15장. 네트워크 보안

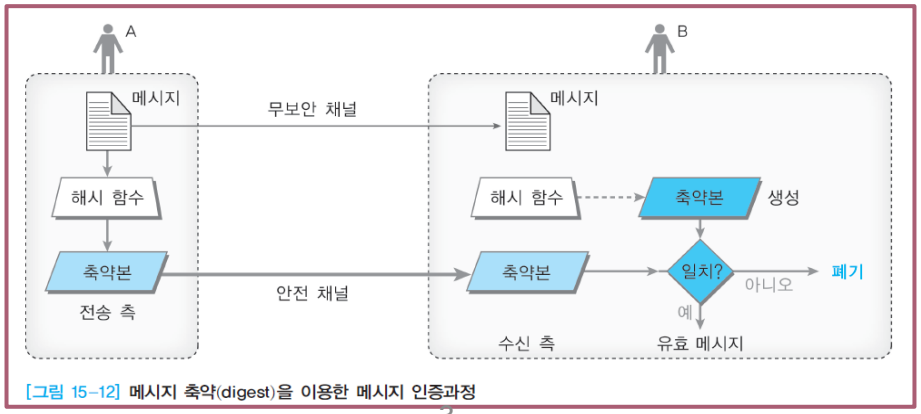
15-3 메시지 인증과 디지털 서명

메시지 인증 (1)

- 문서의 무결성은 문서의 내용을 함부로 수정할 수 없도록 하는 것과 관련됨
- 메시지의 인증이란?
  - 무결성을 유지하는 것과 관련 → 원래의 내용이 변경되지 않은 진본 상태임을 입증
- 메시지 인증 기법
  - 암호 해시 함수(cryptographic hash function)은 무결성 유지 문제의 좋은 해결 방안
  - 메시지 축약을 사용하여 메시지 무결성을 확인하는 과정

메시지 인증 (2)

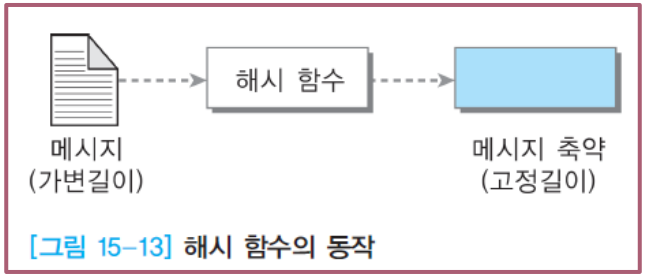
- 메시지 인증 기법 (계속)
  - ❶ 암호 해시 함수를 이용하여 메시지 축약(digest) 생성
    - 메시지의 지문(fingerprint)과도 같은 compressed image 생성
  - ❷ 이렇게 만들어진 메시지 digest는 보안 채널을 통해 별도로 전송
  - ❸ 수신 측에서 메시지와 메시지 축약을 받으면 암호 해시 함수를 이용하여 다시 새로운 메시지 축약을 생성
  - ❹ 새로 생성된 축약과 이전의 메시지의 축약과 비교
    - 두 개가 일치하면 원래의 메시지가 수정되지 않은 진본 메시지임을 확인



[그림 15-12] 메시지 축약(digest)을 이용한 메시지 인증과정

해시함수와 해시 알고리즘 (1)

- 해시 함수(hash function)



[그림 15-13] 해시 함수의 동작

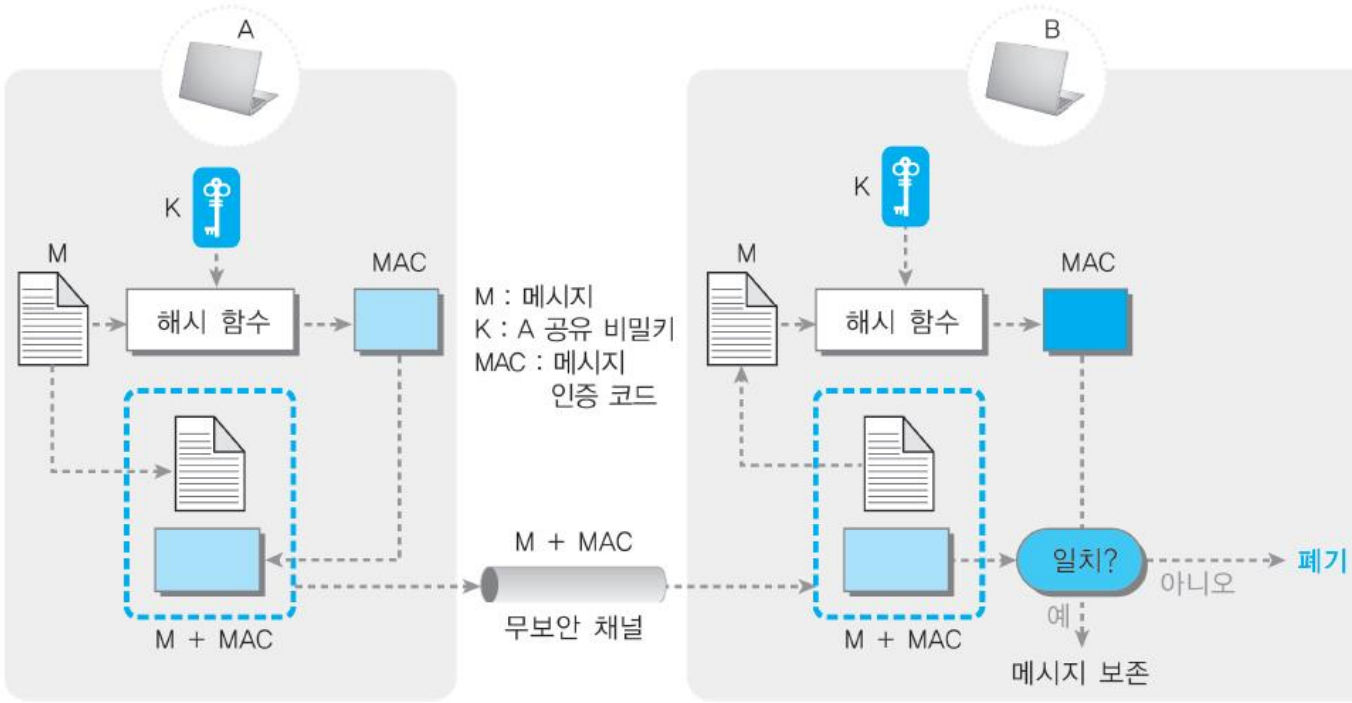
  - 가변적 길이를 갖는 메시지를 입력값으로 하여 고정된 길이의 메시지 축약본을 생성하는 함수
  - 해시 함수를 만드는 최적의 방법 → 반복법 이용함
  - 가변적 길이의 메시지 입력값을 갖는 해시 함수를 사용하는 대신 → 고정된 길이의 입력값을 갖는 해시함수를 만들어냄
  - 원하는 길이의 고정된 입력값이 될 때까지 → 해시 함수를 반복적으로 사용 → 축약본 생성
  - 생성된 고정된 길이의 입력 함수 → 압축 함수라고 함
    - n-비트문 자열을 압축하여 m-비트문자 열 생성 → '반복 암호화 해시 함수(iterated cryptographic hash function)' 기법

해시함수와 해시 알고리즘 (2)

- 해시 알고리즘의 예
  - MD2 (Message Digest 2), MD4, MD5 등 해시 알고리즘
    - 론 리베스트(Ron Rivest)에 의해 설계
    - MD5 → 128비트 축약본을 만들어내도록 설계됨
  - NIST에서 개발한 표준 → SHA(Secure Hash Algorithm)가 있음
    - SHA-1인 경우 → 160-비트 축약본 생성

MAC을 이용한 인증 기법 (1)

- 작성된 메시지가 작성자에 의한 진본이었나 하는 '인증' 문제
- 메시지 인증 코드(MAC)를 사용하는 방법



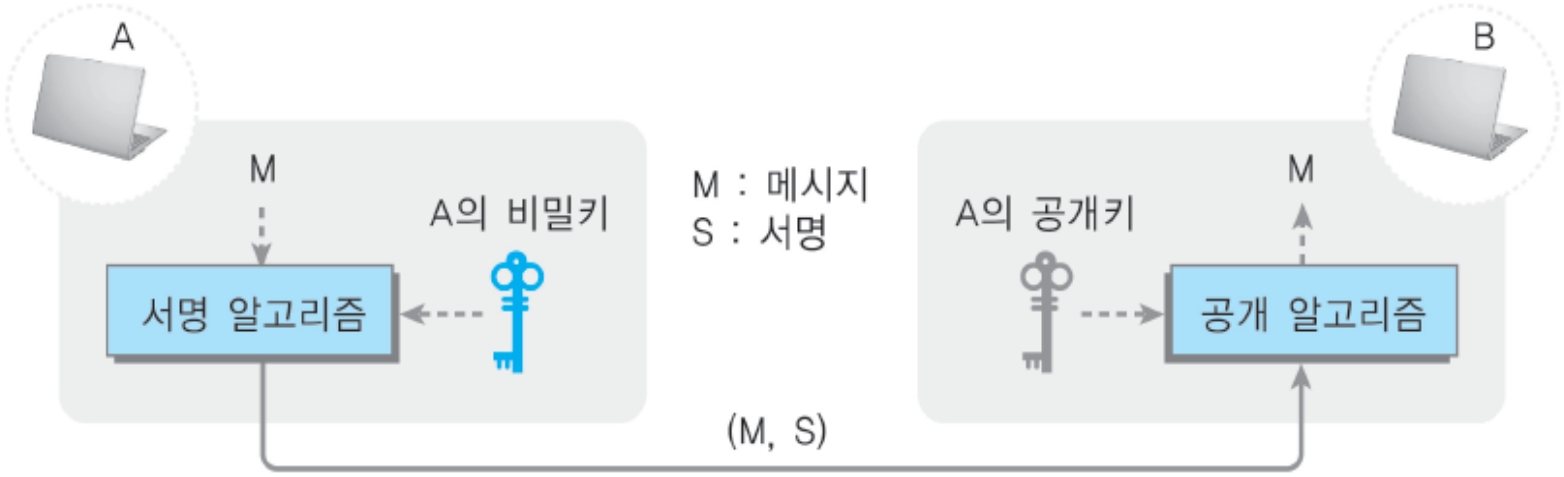
[그림 15-14] 메시지 인증 코드를 사용한 인증 기법

MAC을 이용한 인증 기법 (2)

- 메시지 인증 코드(MAC)를 사용하는 방법 (계속)
  - A는 해시함수를 사용하여 메시지 인증 코드(MAC)를 만든 다음, 메시지와 MAC을 함께 B에게 전송
  - 이것을 수신한 B는 메시지와 MAC을 분리하고, 메시지와 비밀키를 이용하여 새로운 MAC 생성
  - 두 개의 MAC을 비교
    - 서로 일치되면 수신된 메시지가 인증/ 동시에 전송 중에 수정되지 않았다는 무결성이 확인됨

디지털 서명 (1)

- digital signature
- 디지털 서명 기법



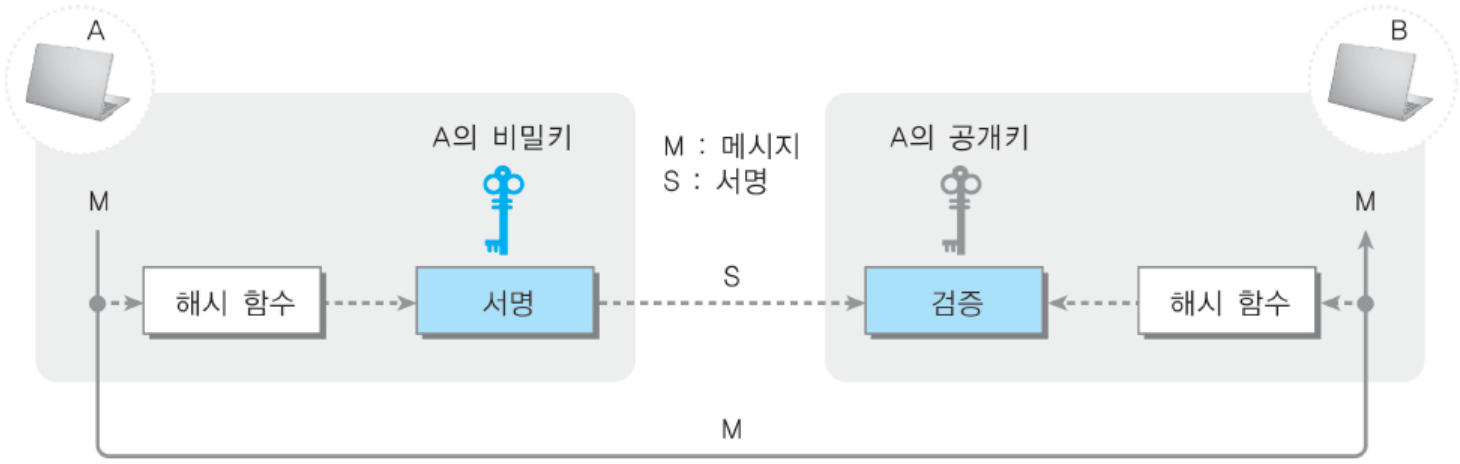
[그림 15-15] 디지털 서명 과정

디지털 서명 (2)

- 디지털 서명 기법 (계속)
  - MAC은 메시지 축약을 보호하기 위해 비밀키를 사용하는 반면, 디지털 서명에서는 개인키와 공개키 둘 다 사용
  - 전송 측은 서명 알고리즘(signing algorithm)과 개인키를 이용하여 메시지에 대한 서명
  - 메시지와 서명을 함께 전송
  - 수신 측
    - 전송 측의 공개키를 이용하여 검증 알고리즘 사용
      - 그 결과가 참(true)이면 메시지는 수락(accepted)
      - 그렇지 않으면 거절(rejected)됨

메시지 축약 서명 (1)

- 비대칭키 암호방식은 긴 메시지를 다루기에 부적합
  - 디지털 서명에서 비대칭키를 사용한다는 점을 고려하면 긴 메시지인 경우에 적합하지 않게 됨
- 이에 대한 해결책
  - 메시지의 축약에 디지털 서명을 하는 방법 사용



[그림 15-16] 메시지 축약 서명

메시지 축약 서명 (2)

- 전송 측
  - 메시지의 축약을 만들고 이것에 대하여 디지털 서명을 하여 전송
- 수신 측
  - 수신된 메시지와 해시 함수를 이용하여 축약본을 만든 다음 → 전송 측의 공개키를 이용하여 검증 과정