제9장 전자서명

메시지 인증 코드(Message Authentication Code)

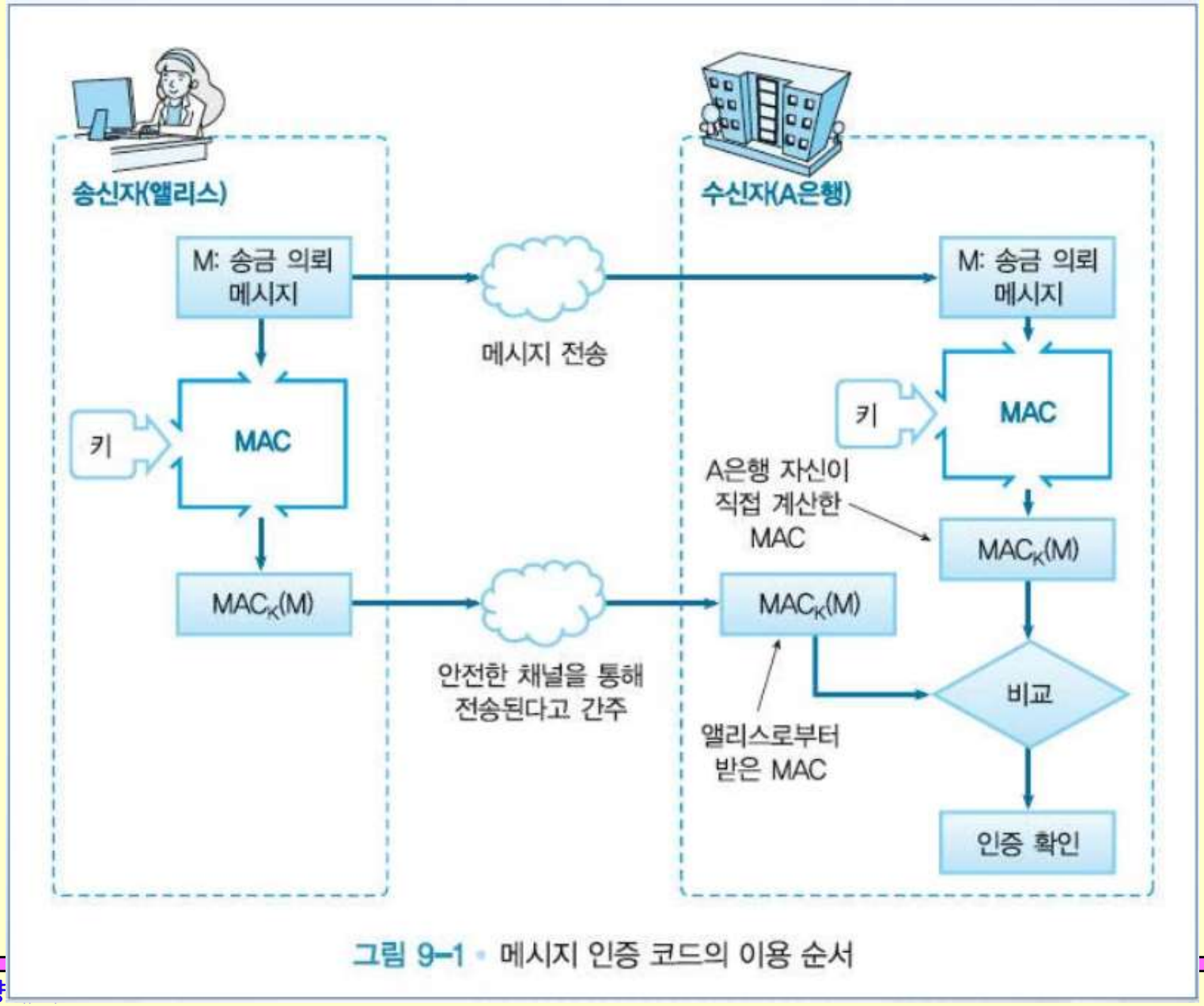


그림 9-1 · 메시지 인증 코드의 이용 순서

MAC로 해결할 수 없는 문제

- 제3자에 대한 증명
  - Alice로부터 메시지를 받은 Bob이 "이 메시지는 Alice가 보낸 것이다"라는 것을 제3자인 검증자 Victor에게 증명할 수 없음
  - 이유 :
    - 일단 키를 Victor에게 알려줘야 함
    - Alice와 Bob 모두가 키를 가지고 있으므로 둘 중 누가 작성했는지 알 수 없음
- 부인방지
  - Bob이 MAC 값이 첨부된 메시지를 받았고, 「이 메시지는 Alice로부터 온 것이다」라는 걸 확실히 알 수 있다.
  - 하지만 Alice가 전송 자체를 부정할 경우 제3자에게 이 사실을 증명할 수 없다.
  - Alice의 송신자체에 대한 부정을 부인(repudiation)이라고 한다.
  - 메시지 인증 코드로는 부인 방지(nonrepudiation)를 할 방법이 없다.
- 메시지 인증 코드의 한계
  - 메시지 인증 코드를 사용하면 메시지의 변경과 거짓 행세를 검출 가능
  - 제 3자에게 확인 시킬수 없음
  - 메시지 인증 코드는 부인 방지에는 도움이 되지 않음

❑ 메시지 인증 : 통신 쌍방 상호간의 정보유통에 대하여 제 3자로부터의 불법 수정을 보호

❖ 통신 주체 쌍방이 공유하고 있는 키를 사용하여 어느 한쪽이 메시지를 위조하고 인증 코드를 부가할 수 있다.

❖ 통신 쌍방 상호간의 당사자간 분쟁은 해결 불가

❑ 사례

❖ 갑순이는 갑돌이가 보내지 않은 메시지를 만들어 인증코드를 붙이고 갑돌이에게서 수신했다고 주장한다.

❖ 갑돌이는 자신이 보내고서도 갑순이가 거짓으로 만든 메시지라고 주장할 수 있다.

디지털 서명(Digital Signatures)

- ❑ 배경
  1) 종이 문서 사회에서 정보화 사회로의 진전으로 다양한 서비스 요구
  2) 데이터 무결성 및 사용자 인증 서비스가 필수적
- ❑ 정의
  - ❖ 전자적 문서에 서명하고, 그 문서에 대한 서명자의 유일한 신원 증명과 서명문서의 일체 내용을 검증하기 위한 행위
- ❑ 목적
  - ❖ 신뢰성 확보 (내용의 위·변조, 신분 확인, 부인봉쇄에 사용)
- ❑ 전자 서명의 속성
  - ❖ 서명의 서명자와 날짜, 시간을 확인할 수 있어야 함
  - ❖ 서명할 때의 내용을 인증할 수 있어야 함
  - ❖ 서명은 분쟁을 해결하기 위해서, 제 3자에 의해서 확인할 수 있어야함

적용 예

❖ 전자식 자금 전달의 경우

➢ 수신자 : 1. 전달된 자금의 양을 증가

2. 송신자로부터 해당 금액이 왔다고 주장

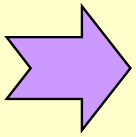
❖ 주식 매매 요청의 경우

➢ 송신자 : 1. 단말기를 통해 주식 매매 요청

2. 주식 값이 하락

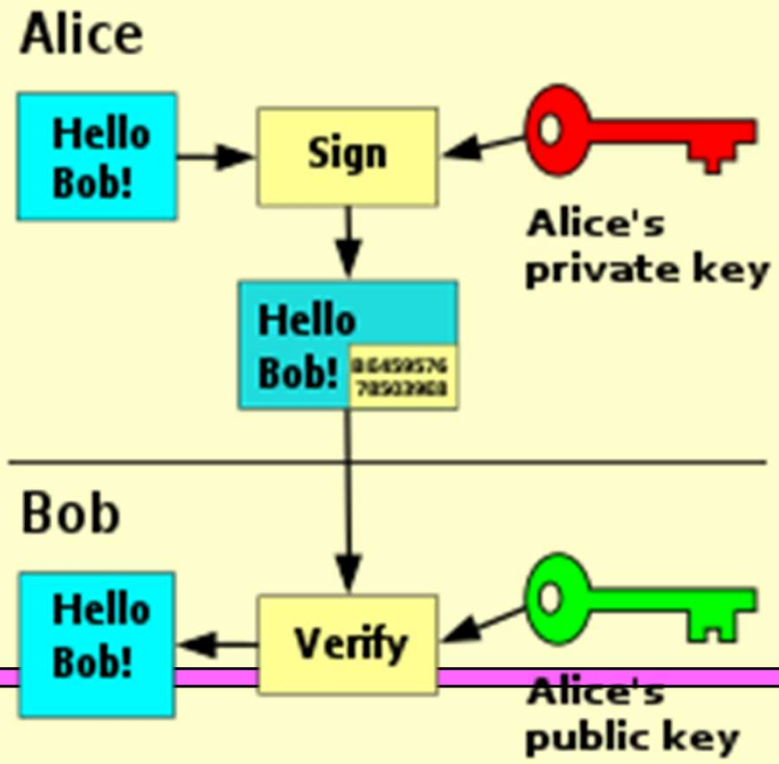
3. 자신이 요청을 한 적이 없다고 주장

송신자와 수신자의 완벽한 신뢰가 없는 상황에서 인증 이상의 어떤 것이 필요



디지털 서명

- 디지털 서명은 3개의 알고리즘으로 구성
  - ❖ 하나는 공개 키 쌍을 생성하는 키 생성 알고리즘
  - ❖ 두 번째는 이용자의 개인 키를 사용하여 서명(전자서명) 생성 알고리즘
  - ❖ 그리고 그것과 이용자의 공개 키를 사용하여 서명 검증 알고리즘
  - ❖ 서명 생성 프로세스에서 생성된 데이터를 '디지털 서명'



일반 서명의 특징

- 손으로 쓴 서명
- 해당 서명의 저자, 날짜와 시간의 확인 가능
- 서명할 당시의 내용을 인증 가능
- 분쟁시 제 3자가 확인 가능

전자서명 특징

위조불가(Unforgeable)

: 서명자만이 서명문을 생성 가능

서명자 인증(Authentic)

: 서명문의 서명자를 확인 가능

재사용 불가(Not Reusable)

: 서명문의 서명은 다른 문서의 서명으로 사용 불가능

변경 불가(Unalterable)

: 서명된 문서의 내용 변경 불가능

부인 불가(Nonrepudiation)

: 서명자는 후에 서명한 사실을 부인 불가능

전자서명 요구 조건

- 서명은 메시지에 의존하는 **비트 형태**이어야 한다.
- 위조와 부인 방지 위해, **송신자의 유일한 정보 비트를 이용**해야 함
- 서명문을 **만들기가 쉬워야** 한다.
- 서명문을 **인식, 확인 하기가 쉬워야** 한다.
- 서명문을 **위조하는 것이 계산적으로 실행 불가능**
- 기억장소에 서명문의 **복사본을 유지**하는 것이 실용적이어야 한다.

디지털 서명

- 디지털 서명(digital signature)
  - Alice가 사용하는 Key는 Alice만이 알고 있는 개인적인 것
  - Alice는 메시지 송신 시에 그 개인적인 Key를 써서 「서명」을 작성
  - 수신자 Bob은 Alice의 개인 Key와는 다른 Key를 써서 「서명」을 검증

- 메시지의 서명을 작성하는 행위
  - 디지털 서명
    - 「서명용 Key」와 「검증용 Key」가 분리
    - 검증용 Key로 서명을 작성할 수 없음
- 메시지의 서명을 검증하는 행위
  - 「서명용 Key」는 서명을 하는 사람만이 가지고 있지만,
  - 「검증용 Key」는 서명을 검증하는 사람이라면 누구라도 가질 수 있음

서명 작성과 서명 검증

❑ 공개키 암호와 디지털 서명키 사용 방법

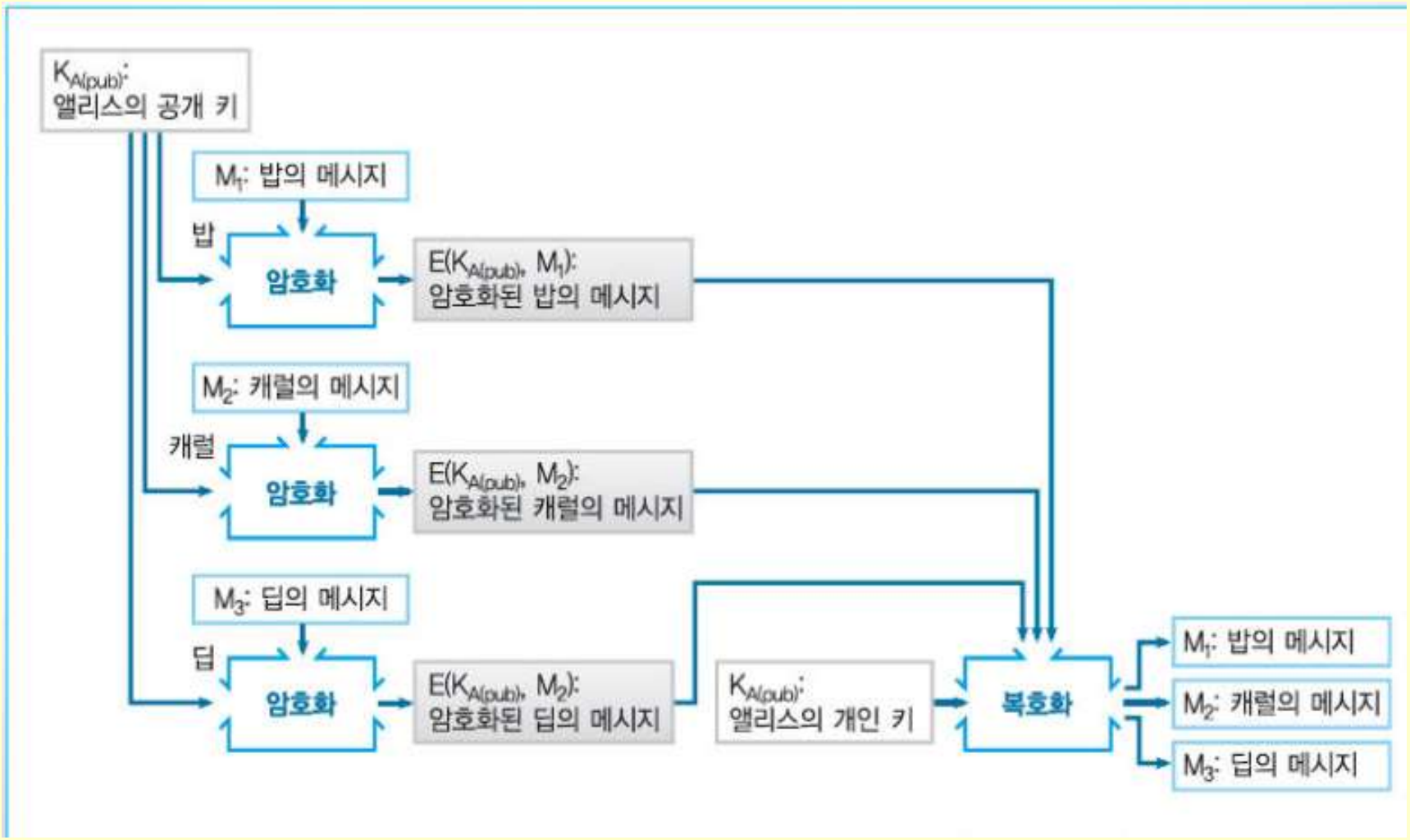
	개인 Key	공개 Key
공개키 암호	• 수신자가 복호화에 사용	• 송신자들이 암호화에 사용
디지털 서명	• 서명자가 서명 작성에 사용	• 검증자들이 서명 검증에 사용
Key는 누가 갖는가?	• 개인	• 필요한 사람은 아무나

❑ 메시지를 개인 Key로 암호화하는 것이 **서명 작성**에 해당

❑ 암호문을 공개 Key로 복호화하는 것이 **서명 검증**에 해당

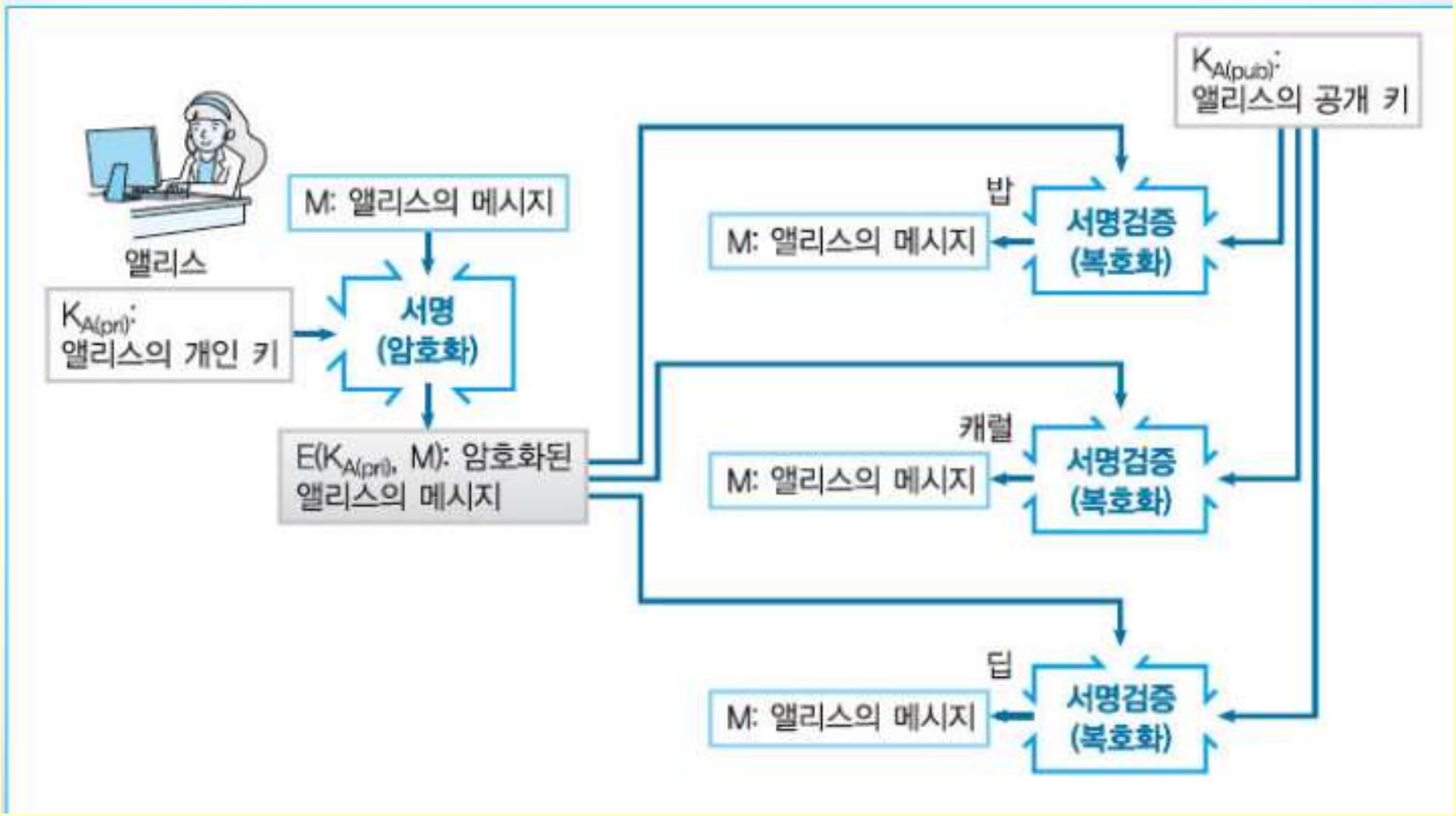
공개키 암호와 디지털 서명

❑ 공개키 암호는 누구라도 암호화



공개키 암호와 디지털 서명

❑ 디지털 서명은 누구라도 서명검증 가능



전자서명 (Digital Signatures)

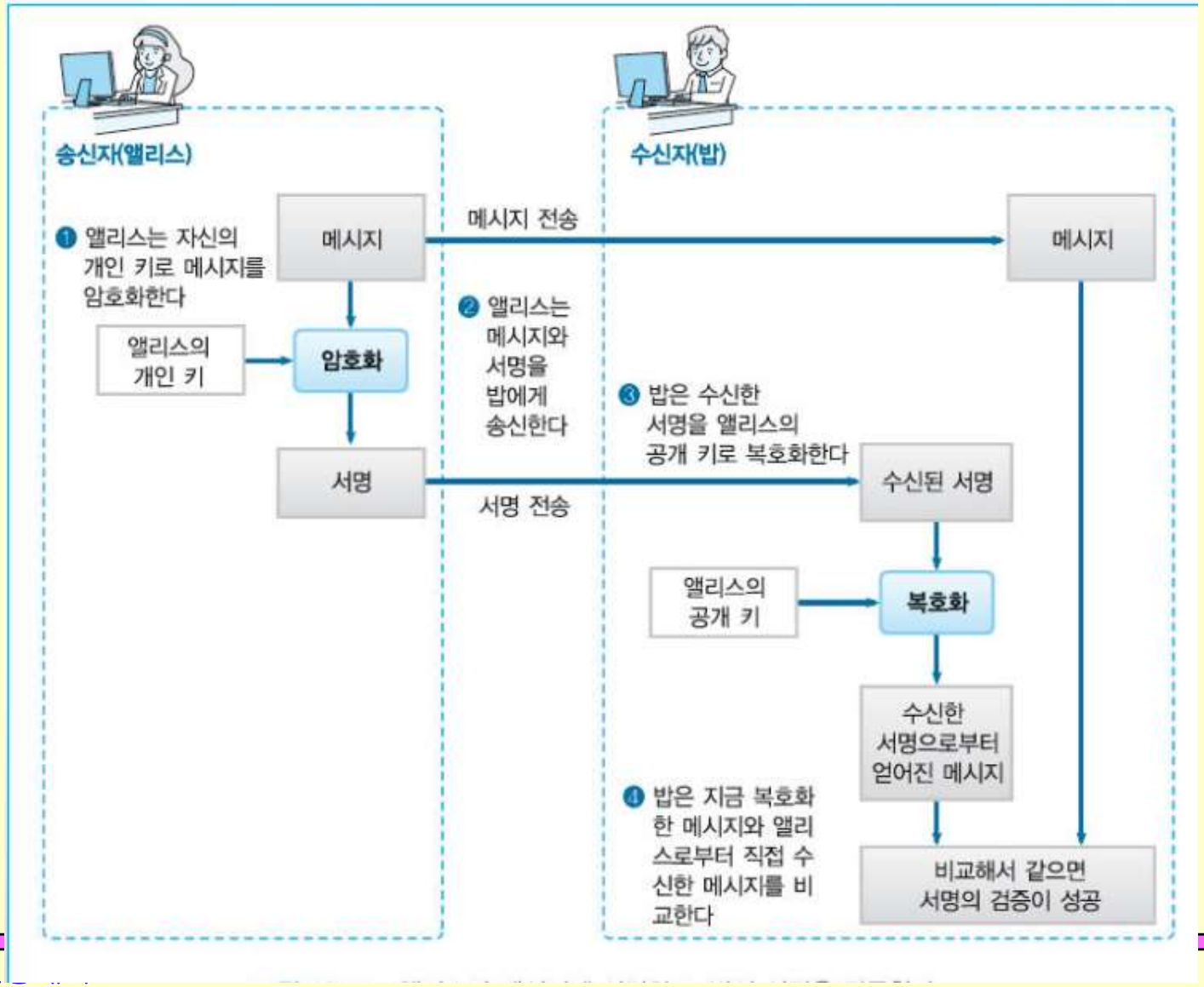
- ❖ 정의
  - ➢ 오직 통신하는 상대방들만이 참여하는 전자서명 기법
    - ▪ 통신하는 상대방(출신, 목적지)만을 포함
  - ➢ 송신자(서명자)의 공개키를 수신자(검증자)가 안다고 가정
- ❖ 직접적 디지털 서명 방식
  - ➢ 송신자의 개인키를 가지고 전체 메시지를 암호화(서명)
  - ➢ 송신자의 개인키를 가지고 메시지의 해쉬 코드를 암호화(서명)
- ❖ 비밀성
  - ➢ 서로 간에 공유되는 비밀키로 메시지와 서명을 암호화함으로써 제공
  - ➢ 서명을 먼저 수행하고 암호화 수행
- ❖ 분쟁
  - ➢ 서명이 암호화된 메시지에 대해 계산된 것이라면, 제 3자는 복호키에 대한 접근 필요
  - ➢ 서명이 평문 메시지에 대한 것이라면, 수신자는 평문 메시지와 서명을 보관하여 차후 분쟁 해결 가능

디지털 서명 방법-메시지에 직접 서명

1. Alice는 자신의 개인 Key로 메시지를 암호화(서명)한다.
2. Alice는 메시지와 서명문을 Bob에게 송신한다.
3. Bob은 수신한 서명을 Alice의 공개 Key로 복호화(검증)한다.
4. Bob은 이제 서명문을 복호화(검증)해서 얻어진 메시지와 Alice로부터 직접 수신한 메시지를 비교한다.

디지털 서명 방법-메시지에 직접 서명

- Alice가 메시지에 서명하고 Bob이 서명 검증



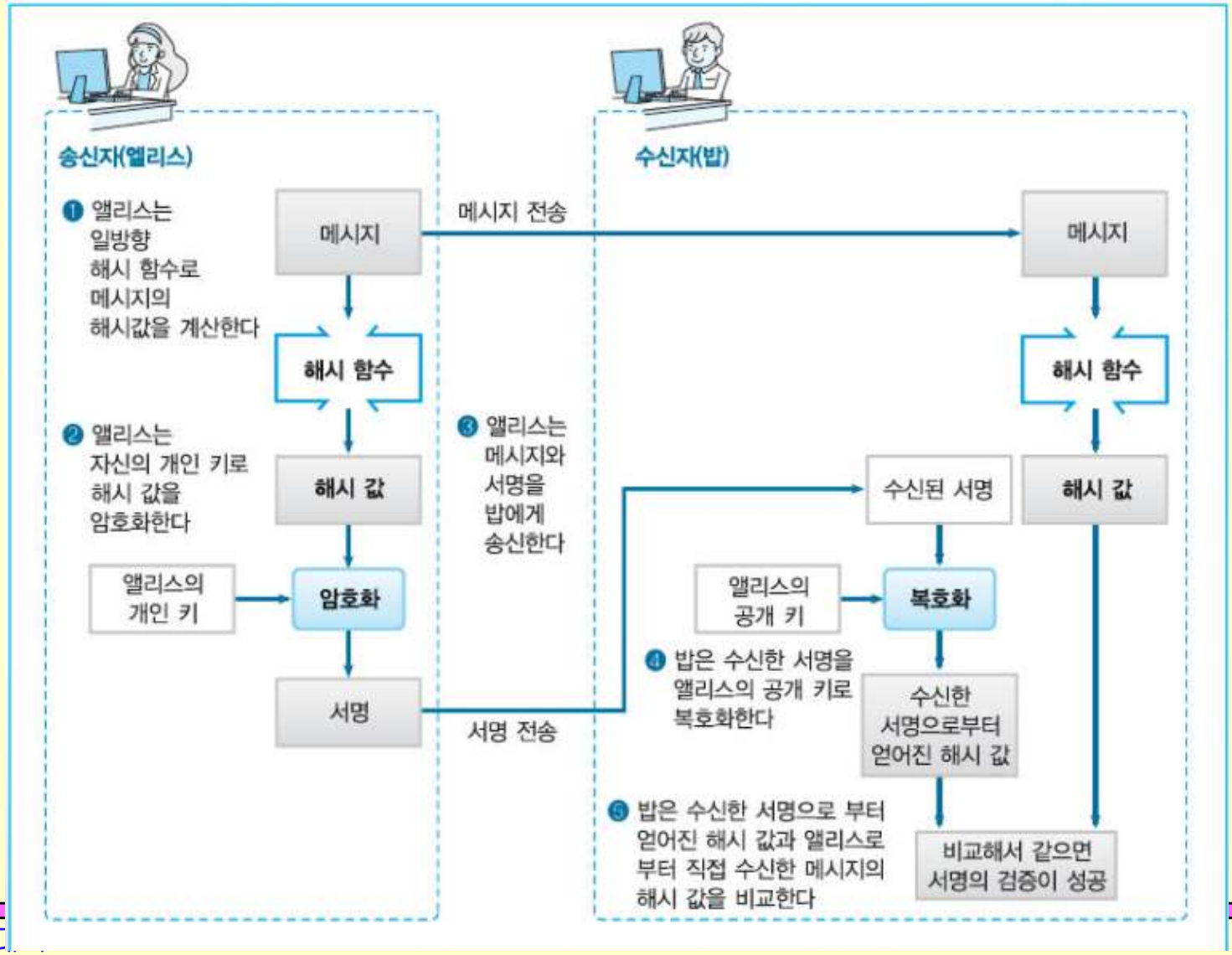
디지털 서명 방법-메시지의 해시 값에 직접 서명

1. Alice는 일방향 해시 함수로 메시지의 해시 값을 계산한다.
2. Alice는 자신의 개인 Key로 해시 값을 암호화 (서명) 한다.
3. Alice는 메시지와 서명을 Bob에게 송신한다.
4. Bob은 수신한 서명을 Alice의 공개 Key로 복호화 (검증) 한다.
5. Bob은 수신한 서명으로부터 얻어진 해시 값과 Alice로부터 직접 수신한 메시지의 해시 값을 비교한다.

✓ 메시지가 아무리 길어도 해시 값은 짧기 때문에 암호화(서명)하는 것이 훨씬 수월해짐

디지털 서명 방법-메시지의 해시 값에 직접 서명

- Alice가 메시지의 해시값에 서명하고, Bob이 서명 검증



직접 서명의 단점

- 구조의 정당성은 송신자의 개인키에 달려 있음
- 송신자가 개인키를 분실, 도난 당했다고 주장이 가능
- 실제로 개인키를 도난 당했을 경우의 대책 미흡
- 사례
  - ❖ 공갈 협박에 의해 개인키을 노출하고 침묵 가능
    - ➢ 신고 이전 까지는 심각한 손상 초래
  - ❖ 사고 발생시에 불리할 경우 도난 당했다고 거짓 주장
  - ❖ 실제로 자신도 모르는 사이에 도난 당했을 경우 가능성 존재

디지털 서명에 대한 의문

- 디지털 서명은 정말로 종이 서명 대용이 되는 것일까?
  - ❖ 한국에서는 1999년 전자서명법이 제정, 시행
  - ❖ 이 법률들은 전자적으로 실현된 「서명」을 날인이나 손으로 쓴 서명과 같이 취급하기 위한 법적인 근거

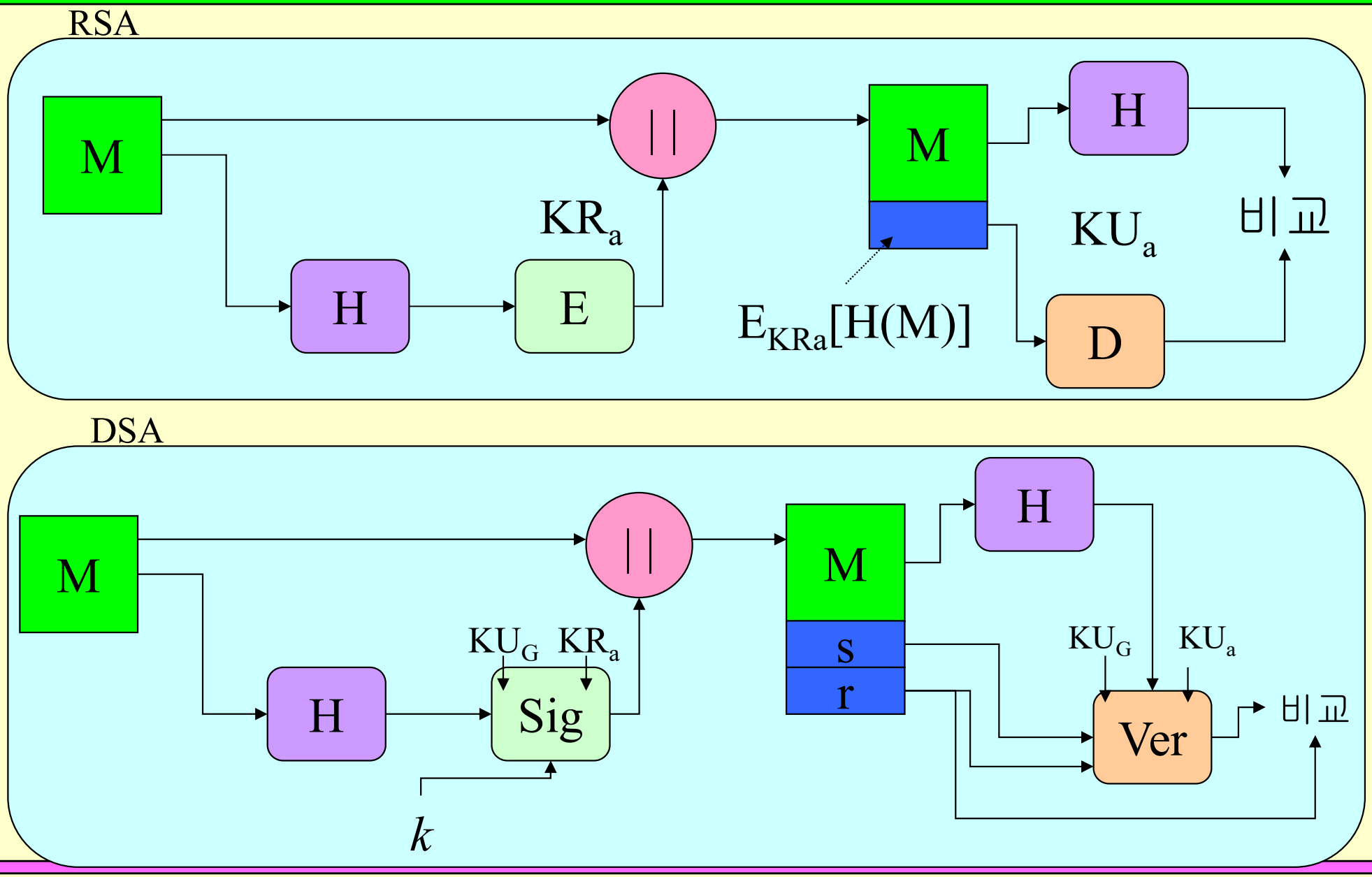
전자서명의 표준 (Digital Signature Standard)

- 전자서명표준
  - DSS (Digital Signature Standard)
  - NIST (National Institute of Standards and Technology) 가 제안한 것으로 FIPS PUB 186임
  - SHA(Secure Hash Algorithm) 이용
- 전자서명알고리즘
  - DSA (Digital Signature Algorithm)
  - DSS에서 이용되는 알고리즘
- 1991년 제안. 1993년 개정. 보조적 추가 개정, 1994년 1월 채택.
- 2000년 확장 버전이 FIPS 186-2로 발표됨
- 2009년 FIPS 186-3으로 개정됨
  - RSA 및 타원곡선암호학 (ECC)에 기반한 전자서명 알고리즘을 포함함

디지털 서명 표준(DSS)

- 1991년 미국 NIST에서 표준안으로 개발
- SHA를 사용하는 DSA(Digital Signature Algorithm)
- 공개키 기술을 사용
  - RSA와는 달리 암호화/키 교환에는 이용되지 않음
  - 서명만을 위한 알고리즘
- 이산 대수의 어려움에 기반을 둔 알고리즘
- ElGamal과 Schnorr에 의해 제안된 알고리즘에 기반을 둠

RSA와 DSS의 접근법 비교



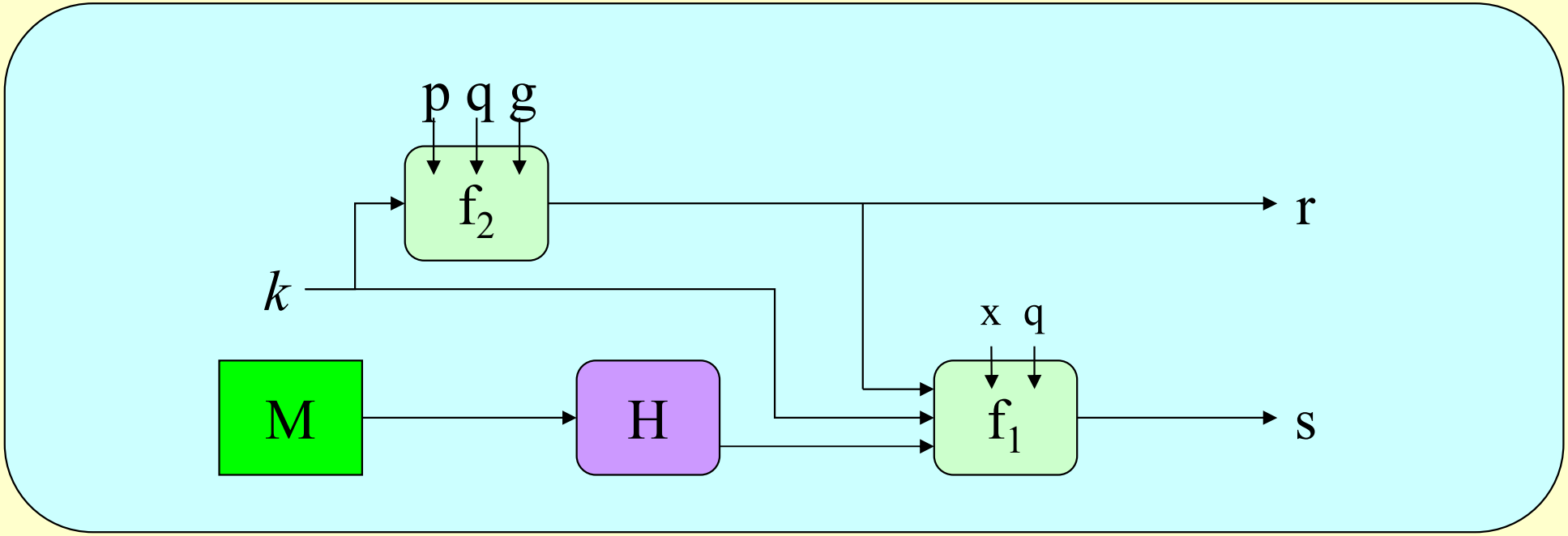
DSS 알고리즘 요약

서명 준비 과정

- 사용자 집단에 공통적으로 사용되는 3개의 매개 변수
  - 160비트 길이의 소수 q 선택
  - 512~1024 비트 사이의 소수 p 선택
  - $g = h^{(p-1)/q} \bmod p$ 형식의 값 선택 ($1<h<(p-1)$)
- 개인키 선택, 공개키 생성
  - 랜덤한 개인키 x 선택 ($0<x<q$)
  - 개인키 x로부터 공개키 y 계산
    - $y=g^x \bmod p$

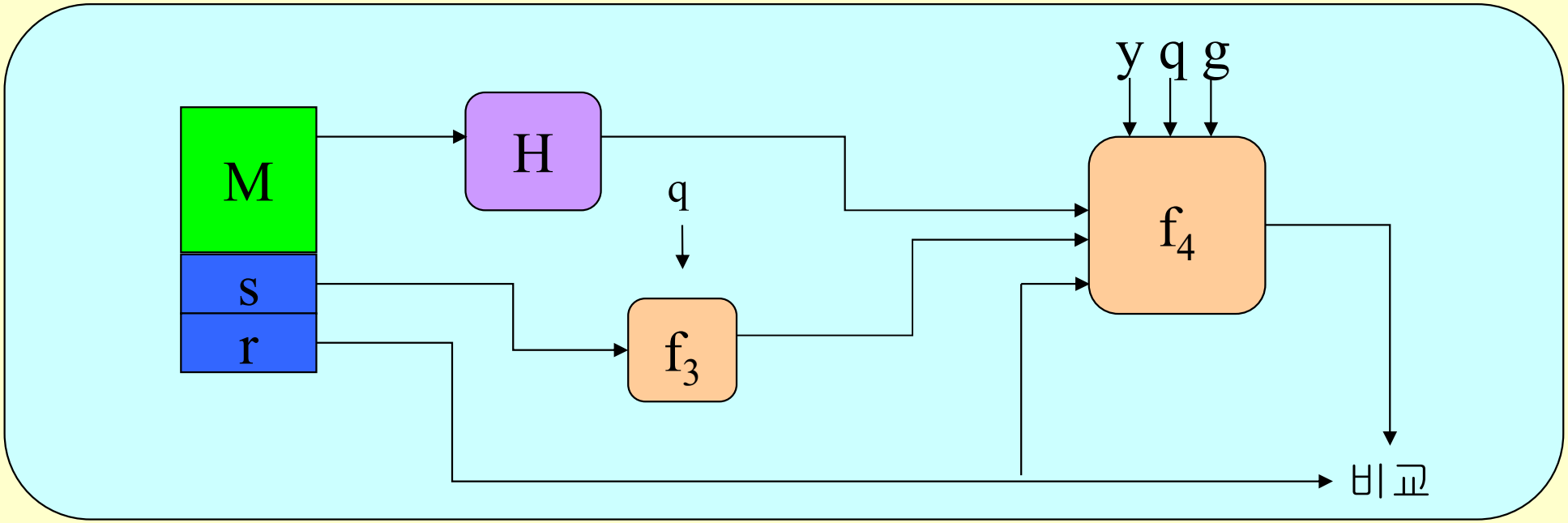
서명 과정

- 메시지별 비밀번호 k를 랜덤하게 생성(0<k<q)
- 서명
  - $r=(g^k \bmod p) \bmod q$
  - $s=[k^{-1}(H(M)+xr)] \bmod q$
  - Signature = (r, s)



서명 검증 과정

- $w=s^{-1} \bmod q$
- $u_1=[(H(M)w)] \bmod q$
- $u_2= rw \bmod q$
- $v=[(g^{u1}\ y^{u2}) \bmod p] \bmod q$
- Verify : **v = r**



DSS에 대한 비판

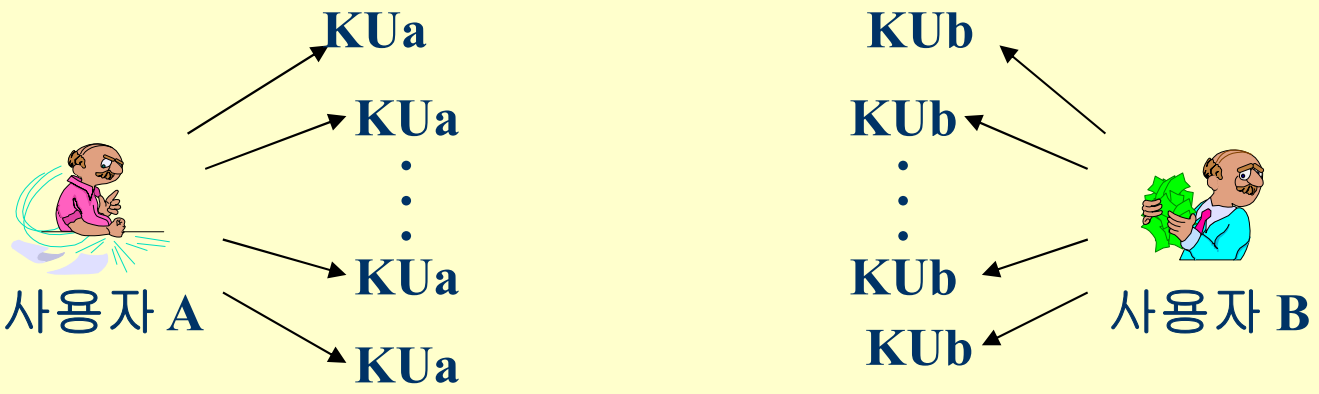
- 암호화나 키 분배에 활용될 수 없다.
- NSA에 의해 주도되었으므로 트랩도어가 있을 수 있다.
- RSA에 비하여 수행 속도가 부분적이긴 하지만 늦다.
- RSA가 이미 널리 쓰이고 있다.
- 제정 과정이 불투명하고 비판할 시간적 여유가 없다.
- 특허에 저촉되고 있다.
- 키의 크기가 작다.

10장 키 관리와 분배
(Key Management and Distribution)

키 관리- 공개키의 분배

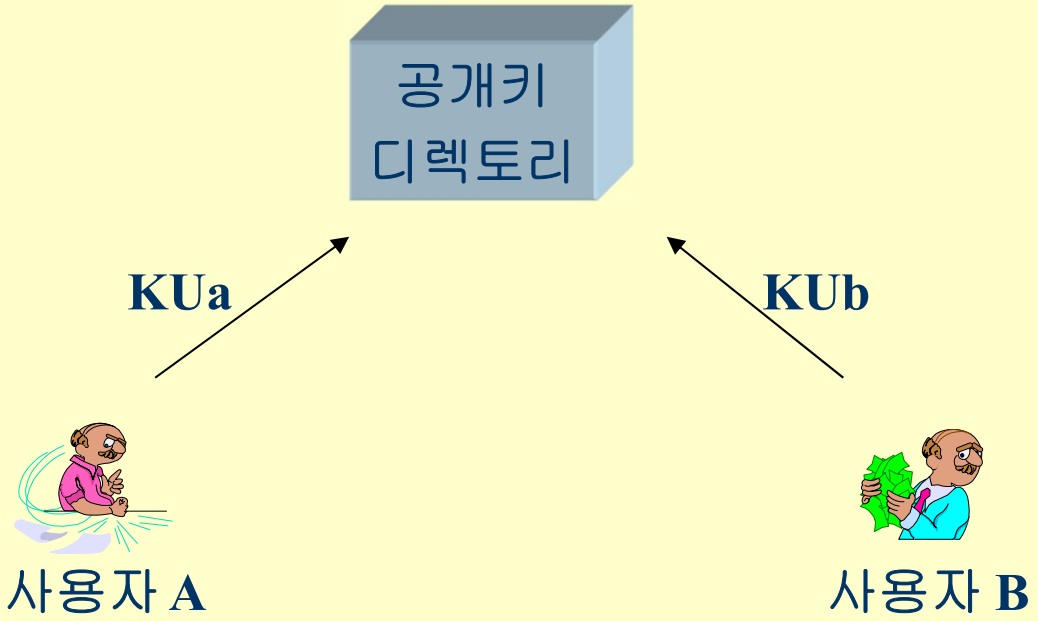
❑ 공개키의 공개 발표

- ❖ 자신의 공개키를 다른 사용자에게 전송 등의 방법으로 공개
- ❖ 문제점
  - ➢ 어떤 사용자가 다른 사용자 A로 위장하여 공개키 공개 (A에 전송되는 암호화 메시지를 읽을 수 있게 됨)



키 관리- 공개키의 분배

❑ 공개적으로 사용 가능한 디렉토리



키 관리- 공개키의 분배

❑ 공개적으로 사용 가능한 디렉토리(계속)

❖ 필요한 사항

- 기관은 각 가입자에 대한 {이름, 공개키}의 디렉토리 유지
- 각 가입자는 디렉토리 기관에 공개키 등록
- 가입자는 필요시 새로운 것으로 교체 가능
- 기관은 디렉토리를 공포
- 가입자는 전자적으로 디렉토리 접근 가능

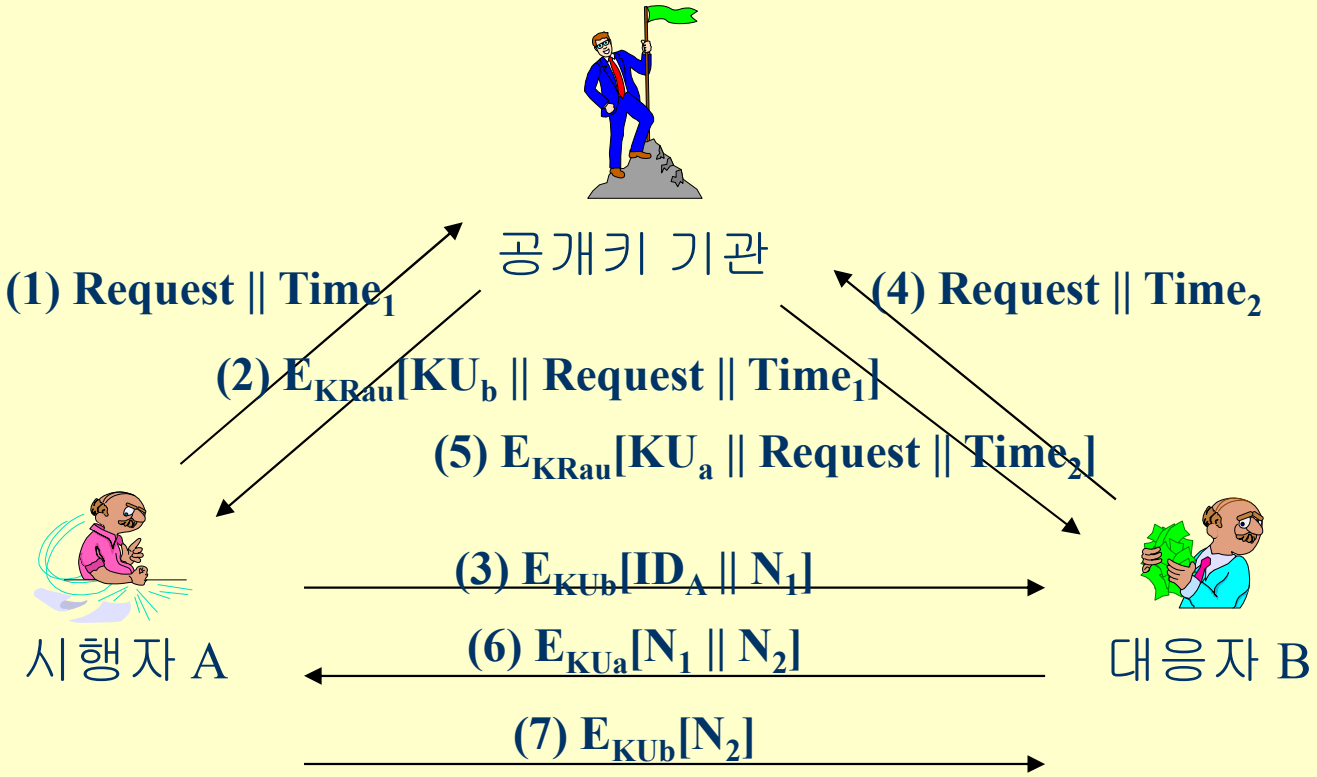
❖ 문제점

- 디렉토리 기관으로 위장하여 공포 및 갱신 배포 교란
- 제 3자가 다른 가입자의 공개키 정보를 수정

키 관리- 공개키의 분배

❑ 공개키 기관

: 공개키 기관에서 공개키 발행, 보관 분배 제어



키 관리- 공개키의 분배

❑ 공개키 기관 (계속)

(1) 단계

❖ B의 공개키에 대한 요구를 타임스템프와 함께 전송

(2) 단계

❖ 공개키 기관은 B의 공개키와 (1) 단계의 메시지를 자신의 개인키로 암호화하여 전송

(3) 단계

❖ B의 공개키를 저장하고, A의 식별자(ID_A)와 임시비표(N_1)을 B의 공개키로 암호화하여 전송

키 관리- 공개키의 분배

❑ 공개키 기관 (계속)

(4), (5) 단계

❖ B는 (1), (2) 단계와 같은 방법으로 A의 공개키 획득

(6) 단계

❖ B는 임시비표 N_1, N_2를 A의 공개키로 암호화하여 전송

(7) 단계

❖ A는 N_2를 B의 공개키로 암호화하여 전송

❑ 문제점

❖ 공개키 기관의 디렉토리 수정에 취약

❖ 공개키 기관의 온라인 응답 한계와 시스템의 병목 현상

키 관리- 공개키의 분배

❑ 공개키 인증서

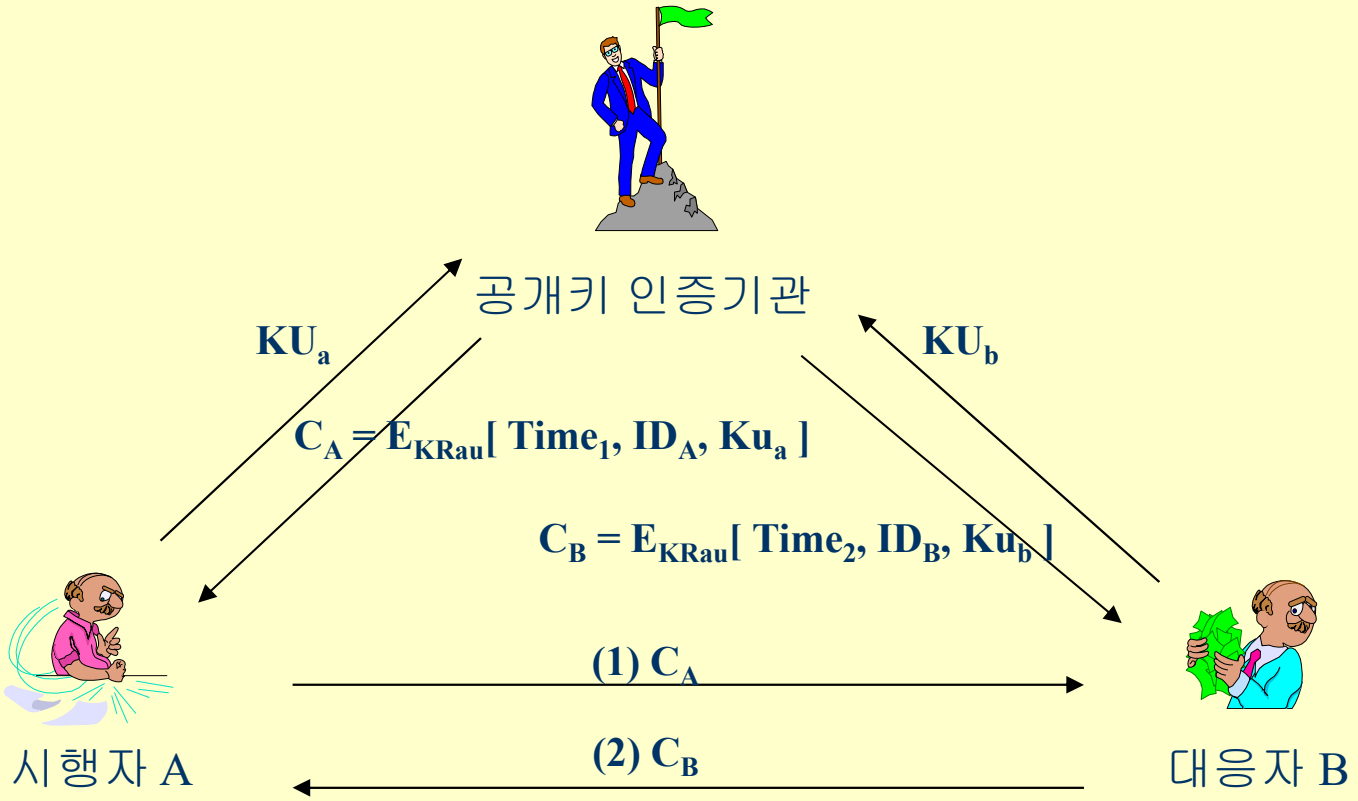
❖ 인증기관이 공개키 인증서를 발행하여 공개키의 소유자, 유효기간, 무결성을 보증

❖ 인증서 방식을 위한 요구 사항

- ➢ 임의의 가입자는 인증서의 내용(이름, 공개키) 확인 가능
- ➢ 임의의 가입자는 인증서의 정당성 확인 가능
- ➢ 인증 기관만이 인증서 생성과 갱신 가능
- ➢ 임의의 가입자는 인증서의 적시성 확인 가능

키 관리- 공개키의 분배

❑ 공개키 인증서 (계속)



: **사용자는 공개키 인증서를 받은 후 인증서로 공개키 전송**

공개키 분배

❑ 공개키 인증서(certificate)

- ❖ 인증서 사용
  → 공개키 기관을 경유하지 않으면서 신뢰성 있는 키 교환
- ❖ 인증서
  - ➢ 공개키, 소유자 식별자, 전체 블록에 대한 제 3의 신뢰 개체의 서명
  - ➢ 제 3의 신뢰 개체: 정부 기관, 금융 기관 등
- ❖ 사용자는 기관에 공개키 제출 후 인증서 받아 공개
- ❖ 누구나 인증서 얻을 수 있음
- ❖ 서명 통해 인증서의 유효성 검증
- ❖ 인증서 전송으로 키 정보 전달
- ❖ 기관이 생성한 인증서임을 검증 가능

공개키 분배

❑ 공개키 인증서 요구사항

- ❖ 어떤 참가자도 인증서로 부터 인증서 소유자의 이름과 공개키를 알 수 있어야 한다
- ❖ 어떤 참가자도 인증서가 위조된 것이 아니며, 인증기관으로 부터 생성된 것임을 검증할 수 있어야 한다.
- ❖ 인증기관만이 인증서를 생성하고 갱신할 수 있어야 한다.
- ❖ 어떤 사용자도 인증서의 현재성을 검증할 수 있어야 한다.

- ❖ X.509 표준
  - ➢ 널리 수용된 인증서 형식
  - ➢ IP 보안, 전송 계층 보안, S/MINE 등에서 사용

공개키 분배

❑ 공개키 인증서

- ❖ PR_{auth}: 기관의 개인키
- ❖ T: 타임스탬프

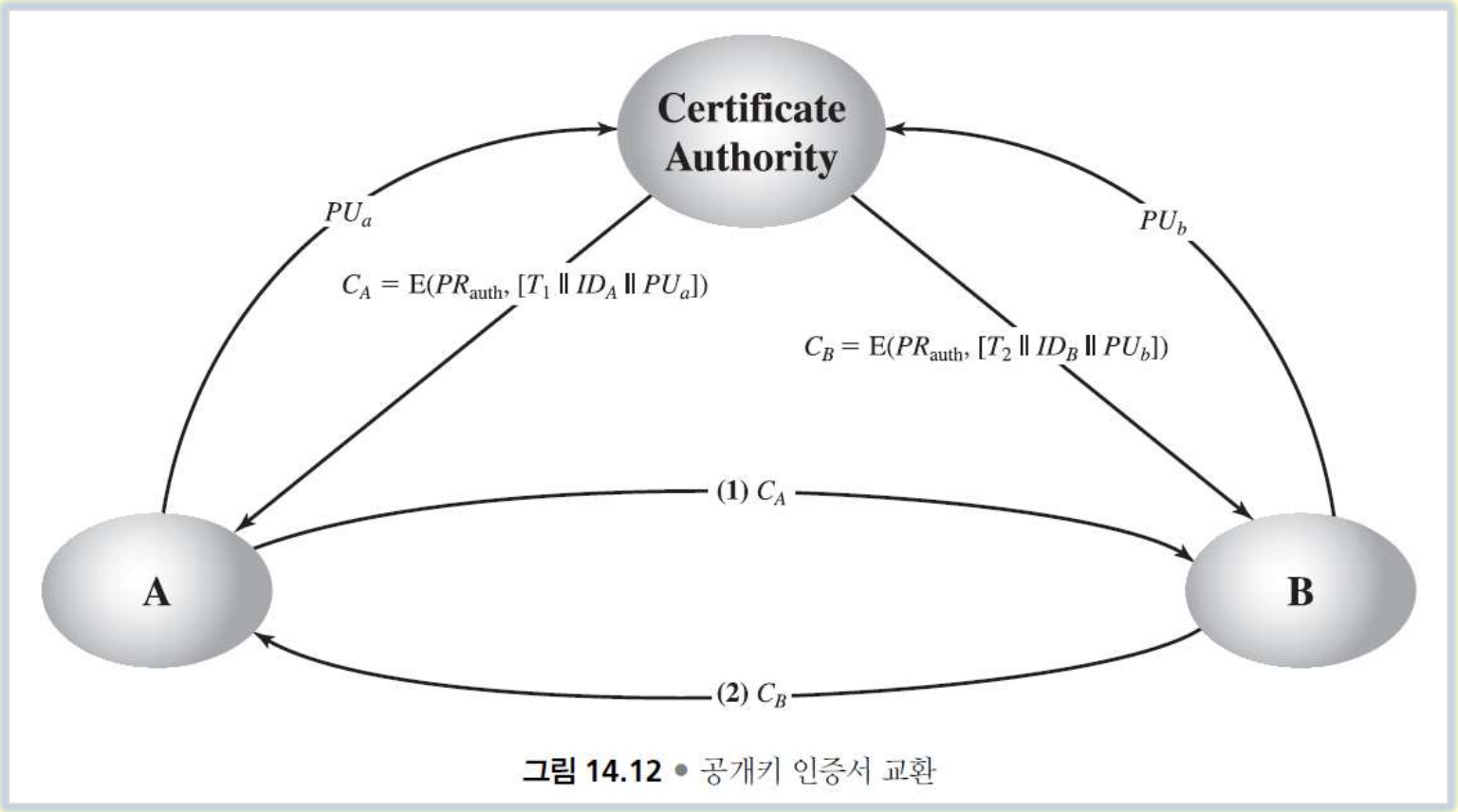
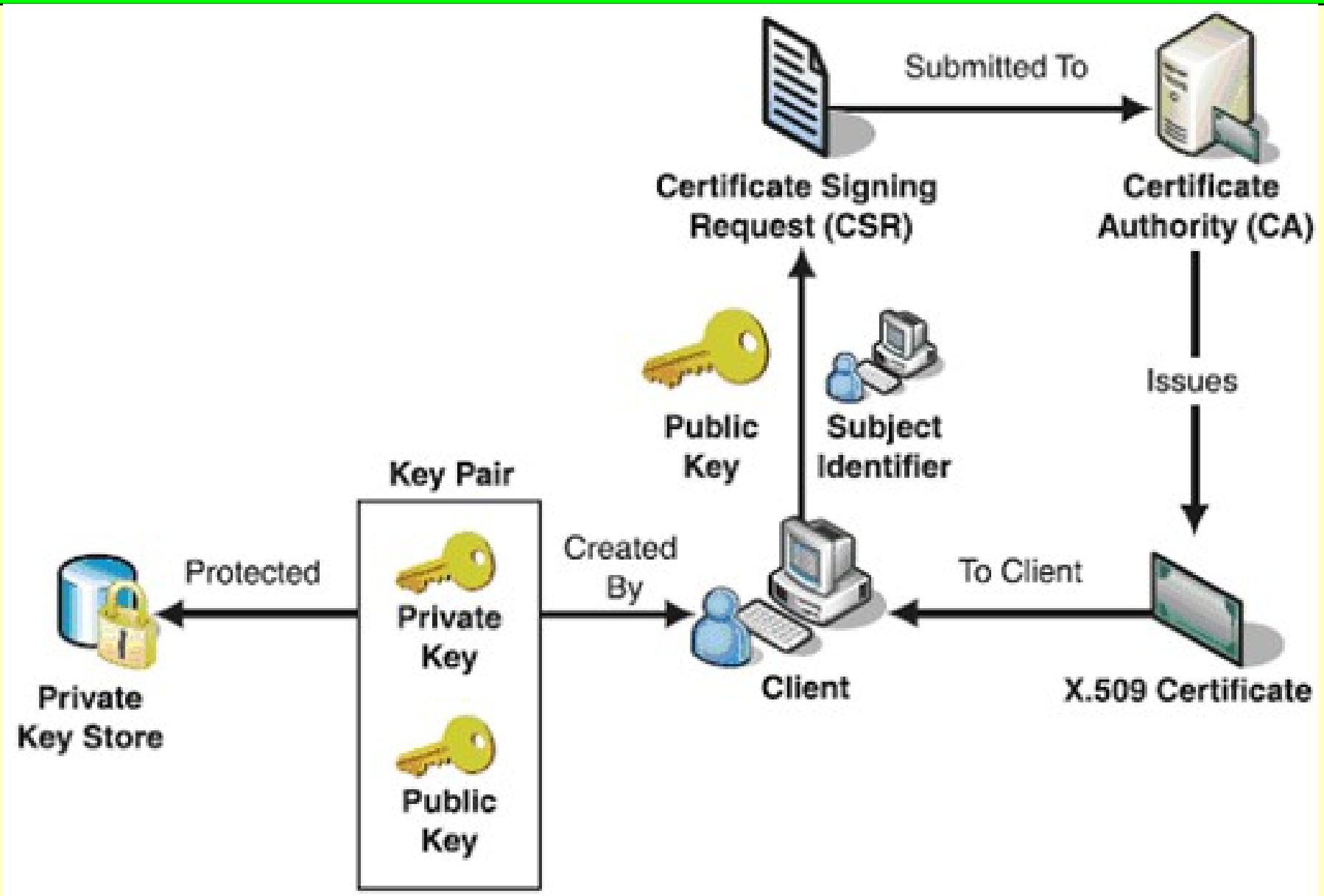
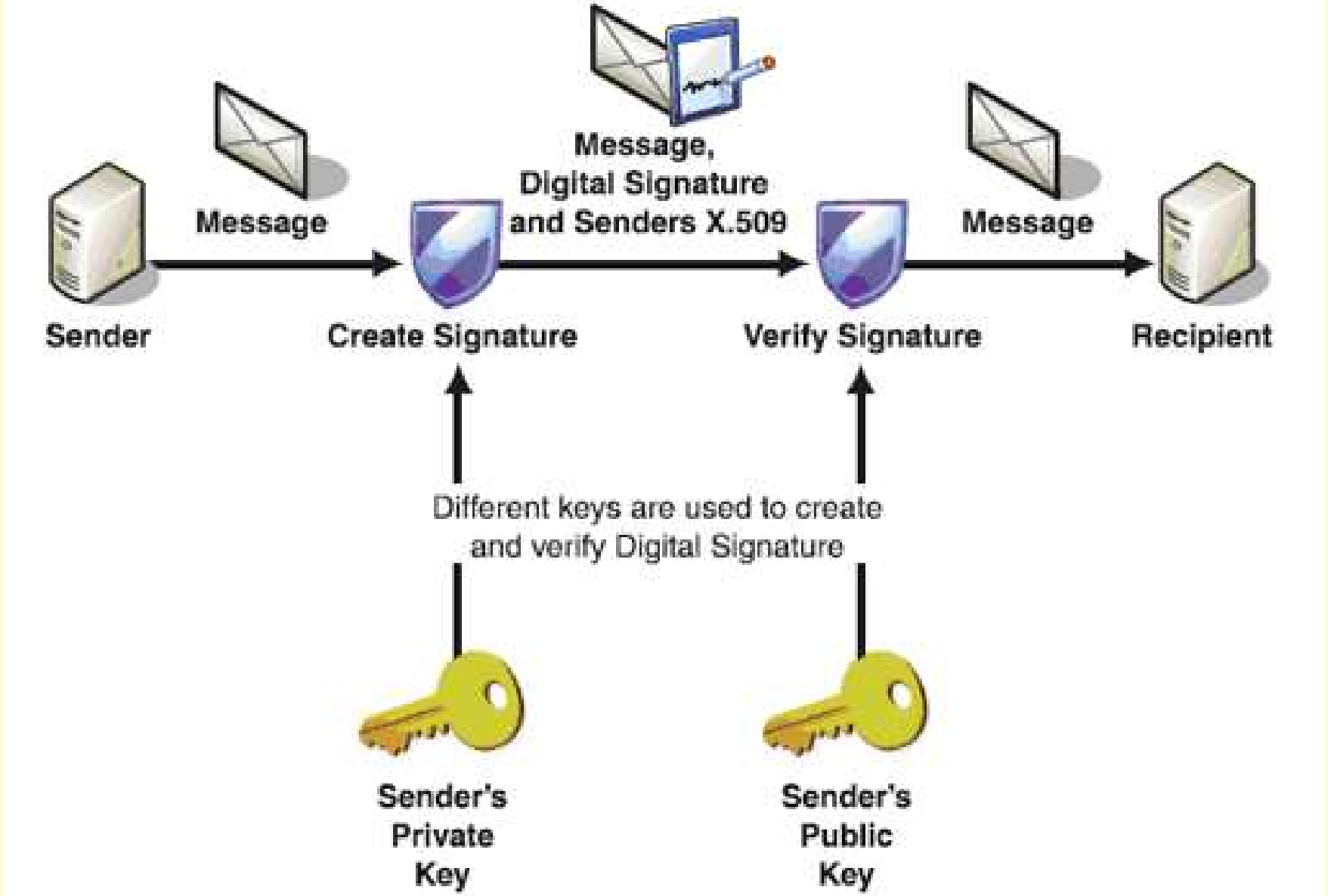


그림 14.12 • 공개키 인증서 교환

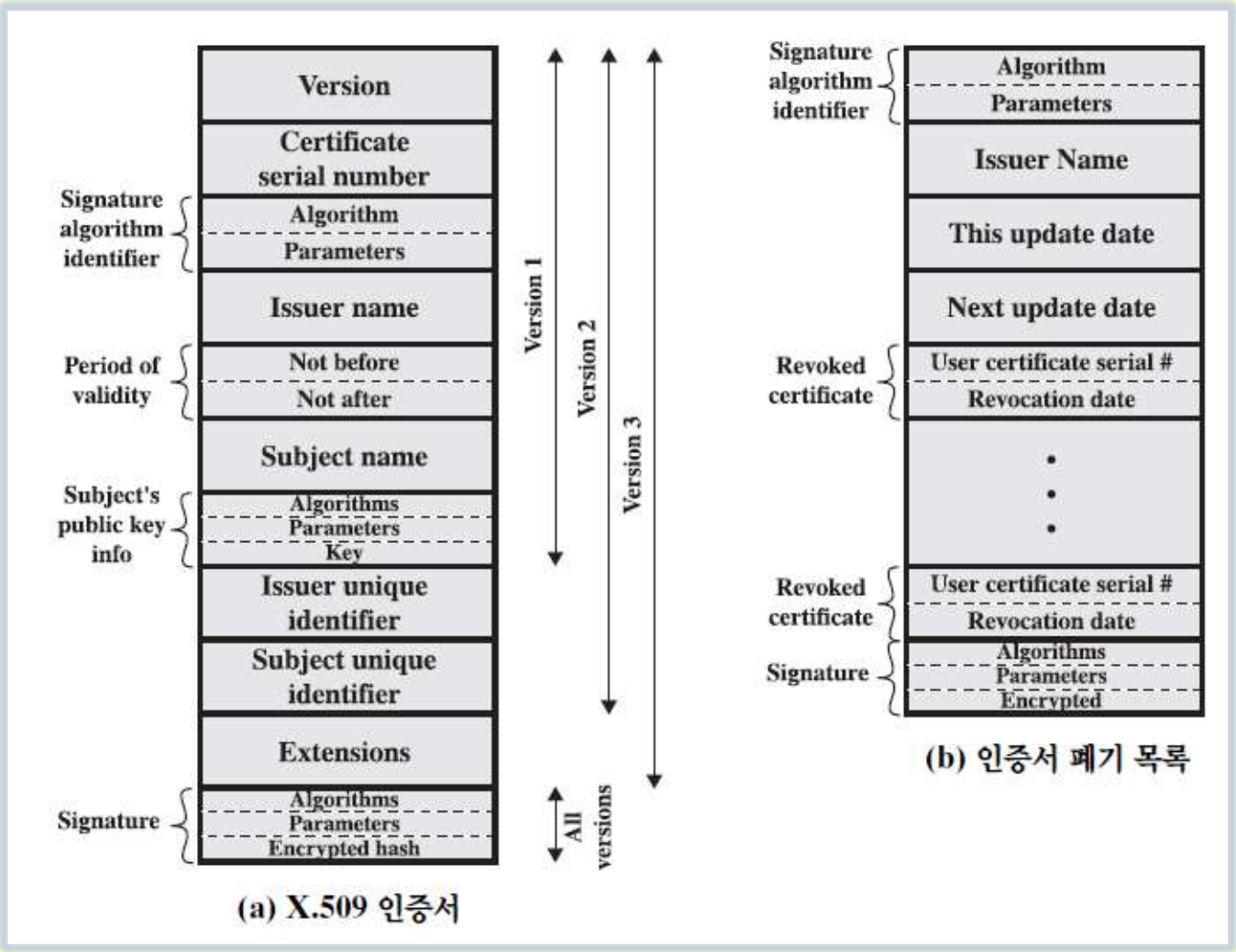

Submitted To
Certificate Signing Request (CSR)
Certificate Authority (CA)
Public Key
Subject Identifier
Issues
Key Pair
Protected
Private Key
Created By
To Client
Private Key Store
Public Key
Client
X.509 Certificate


Message
Message,
Digital Signature
and Senders X.509
Message
Sender
Create Signature
Verify Signature
Recipient
Different keys are used to create
and verify Digital Signature
Sender's
Private
Key
Sender's
Public
Key

X.509 인증서

❑ 인증서

- ❖ 신뢰 인증기관 (CA)가 발행
- ❖ CA에 의해 디렉토리에 저장
- ❖ 인증서 형식



X.509 인증서

❑ 인증서

❖ 인증서 형식

- **버전**: 인증서 형식의 버전. 기본 버전은 1. 만약 발행자 유일 식별자나 주체 유일 식별자가 있으면 버전은 2. 하나 이상의 확장 필드들이 있으면 버전은 3
- **일련번호**: 이 인증서와 연관된 값. 발행 CA 내부의 유일한 정수 값.
- **서명 알고리즘 식별자**: 인증서를 서명하기 위해 사용된 알고리즘 및 관련된 파라미터들.
- **발행자 이름**: 이 인증서를 발행하고 서명한 CA의 X.500이름.
- **유효기간**: 인증서가 유효한 첫 번째와 마지막 날짜.
- **주체 이름**: 이 인증서가 가리키는 사용자의 이름.

X.509 인증서

❑ 인증서

❖ 인증서 형식

- **주체의 공개키 정보**: 주체의 공개키와 이 키가 사용될 알고리즘의 식별자 및 관련 파라미터들
- **발행자 유일 식별자**: X.500 이름이 다른 개체들에 의해 재사용될 경우, 발행자 CA를 유일하게 식별하기 위해 사용되는 비트열. 옵션
- **주체 유일 식별자**: X.500 이름이 다른 개체들에 의해 재사용될 경우, 주체를 유일하게 식별하기 위해 사용되는 비트열. 옵션 필드
- **확장**: 하나 이상의 확장 필드들의 집합. 이 확장들은 버전 3에서 추가
- **서명**: 인증서의 다른 모든 영역에 대한 해쉬값을 CA의 개인키로 암호화한 값을 포함. 이 영역은 서명 알고리즘 식별자를 포함

❖ CA가 자신의 개인키로 인증서 서명 → 인증서 유효성 확인 가능

공인인증서의 개념

❑ 공인인증서의 개요

❖ 공인인증서

- 전자서명이란 전자문서를 작성한 사람의 신원과 전자문서의 변경여부를 확인할 수 있도록 하는 고유정보를 의미하며 전자문서의 인감과 같은 역할

- 전자서명은 공개키 암호기술을 이용한 것으로 전자서명 생성키(개인키)와 전자서명 검증키(공개키)로 구성되는 하나의 키 쌍으로 이루어짐

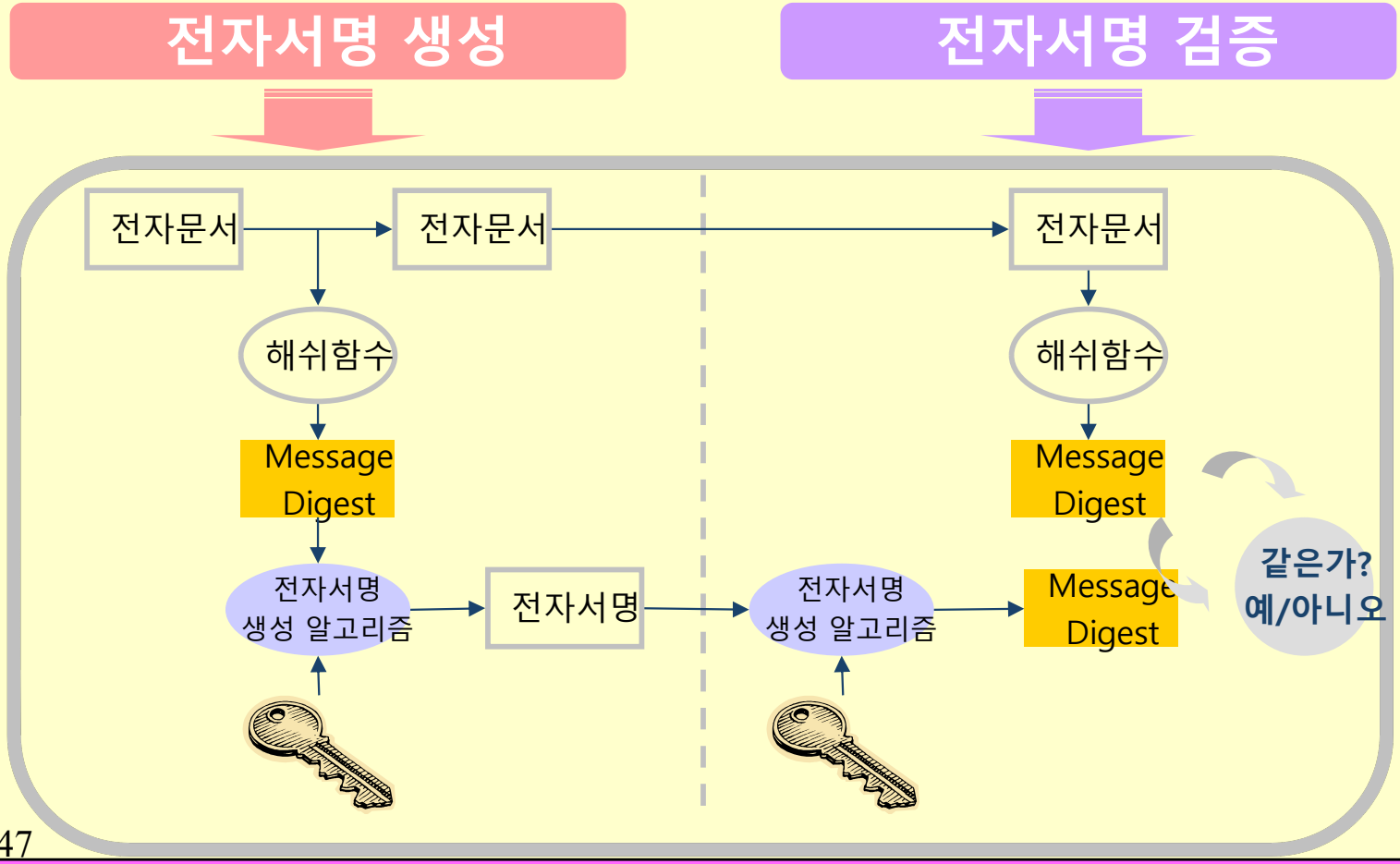
- 공인인증기관이 발행한 사이버 거래용 인감증명서

  - PKI(Public Key Infrastructure) 기반 구조
  - CA(Certificate Authority)

공인인증서의 개념

❑ 공인인증서의 개요

❖ 전자서명 생성 및 검증 과정



공인인증서의 개념

❑ 전자서명 공인 인증서 발급

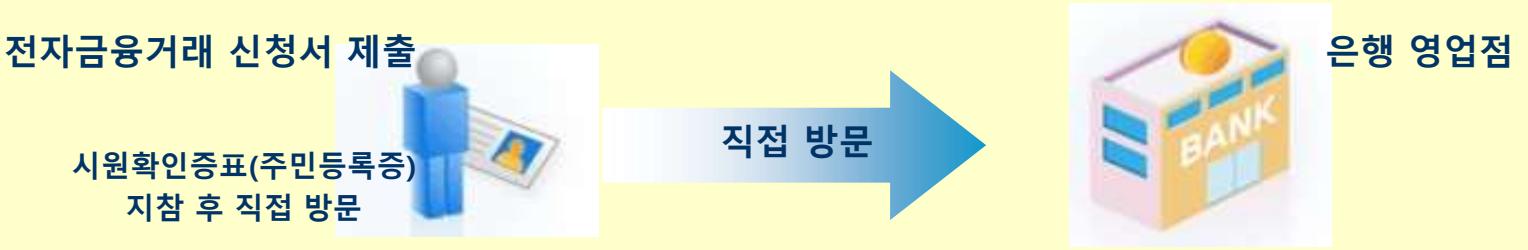
공인인증서의 개념

❑ 공인인증서의 발급

❖ 발급 절차

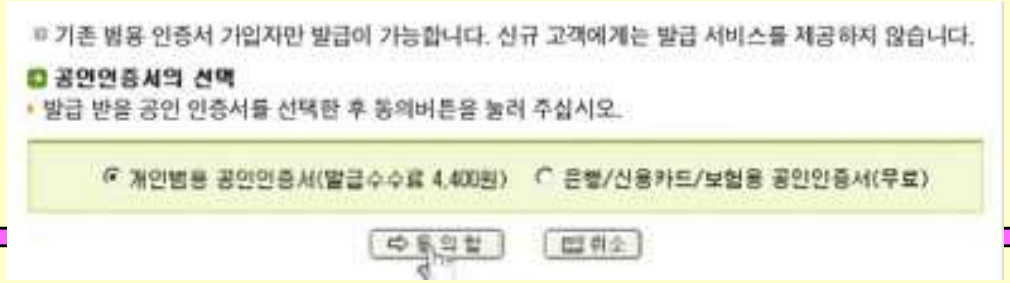
➢ Step 1 : 은행 방문

- **전자금융거래신청서 제출**



➢ Step 2 : 인증서 신청

- **전자금융거래 이용자 등록 후 가정이나 사무실 PC를 통해 해당 은행 인터넷 뱅킹 홈페이지에 접속하여 약관에 동의 후 인증서 발급을 신청**



공인인증서의 개념

- 공인인증서의 발급
  - ❖ 발급 절차
    - ➢ Step 3 : 정보 입력
      - 인증서 발급을 위한 필요 정보를 입력

- ➢ Step 4: 저장매체 선택
  - 인증서의 저장매체를 선택한 후 인증서 암호를 설정

공인인증서의 개념

“공인 인증서” 관리 10계명

- 제 3자가 쉽게 추측할 수 있는 비밀번호를 사용하지 말 것
- 전자금융거래 비밀번호와 계좌 번호를 반드시 다르게 사용할 것
- 비밀번호를 정기적으로 변경하고, 노출되었다고 의심되는 경우 빠른 시간 내에 금융회사 통보 및 변경 조치할 것
- 공인인증서를 하드디스크에 저장하지 말 것
- 전자금융거래에 필요한 정보를 수첩, 지갑 등에 기록하지 말 것
- 전자금융거래를 절대로 타인에게 위탁하거나 관련 정보를 알려 주지 말 것
- 전자금융거래 이용내역을 본인에게 즉시 알려주는 서비스를 적극 이용할 것
- PC 방 등 개방된 컴퓨터는 가급적 사용을 자제하고, 사용한 경우에는 관련 정보를 삭제할 것
- 전자금융거래의 1회 이체한도 및 일일 이체한도를 적절히 설정할 것
- PC에 백신 프로그램을 설치하여 실행함으로써 해킹 등의 보안 침해 사고에 대비할 것

<출처 : 금융 감독원>