이미 접속중인 세션을 종료하는 기능 제공
    - ✓ WAE를 두 개의 세션서비스로 연결시키는 계층
      - 연결 지향 서비스 – WTP위에서 운영; 클라이언트가 인터넷 서비스 받는 동안 접속상태 유지하며 동작하는 방식
      - 비연결 서비스 – WDP위에서 운영; 컨텐츠만 전달하고 바로 접속 종료하는 방식

WAP 프로토콜 구조

- **WAP 프로토콜**
  - 무선 트랜잭션 프로토콜(WTP)
    - ✓ Wireless Transaction Protocol의 약자
    - ✓ WTP는 유선 인터넷에 없는 프로토콜로, 트랜잭션 중심 (Transaction-Oriented)의 통신 방식 지원
      - WAP 브라우저와 WAP 서버 사이에 발생하는 각각의 통신을 각각의 트랜잭션으로써 처리
      - 트랜잭션 방식의 통신은 유선에 비해서 낮은 대역폭을 가지는 무선 통신에 알맞은 통신 방식
  - 무선 전송 보안 프로토콜(WTLS)
    - ✓ Wireless Transport Layer Security의 약자
    - ✓ TLS/SLL 프로토콜 표준을 기반 보안서비스 제공
    - ✓ 뒤에 자세하게 다시 다룸

WAP 프로토콜 구조

- **WAP 프로토콜**
  - 무선 데이터그램 프로토콜(WDP)
    - ✓ Wireless Datagram Protocol의 약자
    - ✓ WAP상위프로토콜에 일관 된 서비스 제공, 사용 가능한 베어러 서비스 중 하나와 통신
    - ✓ 포트번호 주소화,분리 및 재 조립, 오류 탐지
  - 데이터 전송 방식(Bearers)
    - ✓ 모든 베어러 네트워크는 WDP 하위에 있음
    - ✓ 짧은 메시지를 보내기 위한 장치인 SMS(Short Message Service),USSD(Unstructured Supplementary Service Data)등을 포함

무선 전송 층 보안(WTLS)

- **WTLS**
  - WAP에서 안전한 통신을 위해 정의한 보안 프로토콜로, 인터넷의 TCP/IP 위의 TLS를 바탕으로 무선 환경에 최적화 된 프로토콜
  - WAP장치와 WAP 게이트웨이 간의 보안서비스를 제공
  - WAP 게이트웨이와 Web 서버 사이의 보안은 TLS

- **WTLS 의 기능**
  - 데이터 무결성 : 메시지 인증
  - 기밀성 : 암호화
  - 인증 : 디지털 인증서

무선 전송 층 보안(WTLS)

- **WTLS의 고려사항**
  - TLS 기반이지만, 무선환경으로 인하여 몇 가지 고려사항/특징 가짐
  - TLS는 TCP 위에서 동작하지만 WTLS는 WDP 위에서 동작하므로 데이터그램의 유실, 중복, 순서바뀜 등을 고려해야함
  - 통신속도의 제한이 있기 때문에 가능한 프로토콜에서 사용하는 통신데이터 최소화 해야 함.
    - ✓ 최적화된 크기의 인증서 사용; WTLS 인증서는 아래의 필드로 구성되어 있음

Certificate_version: 인증서 버전	Subject : 인증되는 공개키 주인
Signature_algorithm: 인증서 서명할때 사용하는 알고리즘	Public_key_type: 공개키 중류(알고리즘)
Issur: 인증서에 서명한 주체,보통은 특정 CA	Parameter_specifier: 파라미터
Vaild_not_before: 인증서 유효기간 시작	Public key: 인증되는 공개키
Vaild_not_after: 인증서 유효기간 종료	Signature: CA의 개인키로 서명

  - 무선 단말기의 메모리와 프로세서의 파워가 제한적이므로, 연산이 많이 소요되는 암호 알고리즘등은 적용되기 어려울 수 있음

무선 전송 층 보안(WTLS)

- **WTLS의 안전세션과 안전연결**
  - 안전 세션?
    - ✓ 하나의 클라이언트와 하나의 서버간의 연관
    - ✓ 핸드쉐이크 프로토콜에 의해 생성
    - ✓ 다수의 연결이 공유하는 암호적 보안 매개변수 정의
    - ✓ 각 연결마다 해야하는 새 보안 매개변수 협상을 피하기 위해 사용
  - 안전 연결?
    - ✓ peer to peer 관계
    - ✓ 일시적이고 세션과 연관되어있음

무선 전송 층 보안(WTLS)

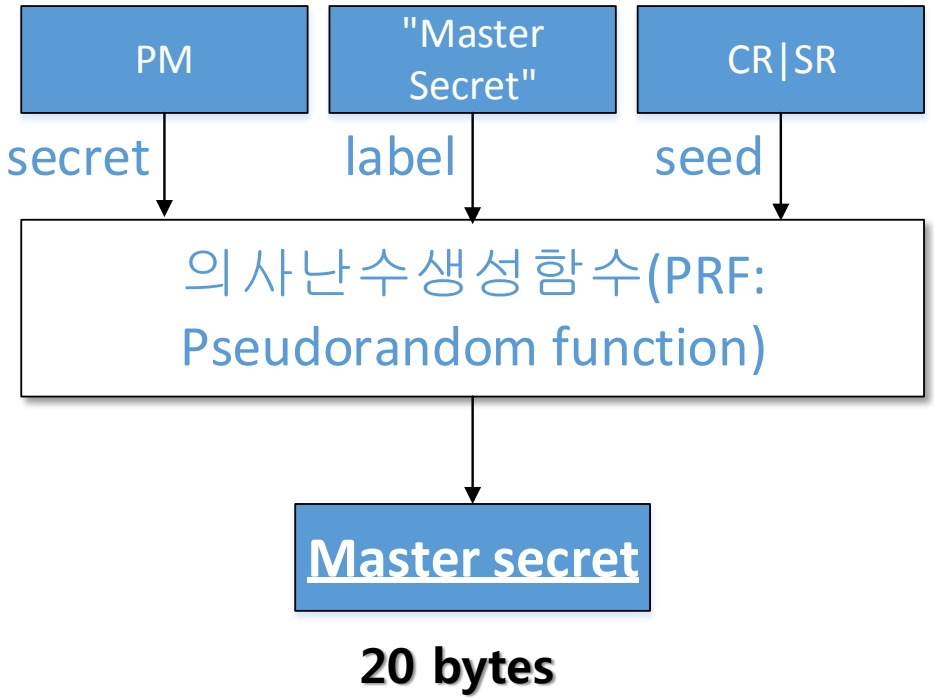
- **WTLS의 안전세션과 안전연결에서의 파라미터**
  - 세션 식별자 : 기존세션상태, 재시작 가능 세션상태 구분 하기 위한 서버가 선택한 임의의 바이트 열
  - 프로토콜 버전 : WTLS 버전번호
  - 인증서
  - 압축방법 : 암호화하기 전에 하는 압축 알고리즘
  - 암호명세 : 암호화 알고리즘 및 MAC 계산 알고리즘, 암호속성(ex, hash_size)
  - 서버/클라이언트 난수 : : 연결에 사용되는 서버/클라이언트가 선택하는 임의의 비트(16byte)
  - Master secret: 클라이언트와 서버가 공유하는 20bytes 비밀값
  - PreMaster Secret: Master secret을 만들기위한 비밀값; 핸드쉐이크 과정에서 교환한 변수를 이용

무선 전송 층 보안(WTLS)

- **WTLS의 안전세션과 안전연결에서의 파라미터**
  - 키 갱신: 연결 파라메터가 갱신되는 주기 결정
    - ✓ 키는$n = 2^{key_refresh}$ 마다 갱신되어야 함
  - 순서번호 모드 : 순서번호 보낼 때 사용하는 시스템
    - ✓ 함축 순서번호: MAC 계산시 사용되지만 실질적으로 PDU에 적용되지 않음
    - ✓ 명백한 순서번호: 순서번호가 레코드 계층의 평문(PDU)으로 전달되며 MAC입력으로 사용
    - ✓ 순서번호없음: 순서번호 없음
  - 클라이언트/서버 MAC secret
  - 클라이언트/서버 IV
  - 클라이언트/서버 Encryption Key
  - 재개 가능성 : 새로운 연결을 시작하는데 세션 이용가능한지 여부 판단 플래그

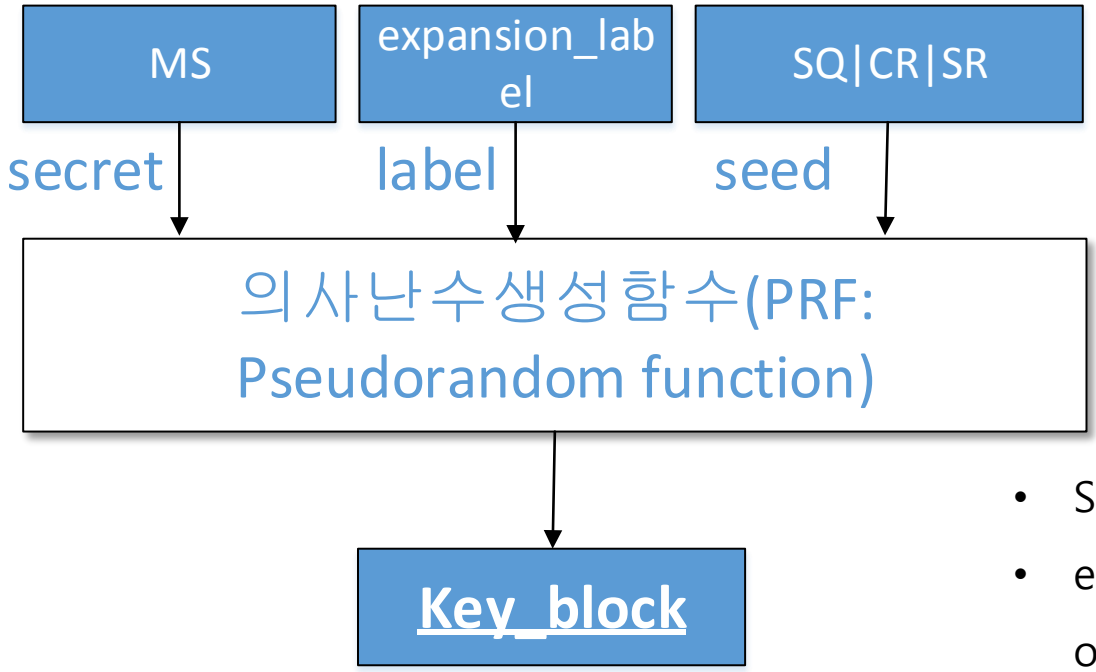
무선 전송 층 보안(WTLS)

- **Premaster secret으로 master secret 계산**
  - PM : pre-master secret
  - CR/SR : client/server 난수



무선 전송 층 보안(WTLS)

- **master secret 이용해 키 블록생성**
  - MS : master secret
  - CR/SR : client/server 난수
  - SQ : 순서번호



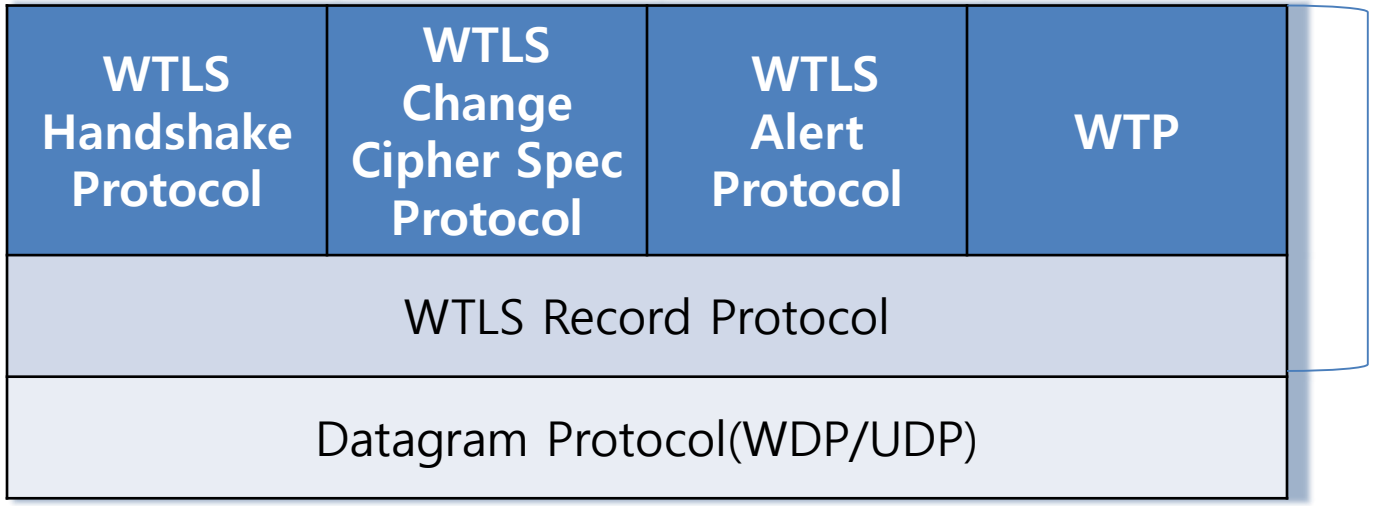
- SQ : 서버/클라이언트의 순서번호
- expansion_label : "server expansion" or "client expansion"

무선 전송 층 보안(WTLS)

- **키 블록 구조**
  - 클라이언트와 서버가 각각 두 개의 키 블록을 생성해 암호/복호화 과정에 사용
  - 서버의 키 블록
    - ✓ 서버 순서번호, "server expansion" 들어감
    - ✓ 키 블록에서 각 암호 키로 분리시킴
      - server_write_MAC_secret[SecurityParameters.mac_key_size]
      - server_write_encryption_key[SecurityParameters.key_material_length]
      - server_write_IV[SecurityParameters.IV_size]
  - 클라이언트의 키 블록
    - ✓ 클라이언트 순서번호, "client expansion" 들어감
    - ✓ 키 블록에서 각 암호 키로 분리시킴
      - client_write_MAC_secret[SecurityParameters.mac_key_size]
      - client_write_encryption_key[SecurityParameters.key_material_length]
      - client_write_IV[SecurityParameters.IV_size]

무선 전송 층 보안(WTLS)

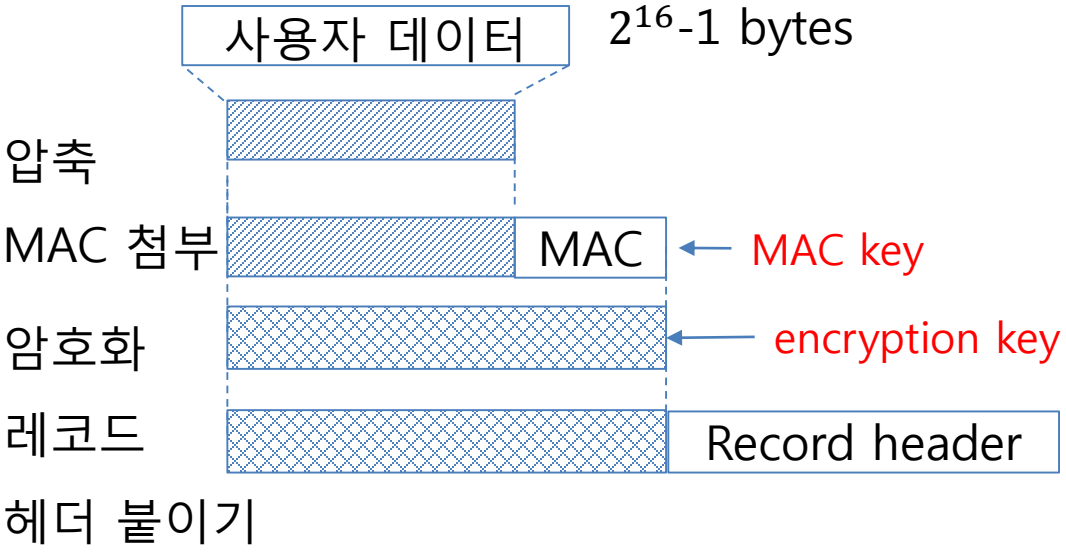
- **WTLS 구조**



- WTLS 핸드쉐이크 프로토콜
- WTLS 레코드 프로토콜
- WTLS 암호명세변경 프로토콜
- WTLS 경고 프로토콜

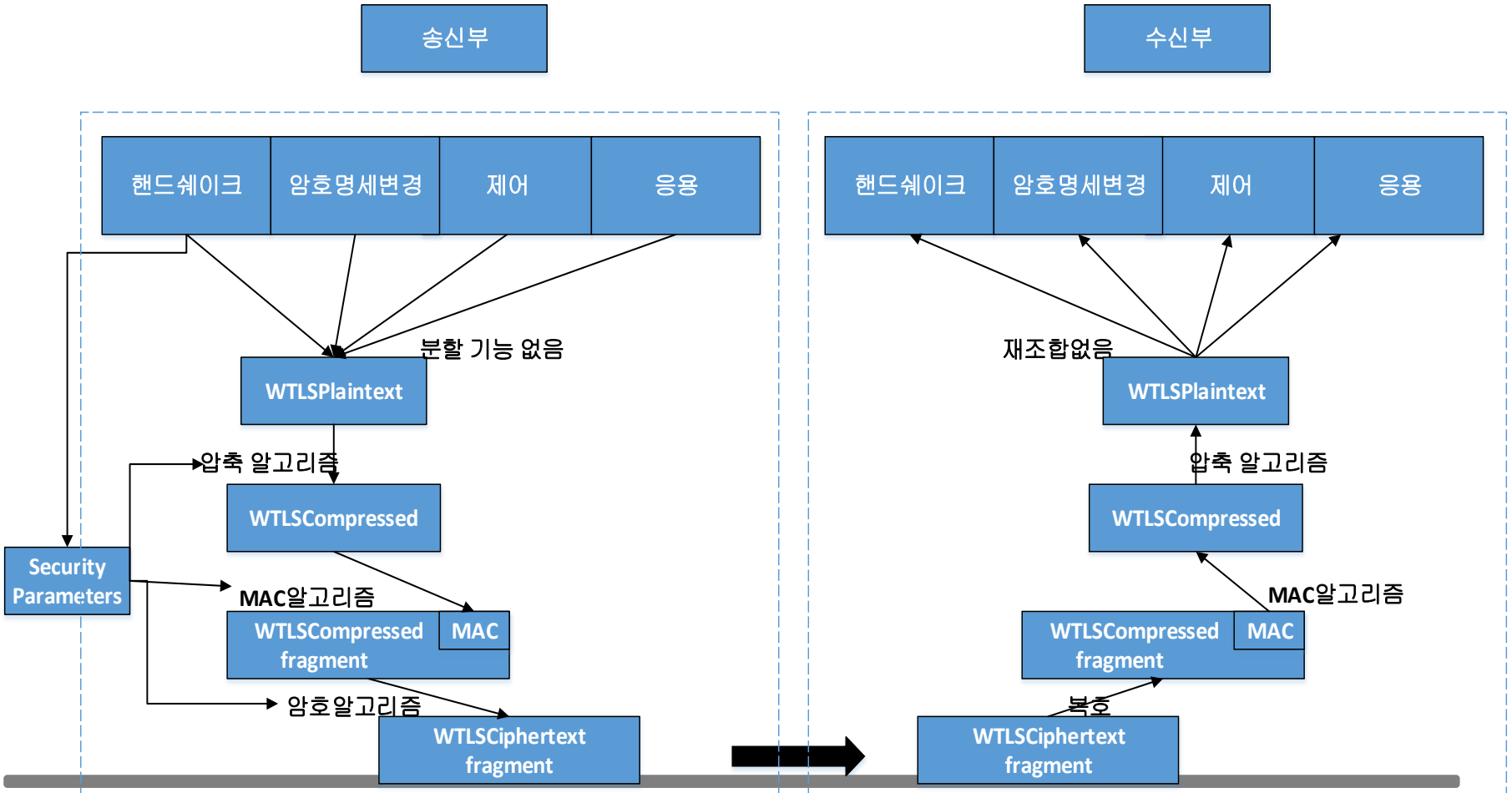
무선 전송 층 보안(WTLS)

- **WTLS 레코드 프로토콜**
  - 응용계층 데이터를 암호화된 WDP/UDP데이터그램으로 변환
  - WTLS 하위에 위치하는 WDP 계층에서 데이터에 대한 단편화가 이미 이루어지기 때문에 따로 단편화 과정 없음
  - 전반적인 동작 과정



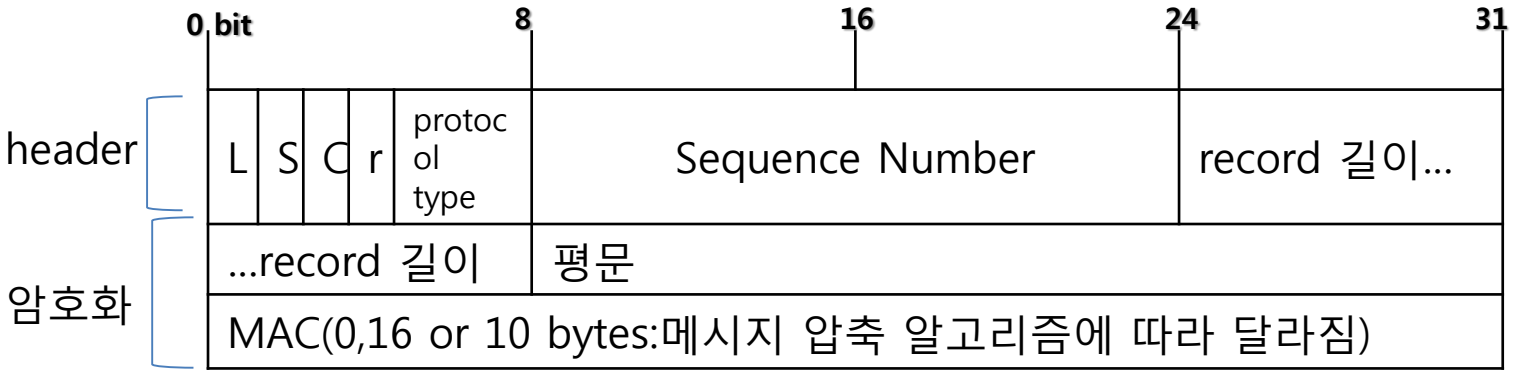
무선 전송 층 보안(WTLS)

- **WTLS 레코드 프로토콜**
  - ➢ 레코드 계층 그림



무선 전송 층 보안(WTLS)

- **WTLS 레코드 필드**
  - 레코드 종류(8비트)
    - ✓ L : 레코드 길이 필드 지시자(레코드 길이 필드의 사용 유무)(1bit)
    - ✓ S : 순서 번호 길이 필드 지시자(다음 레코드 순서 번호 필드의 사용 유무) (1bit)
    - ✓ C : 암호 스펙 지시자(압축 알고리즘의 사용 유무) (1bit)
    - ✓ r : 예약된 필드(1bit)
    - ✓ Protocol type : Change Cipher Spec(1), Alert(2), Handshake(3), Application(4) (4bit)
  - 순서번호(16bit):현재 레코드의 순서번호, 레코드 사이 구분 위함
  - 레코드 길이(16bit) : 평문 데이터의 바이트 단위



무선 전송 층 보안(WTLS)

- **WTLS 암호명세변경 프로토콜**
  - WTLS-지정 프로토콜 중 하나
  - 핸드쉐이크 프로토콜을 통해 알게 된 압축, MAC, 매개변수들을 사용가능함을 알리는 역할
  - 1byte이며, 값1을 갖는 한 개의 메시지로 구성됨

무선 전송 층 보안(WTLS)

- **WTLS 경고 프로토콜**
  - WTLS-관련 경고를 할때 사용
  - 메시지는 2byte로 구성
    - ✓ 첫번째 바이트(LEVEL): 경고(1),중요(2),심각(3)
    - ✓ 두번째 바이트(Alert): 세부적인 에러코드
      - 핸드쉐이크, 암호명세변경, 레코드 프로토콜 수행 시 발생하는 오류메시지

무선 전송 층 보안(WTLS)

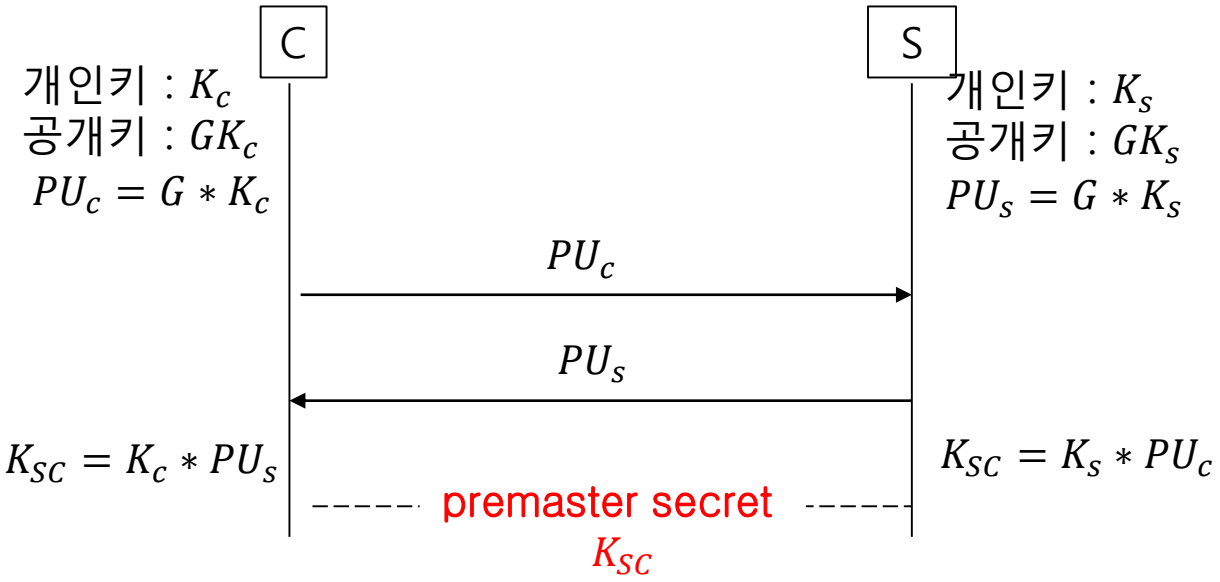
- **WTLS 핸드쉐이크 프로토콜**
  - 데이터를 전송하기 이전에 실행됨
    - ✓ 사용할 파라미터 협상
  - 클라이언트와 서버가 암호 통신에 사용할 공유키, MAC 알고리즘 결정, 인증서를 이용한 인증
  - master secret 확립을 위한 정보를 교환하기 위해 사용

무선 전송 층 보안(WTLS)

- **WTLS 핸드쉐이크 프로토콜**
  - ➢ 키교환 알고리즘
    - ✓ RSA
      - 클라이언트가 생성한 pre-master secret(20byte) 서버의 공개키로 암호화해서 서버에게 전송
    - ✓ Diffie-Hellman
      - DH 계산 수행;
      - Pre-master secret : $g^{cs} mod\ p$
    - ✓ EC Diffie-Hellman
      - ECDH 계산 수행
      - Pre-master secret :K_{SC}
  - ➢ 기존의 세션을 이용하는 경우의
    - ✓ 이전에 계산된 master secret 사용해 세션 다시 시작
    - ✓ 새롭게 교환된 서버/클라이언트 난수 값이 키 블록 계산 시 반영되어 이전 연결과 다른 키 블록 생성됨

무선 전송 층 보안(WTLS)

- **WTLS 핸드쉐이크 프로토콜**
  - ➢ 알고리즘의 종류
    - ✓ EC Diffie-Hellman
      - ✓ 타원곡선 상의 G 공유

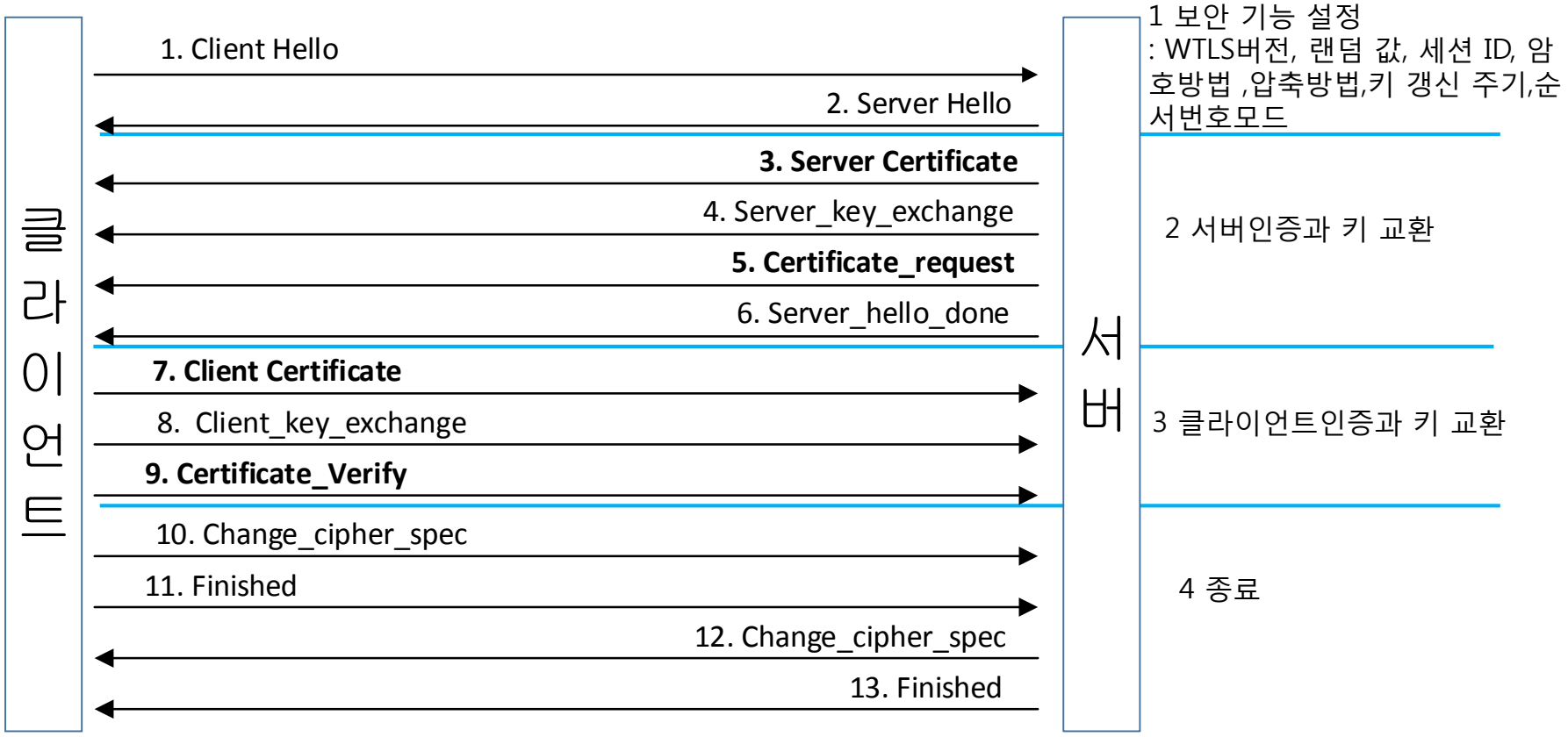


무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - 완전한 핸드쉐이크
    - ✓ 새로운 보안 세션+보안연결을 동시수행
  - 간략화된 핸드쉐이크
    - ✓ 기존의 보안세션 이용해 새로운 연결 설정
  - 최적화된 핸드쉐이크(wtls에서 추가)
    - ✓ premaster secret이 물리적 매체로 교환됨

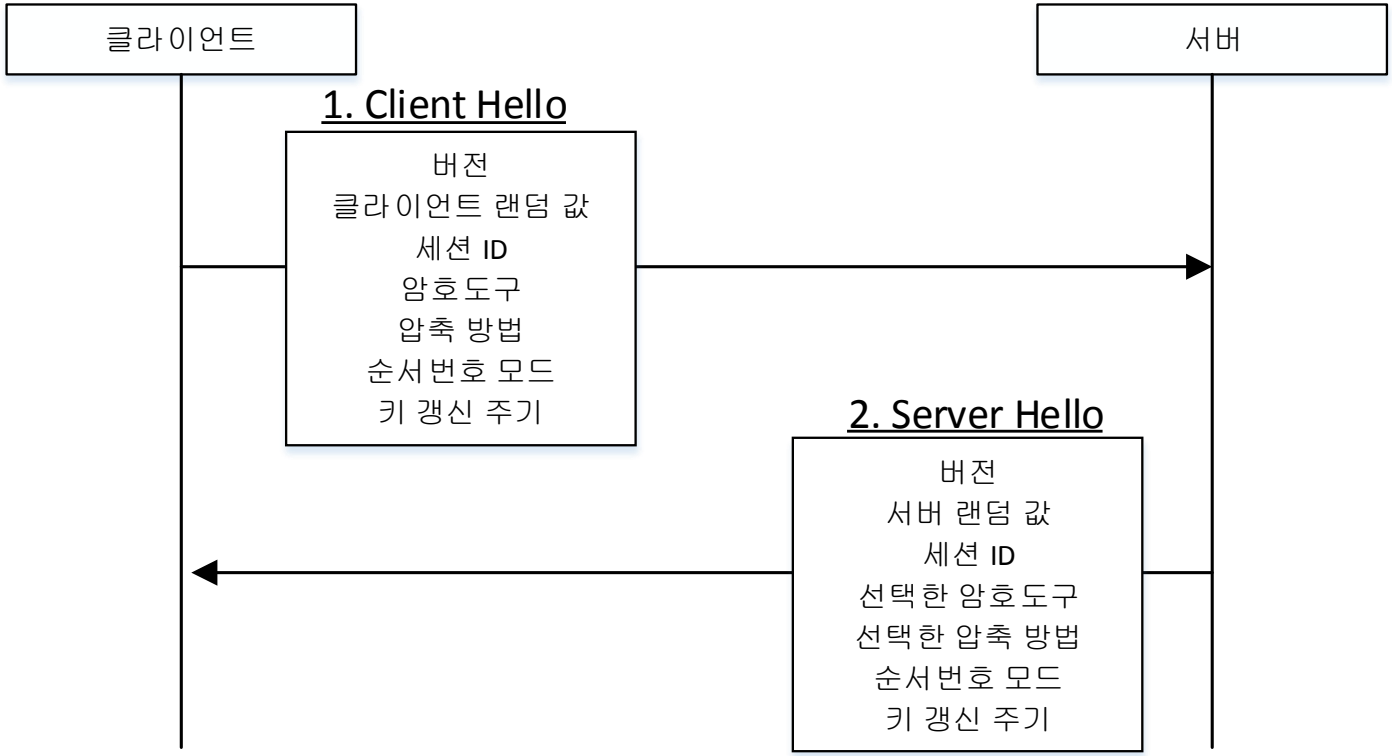
무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - 완전한 핸드쉐이크



무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 1 보안 기능 설정[그림에서 1,2]

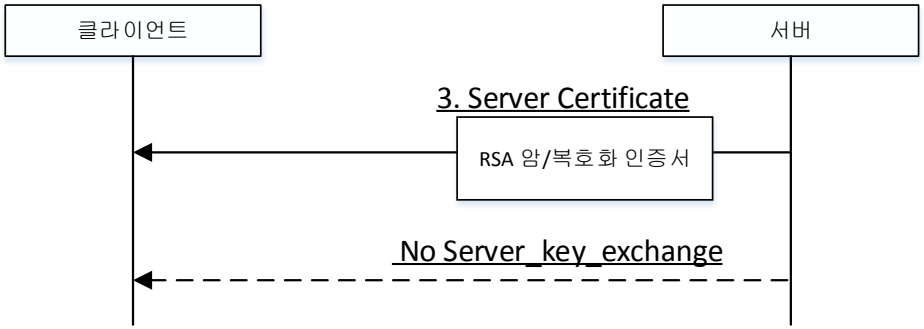


무선 전송 층 보안(WTLS)

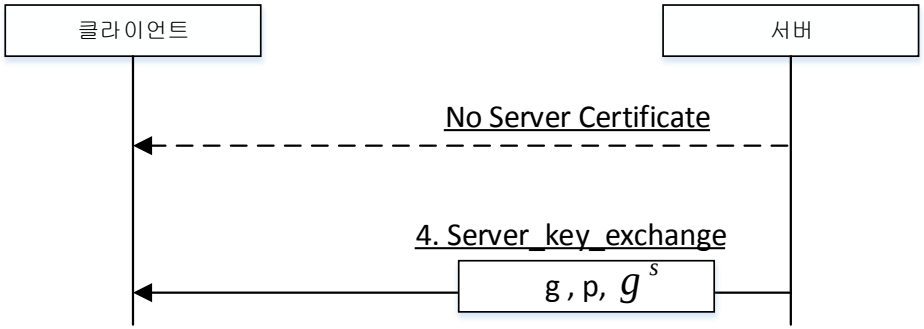
- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 2 서버인증과 키 교환[그림에서 3,4,5,6]

무선 전송 층 보안(WTLS)

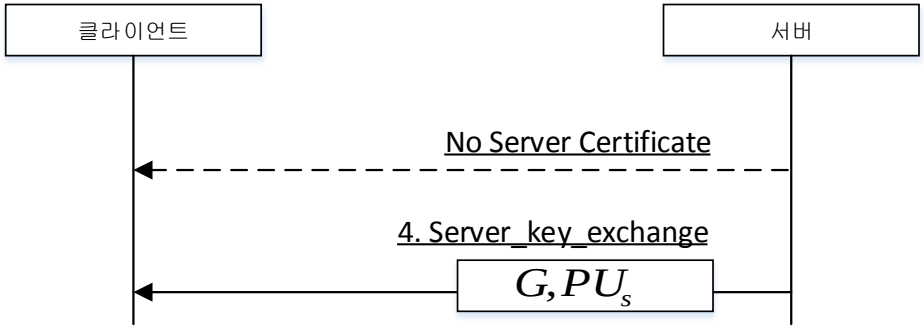
- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 2 서버인증과 키 교환[그림에서 3,4,5,6]에서의 3가지 유형



(a) RSA



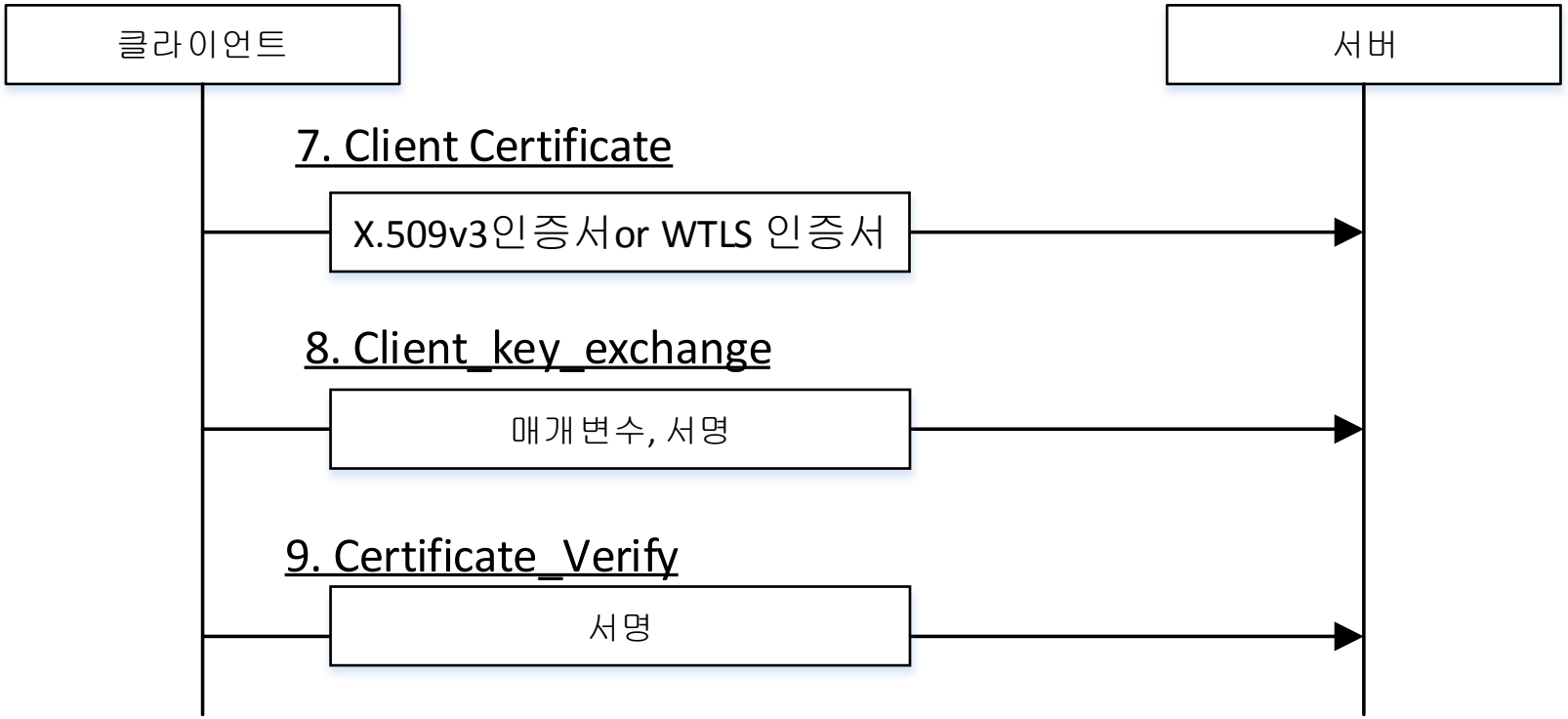
(b) Diffie-Hellman



(c) EC Diffie-Hellman

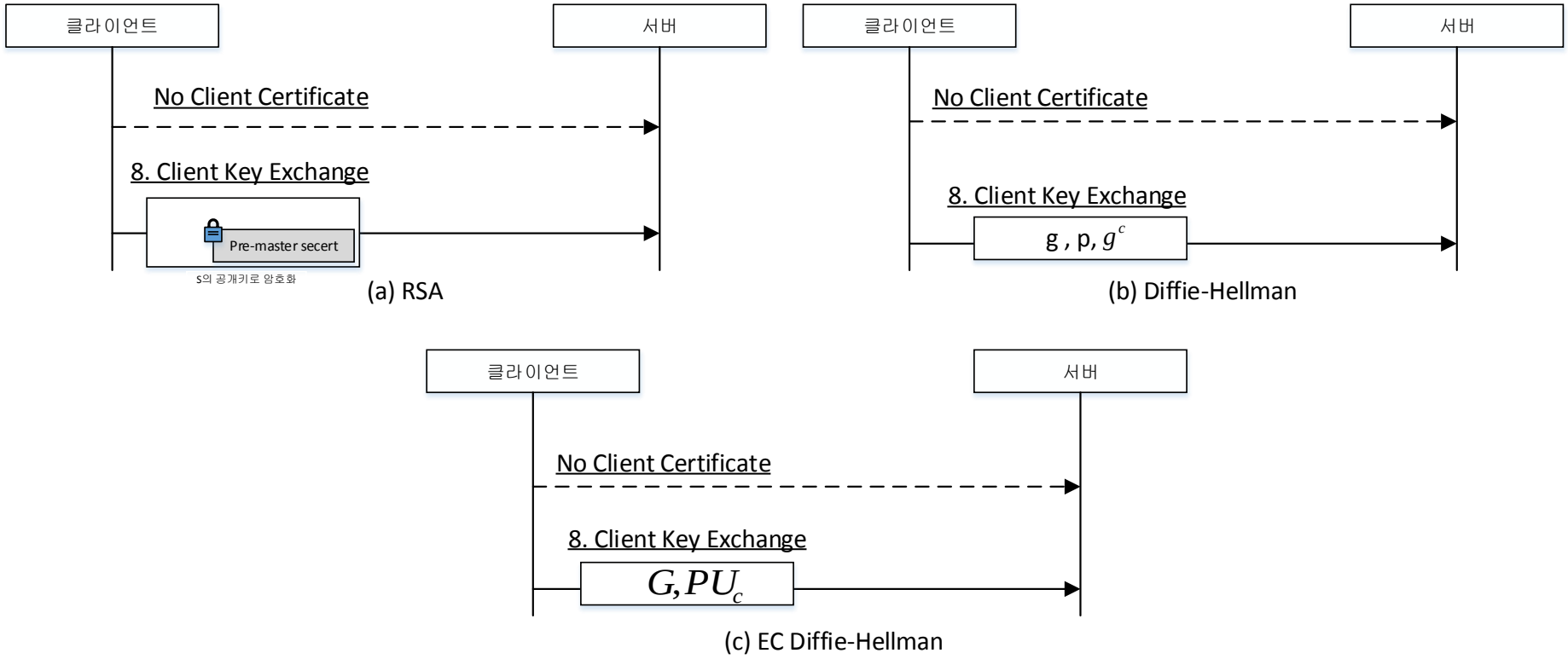
무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 3 클라이언트 인증과 키교환 [그림에서 7,8,9]



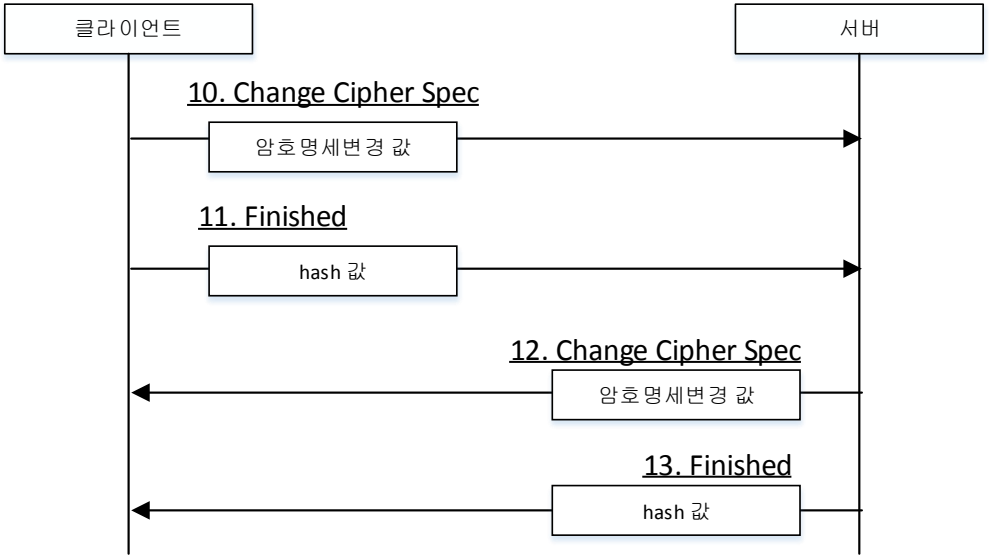
무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 3 클라이언트 인증과 키교환 [그림에서 7,8,9]에서의 3가지 유형



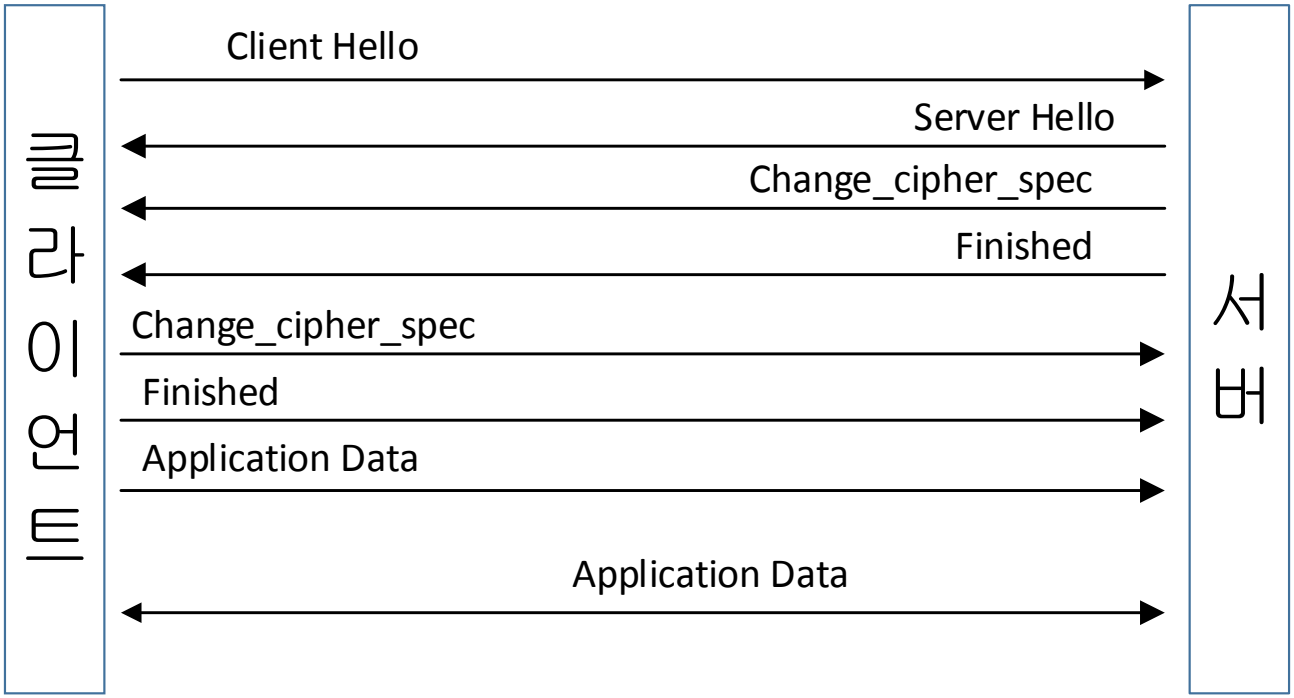
무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - ➢ 완전한 핸드쉐이크
    - ✓ 단계 4 종료[그림에서 10,11,12,13]
      - 안전한 연결 설정 종료
      - Finished 메시지는 키 교환과 인증 과정이 성공적으로 이루어짐 확인(해시값)
        -> sender, master secret,핸드쉐이킹 중에 교환된 메시지 등
        -> Finished 메시지부터 세션 키를 이용한 암호화 시작



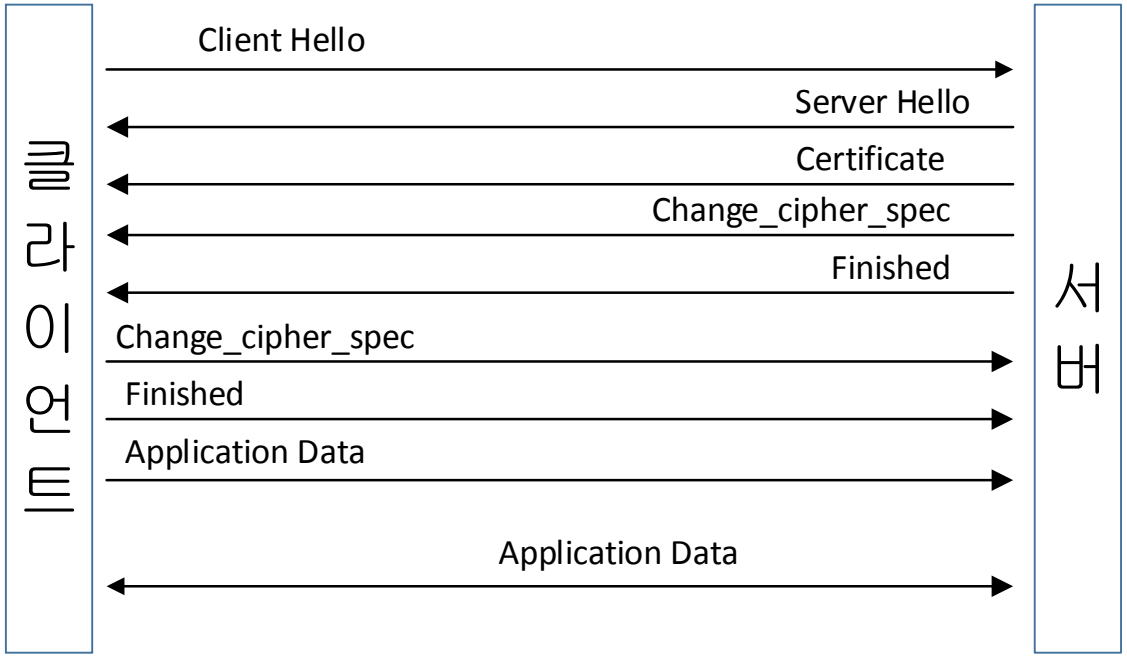
무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - 간략화된 핸드쉐이크
    - 이전에 설정된 보안 세션을 사용할 경우 사용됨
    - 다시 시작할 SessionID 정보 포함
    - 기존에 교환되었던 master secret과 현재 교환한 난수값 이용해 키블록 재생성



무선 전송 층 보안(WTLS)

- **핸드쉐이크 프로토콜의 종류**
  - ➢ 최적화된 핸드쉐이크
    - ✓ Premaster secret이 별도의 하드웨어로 수행됨(사전에 공유)
    - ✓ 서버가 클라이언트의 인증서를 다른 전송 경로를 통해 이미 얻음
    - ✓ 클/서는 sessionID=0,client_key_ids="SHARED_SECRET" (최적화 쓸거임)



WAP 종단-대-종단 보안

- **WTLS의 취약점**
  - ➢ TLS를 무선환경에 맞도록 수정한 프로토콜이기 때문에, 무선 환경의 제약된 환경으로 인해 일부 알고리즘 사용불가로 인해 안정성에 있어서 취약점으로 적용
- **WAP을 이용한 보안 영역의 문제점**

  - ➢ WAP이 사용하는 프로토콜과 인터넷 프로토콜이 서로 다르기 때문에 두 프로토콜의 연결을 위한 스택(WAP 게이트웨이)의 사용으로 인해 발생하는 취약성임
    - ✓ 게이트웨이 안에서 데이터 암호화가 되지 않은 형태로 존재
  - ➢ 그래서 종단-대-종단 보안 제공해야 함!!!!!!!!!!

WAP 종단-대-종단 보안

- **종단-대-종단 보안 방법**
  - TLS-기반 보안
    - ✓ 두 종단 사이에 안전한 TLS세션 설치
    - ✓ WAP 게이트웨이는 TCP 계층의 게이트웨이로 동작
      - 게이트웨이 통과하는 동안 TCP 사용자 데이터필드는 암호화 되어있기 때문에 종단대종단 보안 유지 가능
  - IPsec-기반 보안
    - ✓ 클라이언트나 게이트웨이에서 IPsec 사용하면 모든 과정에서 데이터가 암호화되기 때문에 종단대종단 보안 유지 가능

감사합니다~