# 제 10 장
# 디지털 서명

박 종 혁 교수

Tel: 970-6702

Email: jhpark1@seoultech.ac.kr

1절 디지털 서명

2절 디지털 서명 방법

3절 디지털 서명에 대한 의문

4절 디지털 서명 활용 예

5절 RSA에 의한 디지털 서명

6절 다른 디지털 서명

7절 디지털 서명에 대한 공격

8절 기타 기술과의 비교

9절 디지털 서명으로 해결할 수 없는 문제

# 제1절 디지털 서명

## 1.1 앨리스 차용서

## 1.2 메시지 인증코드에서 디지털 서명으로

## 1.3 서명 작성과 서명 검증

## 1.4 공개 키 암호와 디지털 서명

# 1.1 앨리스의 차용서

- **차용서를 이메일로 보내면 어떨까?**
  - 메일을 누군가가 **변경**했을 수 있다
  - 처음부터 앨리스인 것처럼 **거짓 행세**를 한 누군가가 보낸 것인지도 모른다
  - 나중에 앨리스가 「그런 차용서 난 몰라」라고 부인할 수도 있다

# 1.2 메시지 인증코드에서 디지털 서명으로

- 메시지 인증 코드의 한계
  - 메시지 인증 코드를 사용하면 메시지의 변경과 거짓 행세를 검출할 수 있다
  - 메시지 인증 코드는 부인 방지에는 도움이 되지 않는다

# 1.2 메시지 인증코드에서 디지털 서명으로

- **디지털 서명을 이용한 해결**

  디지털 서명(digital signature)
  - 앨리스가 사용하는 키는 앨리스만이 알고 있는 개인적인 것
  - 앨리스는 메시지 송신 시에 그 개인적인 키를 써서 「서명」을 작성
  - 수신자 밥은 앨리스의 키와는 다른 키를 써서 「서명」을 검증

# 1.3 서명 작성과 서명 검증

- **메시지의 서명을 작성하는 행위**
  - 디지털 서명에서는 「서명용 키」와 「검증용 키」가 나누어짐
  - 검증용 키로 서명을 작성할 수는 없다

- **메시지의 서명을 검증하는 행위**
  - 「서명용 키」는 서명을 하는 사람만이 가지고 있지만,
  - 「검증용 키」는 서명을 검증하는 사람이라면 누구라도 가질 수 있다

# 공개키 암호와 디지털 서명

- 공개 키 암호
  - 「암호 키」와 「복호 키」가 나누어져 있어 암호 키로 복호화를 행할 수는 없다
  - 「복호 키」는 복호화를 행하는 사람만이 가지고 있지만, 「암호 키」는 암호화를 행하는 사람이라면 누구나 가질 수 있다

- 디지털 서명은 공개 키 암호를 「역으로 사용」함으로써 실현

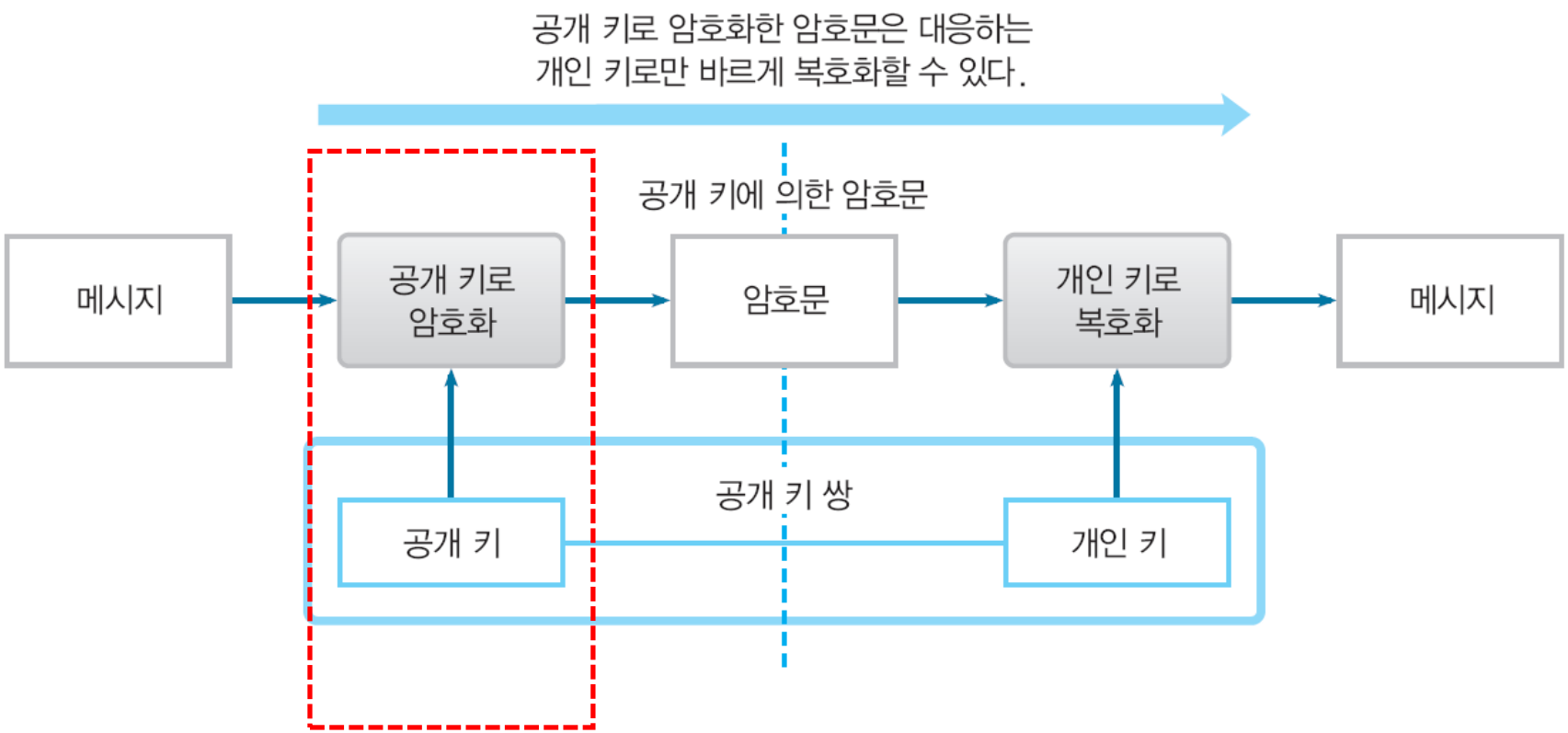
# 공개키 암호와 디지털 서명 키 사용방법

| | 개인 키 | 공개 키 |
|---|---|---|
| 공개 키 암호 | 수신자가 복호화에 사용 | 송신자들이 암호화에 사용 |
| 디지털 서명 | 서명자가 서명 작성에 사용 | 검증자들이 서명 검증에 사용 |
| 키는 누가 갖는가? | 개인이 갖는다. | 필요한 사람은 아무나 가지고 있어도 된다. |

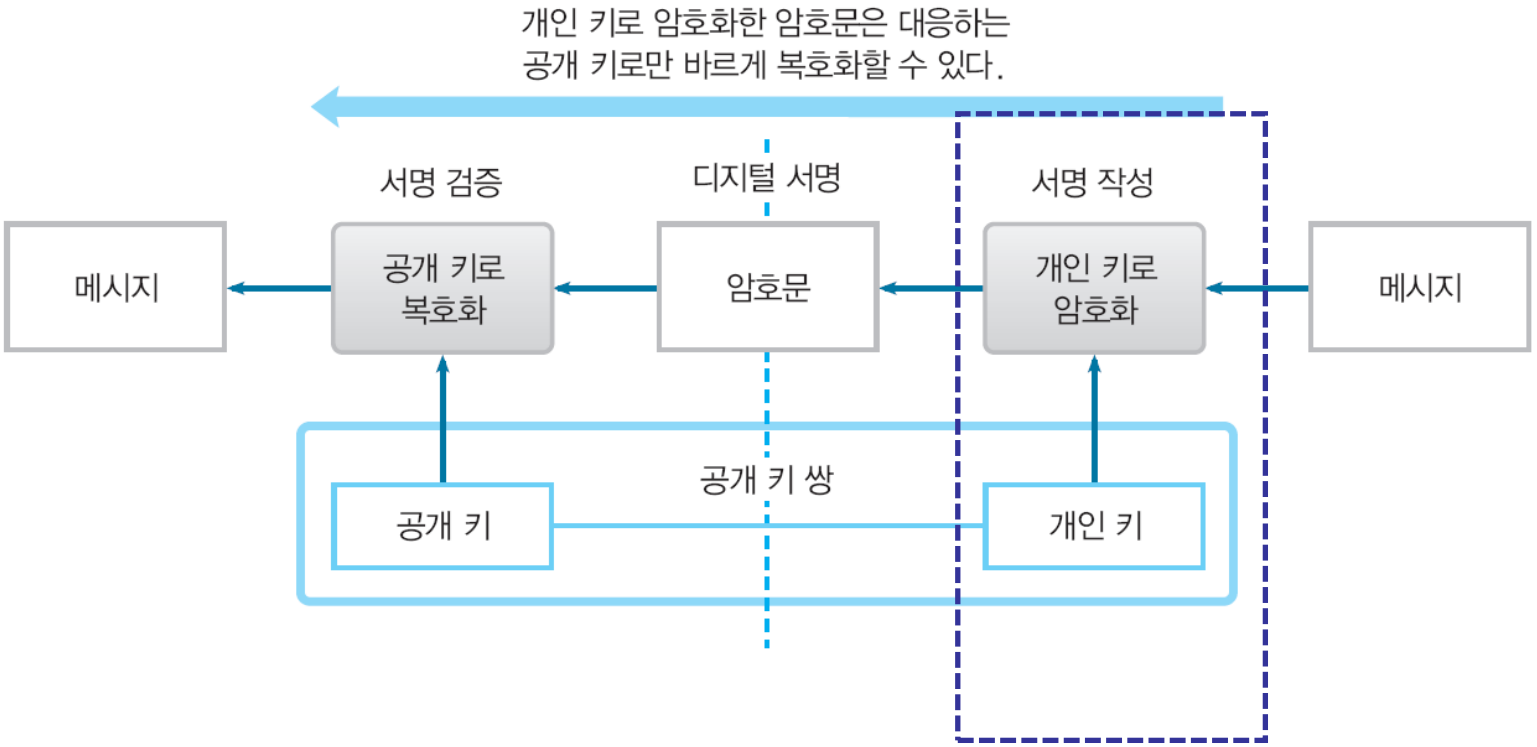
# 1.4 공개 키 암호와 디지털 서명

- 메시지를 개인 키로 암호화하는 것이 서명 작성에 해당
- 암호문을 공개 키로 복호화하는 것이 서명 검증에 해당

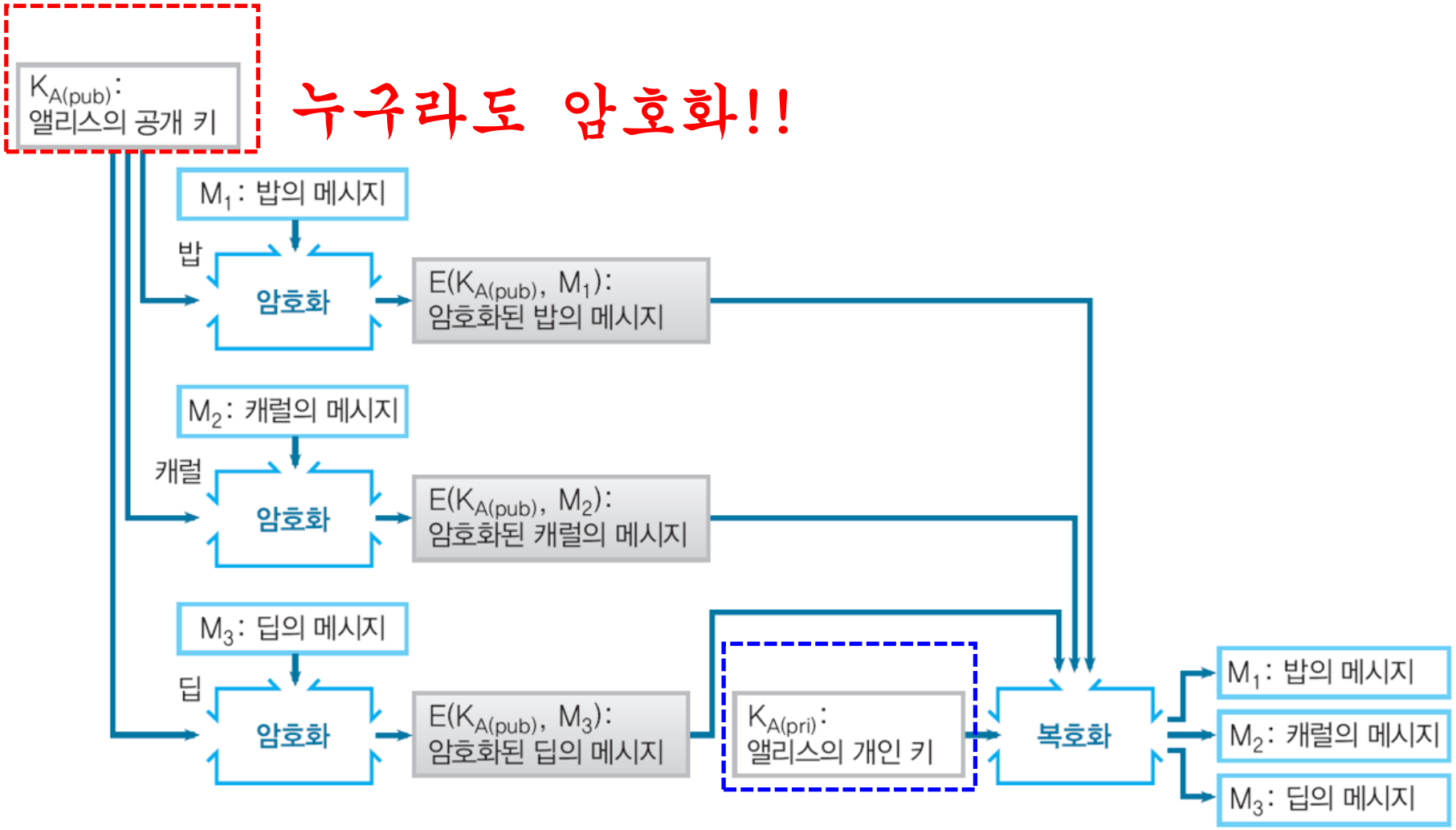
# 공개 키에 의한 암호화


공개 키로 암호화한 암호문은 대응하는
개인 키로만 바르게 복호화할 수 있다.
공개 키에 의한 암호문
메시지
공개 키로
암호화
암호문
개인 키로
복호화
메시지
공개 키 쌍
공개 키
개인 키

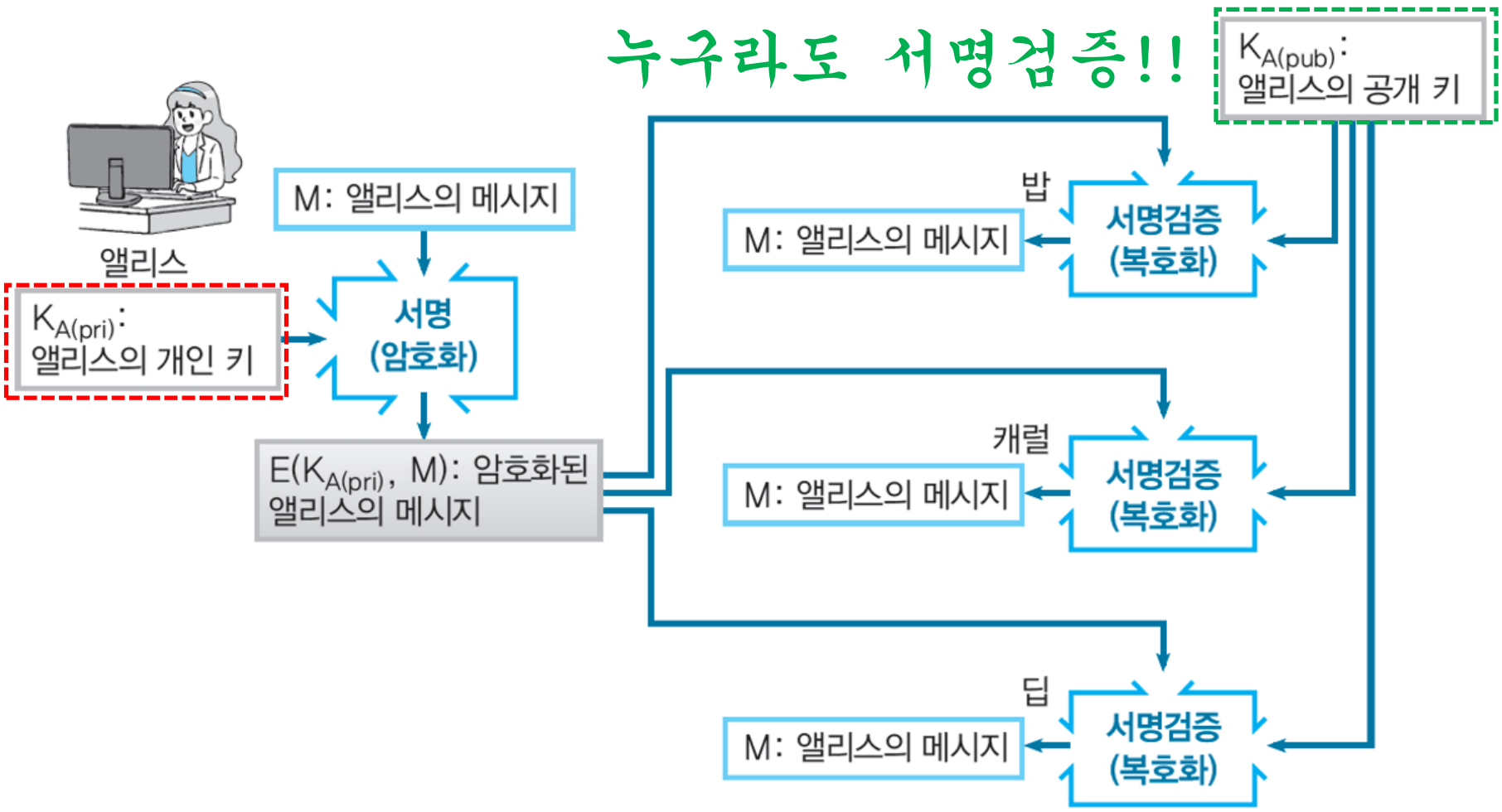
# 개인 키에 의한 암호화


개인 키로 암호화한 암호문은 대응하는
공개 키로만 바르게 복호화할 수 있다.
서명 검증
디지털 서명
서명 작성
메시지
공개 키로
복호화
암호문
개인 키로
암호화
메시지
공개 키 쌍
공개 키
개인 키

# 공개키 암호는 누구라도 암호화



# 디지털 서명은 누구라도 서명검증



# 제2절 디지털 서명 방법

## 2.1 디지털 서명 특징

## 2.2 디지털 서명 요구 조건

## 2.3 메시지에 직접 서명하는 방법

## 2.4 메시지의 해시 값에 서명하는 방법

# 2.1 디지털 서명 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

# 2.2 디지털 서명 요구 조건

- 서명은 메시지에 의존하는 비트 형태 이어야 한다
- 위조와 부인 방지 위해 송신자의 유일한 정보 비트를 이용해야 한다
- 서명문을 만들기가 쉬워야 한다
- 서명문을 인식 / 확인 하기가 쉬워야 한다
- 서명문을 위조하는 것이 계산적으로 실행 불가능해야 한다

서명 방식 2종류 = 2.3 , 2.4

# 2.3 메시지에 직접 서명하는 방법

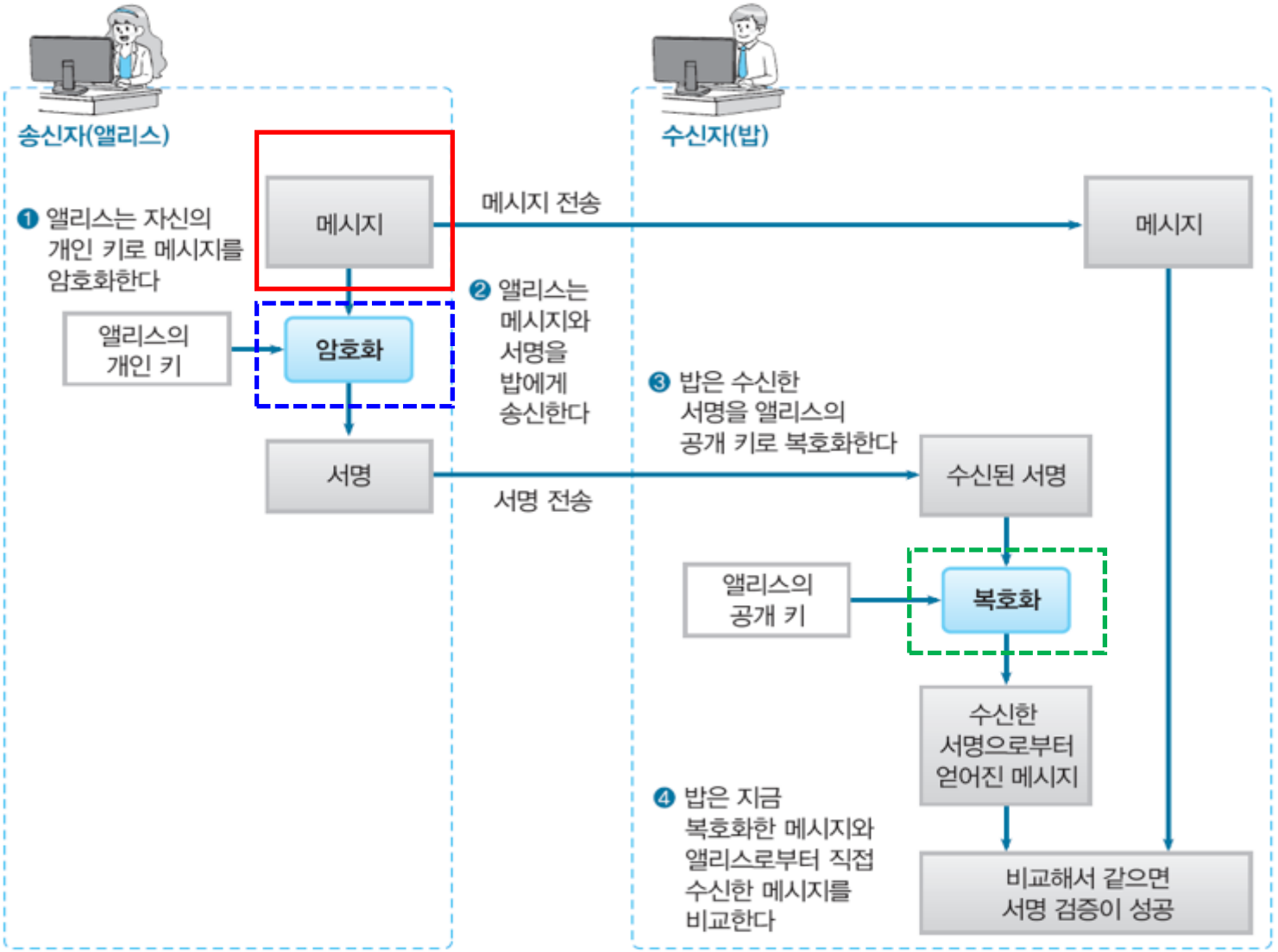
1. 앨리스는 자신의 개인 키로 메시지를 암호화한다.

2. 앨리스는 메시지와 서명을 밥에게 송신한다.

3. 밥은 수신한 서명을 앨리스의 공개 키로 복호화한다.

4. 밥은 이제 서명을 복호화해서 얻어진 메시지와 앨리스로부터 직접 수신한 메시지를 비교한다.

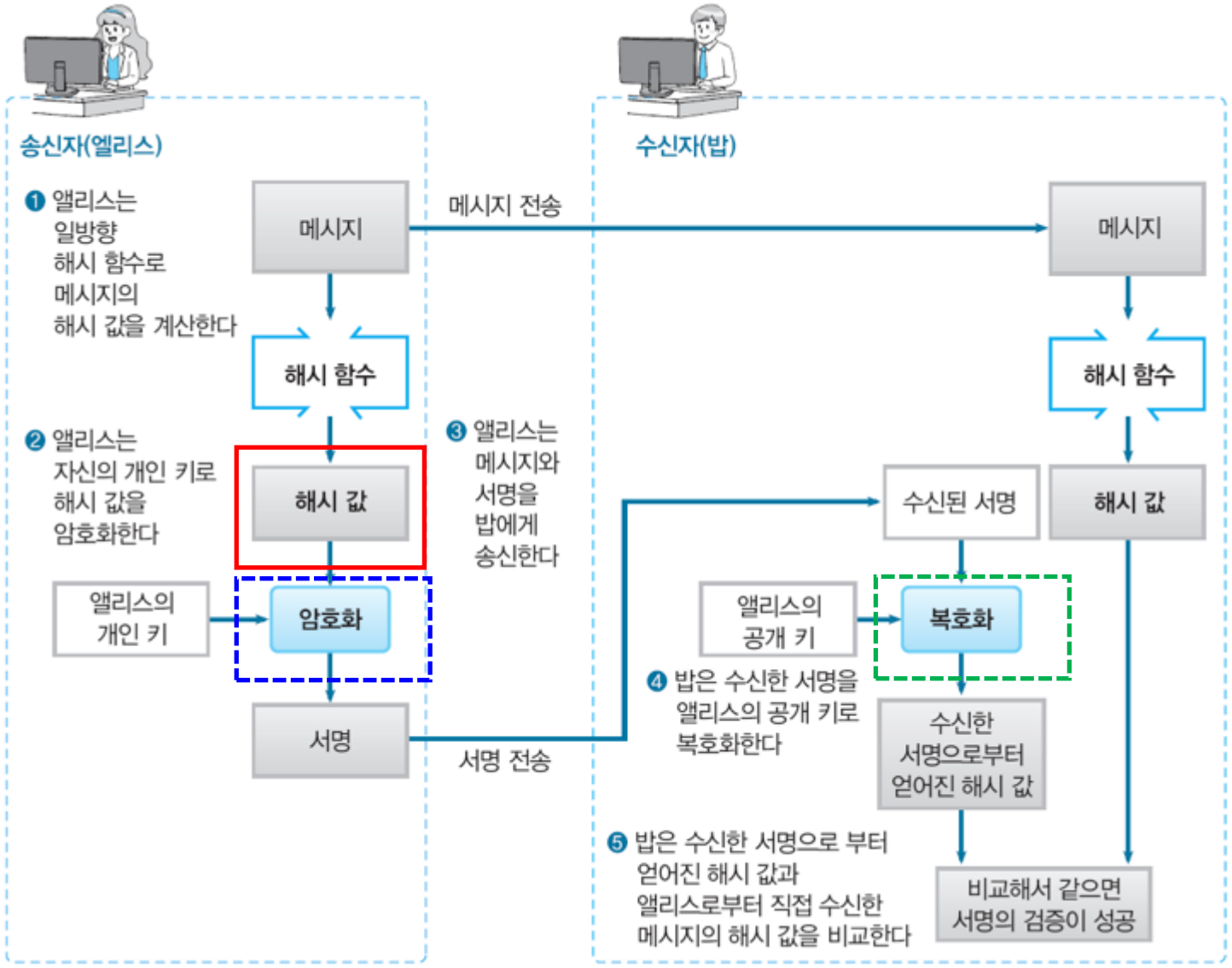
# 앨리스가 메시지에 서명하고 밥이 서명 검증



# 2.4 메시지의 해시 값에 서명하는 방법

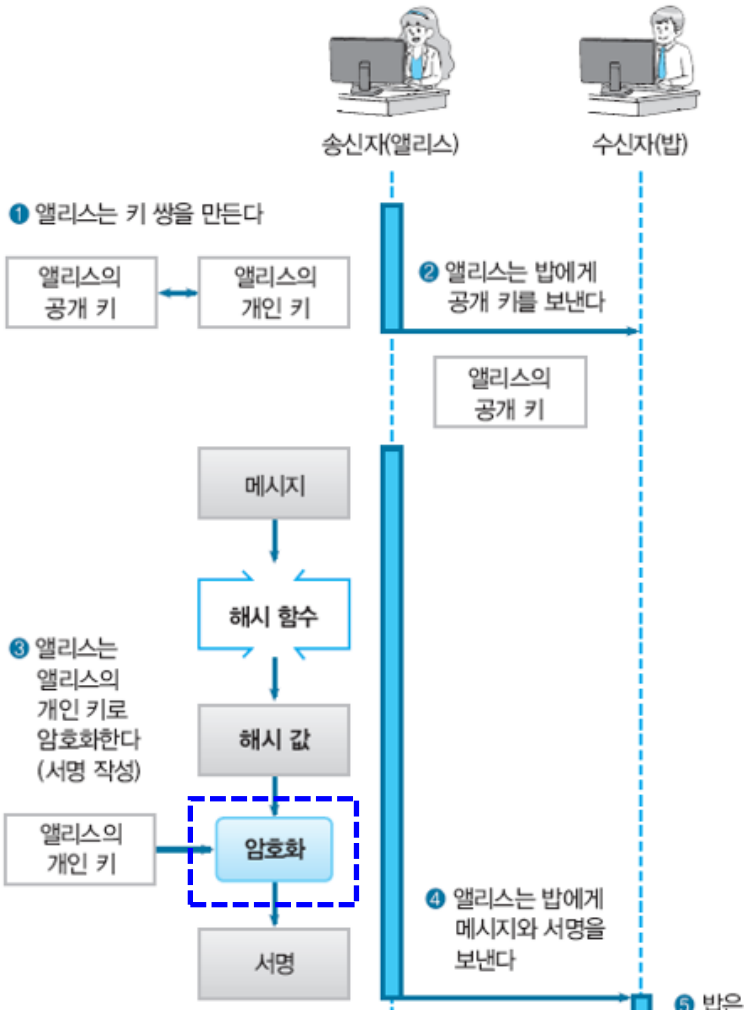
1. 앨리스는 일방향 해시 함수로 메시지의 해시 값을 계산한다.
2. 앨리스는 자신의 개인 키로 해시 값을 암호화한다.
3. 앨리스는 메시지와 서명을 밥에게 송신한다.
4. 밥은 수신한 서명을 앨리스의 공개 키로 복호화한다.
5. 밥은 수신한 서명으로부터 얻어진 해시 값과 앨리스로부터 직접 수신한 메시지의 해시 값을 비교한다.

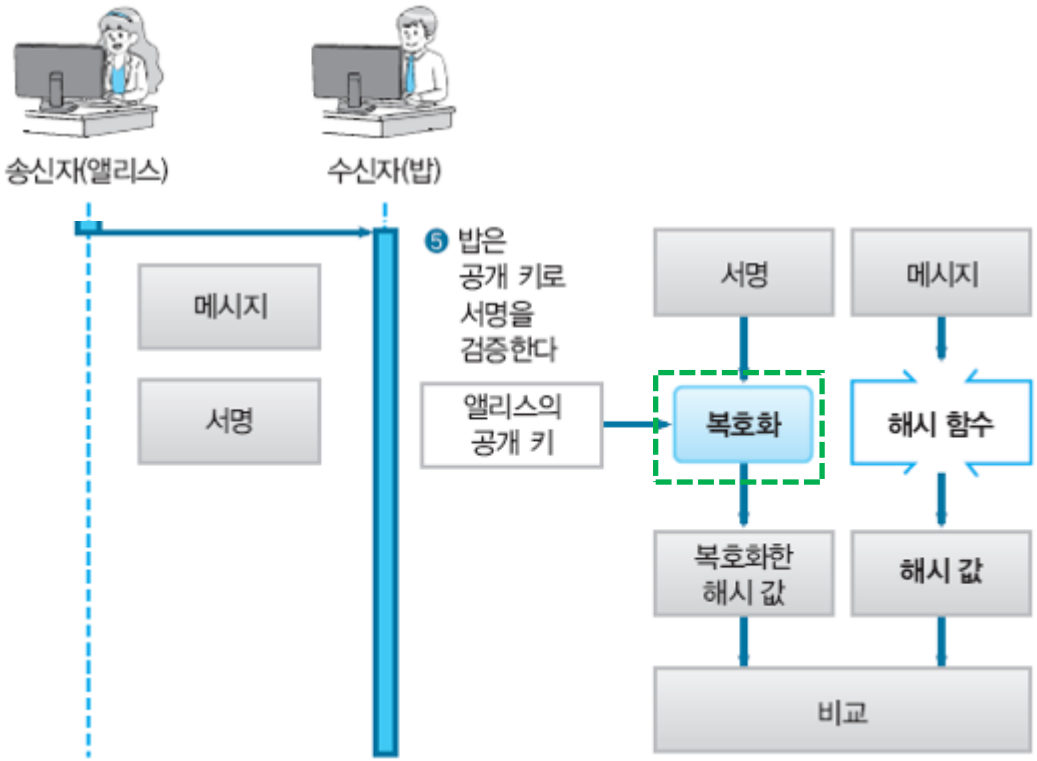
# 앨리스가 메시지의 해시 값에 서명하고 밥이 서명 검증



# 앨리스가 메시지의 해시 값에  서명하고 밥이 서명 검증

- 시간적 흐름




송신자(앨리스)
수신자(밥)
메시지
서명
⑤ 밥은 공개 키로 서명을 검증한다
앨리스의 공개 키
복호화
서명
메시지
해시 함수
복호화한 해시 값
해시 값
비교

# 제3절 디지털 서명에 대한 의문

3.1 암호문이 왜 서명으로서 사용 가능한 것인가?

3.2 기밀성을 유지할 수 없는 것은 아닐까?

3.3 복사된 서명이 만들어지는 것은 아닐까?

3.4 서명 변경이 가능한 것은 아닐까?

3.5 서명만 재이용할 수 있는 것은 아닐까?

3.6 서명을 삭제하더라도 계약파기를 할 수 없는 것은 아닌가?

3.7 어떻게 해서 부인 방지가 되는 것인가?

3.8 디지털 서명은 정말로 종이 서명 대용이 되는 것일까?

# 3.1 암호문이 왜 서명으로서 사용 가능한 것인가?

- 개인 키로 암호화한다는 것
  - 행하고 있는 처리의 내용을 설명한 것
  - 여기에서는 기밀성을 실현하기 위해 암호화하고 있는 것은 아님

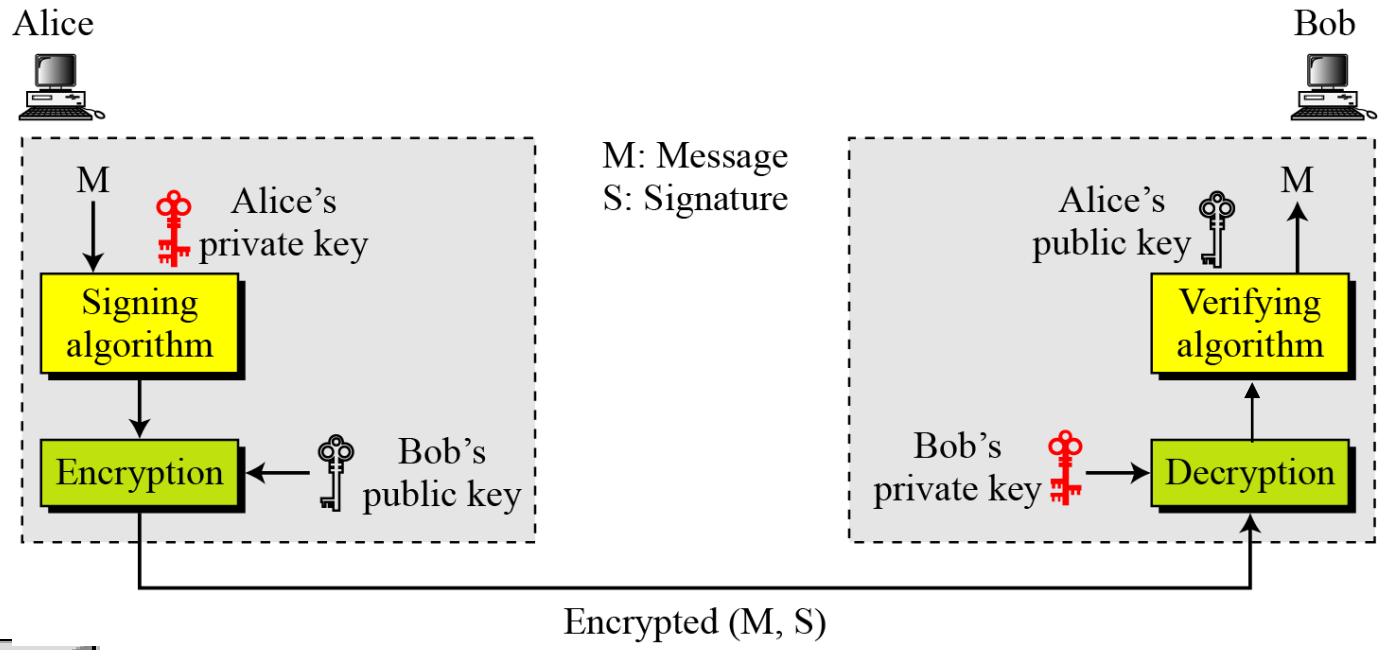
- 인증자(authenticator)
  - 키를 가지고 있는 사람만이 만들 수 있는 정보

# 3.2 기밀성을 유지할 수 없는 것은 아닐까?

- 디지털 서명은 기밀성을 지키기 위한 것은 아님

- 만약 기밀성이 필요하다면
  - 메시지를 그대로 보내는 것이 아니고, 암호화를 별도 실행

- 디지털 서명 구조에 기밀성 추가하기




**Note**

- 디지털 서명은 프라이버시를 보장해주지 못함
- 프라이버시가 필요하다면 암호화/복호화를 할 수 있는 또다른 수단이 적용되어야함

# 3.3 복사된 서명이 만들어지는 것은 아닐까?

- 통상의 파일 복사처럼 서명도 복사본을  간단히 만들 수 있음

- 하지만, 서명 복사를 만들 수 있다고 해서 서명이 무의미해지는 것은 아님
  - 왜냐 하면 복사한 데이터가 표현하고 있는 것은 「특정의 서명자가 특정의 메시지에 대해서 서명했다」고 하는 것뿐이기 때문
  - 즉, 복사해도 서명자는 바뀌지 않고 메시지의 내용도 바뀌지 않음

# 복사된 서명

- 특정 서명자와 특정 메시지가 결부되어 있다는 사실이 중요
- 아무리 복사를 해도 「그 메시지에 누가 서명했는가」하는 사실에는 조금도 변화가 없음
- 복사는 할 수 있다. 그러나 그것에 의해 서명이 무의미해지는 것은 아님

# 3.4 서명 변경이 가능한 것은 아닐까?

- **확실히 서명한 후에 메시지와 서명의 내용을 수정 가능**
  - 그러나 수정해 버리면 서명의 검증에 실패
  - (검증하는 사람은 수정이 행해졌다는 것을 검출할 수 있음)

# 또 다른 의문

- 서명 대상의 메시지와 서명 양쪽을 수정해서 서명의 검증에 성공할 수 있도록 앞뒤를 잘 맞출 수 있지는 않을까?

- 아니다. 그것은 사실상 불가능
  (암호화 시스템의 안전성을 깨는것과 같이 어려움)

# 3.5 서명만 재이용할 수 있는 것은 아닐까?

- 확실히 서명 부분만을 잘라내서 다른 메시지에 첨부하는 것은 가능

- 그러나 서명의 검증에는 실패
  (서명은 특정 서명자와 해당 메시지에 연관되기 때문에 다른메시지에서는 그서명의 검증이 안됨)

# 3.6 서명을 삭제하더라도 계약파기를 할 수 없는 것은 아닌가?

- 분명히 디지털 서명이 붙은 차용서는 삭제해도 파기할 수 없다.

- 디지털 서명이 붙은 차용서를 파기하는 경우
  - 차용서의 내용이 계약파기의 새로운 내용로 변경되어야 하며, 그것에 대해 상대의 디지털 서명이 되어있어야 함

# 3.7 어떻게 해서 부인 방지가 되는 것인가?

- 디지털 서명의 경우 서명을 작성할 수 있는 키(개인 키)는 송신자만 가지고 있음
  - 그러므로 서명을 작성할 수 있는 것은 송신자뿐

- 그렇기 때문에 송신자는 「그 서명을 작성한 것은 내가 아니다」라고 주장할 수가 없음

# 3.8 디지털 서명은 정말로 종이 서명 대용이 되는 것일까?

- 한국에서는 1999년 전자서명법이 제정, 시행
- 전자서명법, [법률 제18479호, 2021. 10. 19., 일부개정]

> 제1조(목적) 이 법은 전자문서의 안전성과 신뢰성을 확보하고 그 이용을 활성화하기 위하여 전자서명에 관한 기본적인 사항을 정함으로써 국가와 사회의 정보화를 촉진하고 국민생활의 편익을 증진함을 목적으로 한다.

- 실제는 디지털 서명에 관한 분쟁이 발생하여 디지털 서명의 유효성을 둘러싸고 재판이 일어날 가능성은 있음

> **제22조(분쟁의 조정)** 전자서명에 관한 분쟁의 조정을 받으려는 자는 「전자문서 및 전자거래 기본법」 제32조에 따른 전자문서·전자거래 분쟁조정위원회에 조정을 신청할 수 있다.

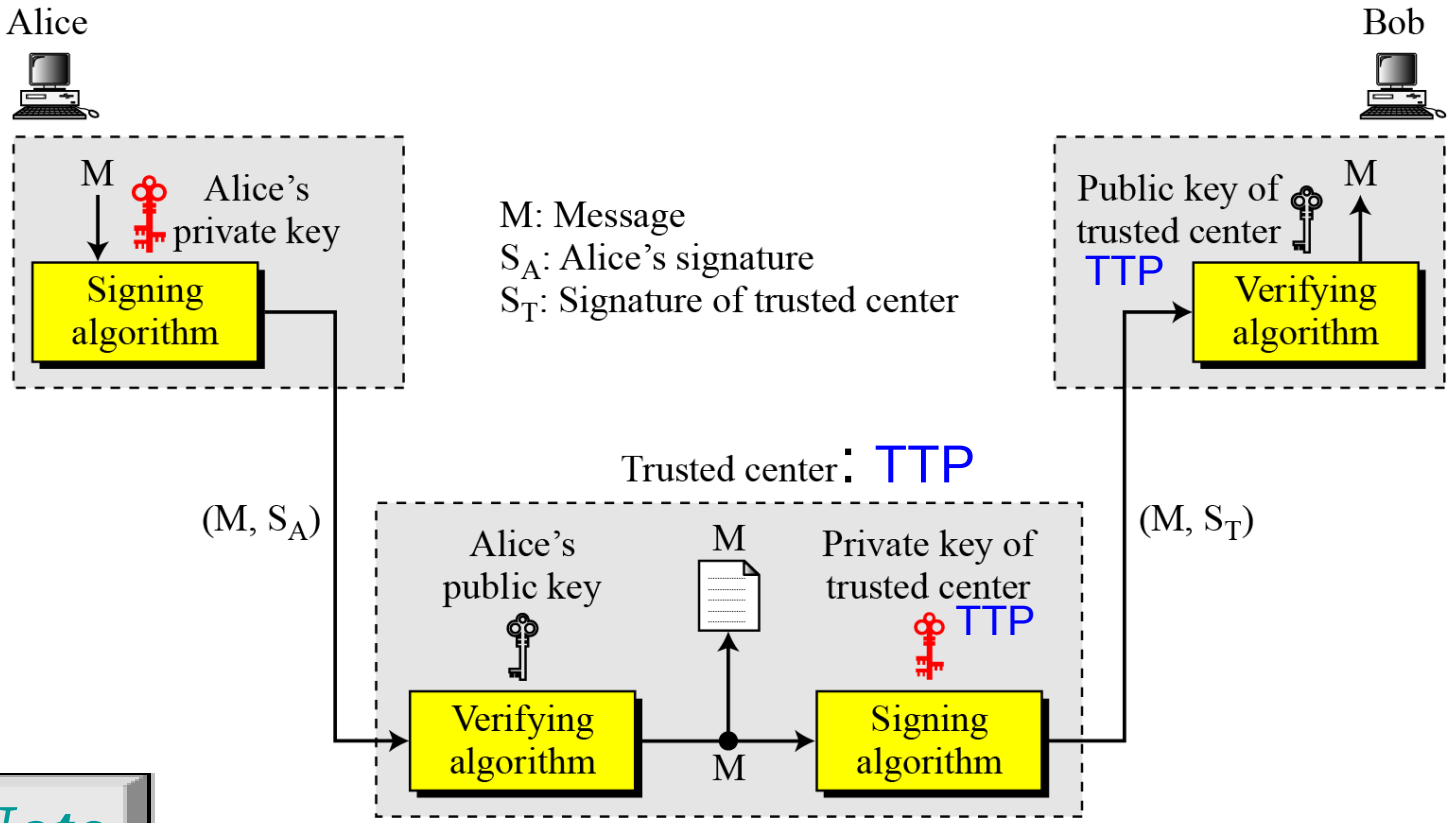
# [전자서명법]

제2조(정의) 이 법에서 사용하는 용어의 뜻은 다음과 같다.

1. "전자문서"란 정보처리시스템에 의하여 전자적 형태로 작성되어 송신 또는 수신되거나 저장된 정보를 말한다.
2. "전자서명"이란 다음 각 목의 사항을 나타내는 데 이용하기 위하여 전자문서에 첨부되거나 논리적으로 결합된 전자적 형태의 정보를 말한다.
   가. 서명자의 신원
   나. 서명자가 해당 전자문서에 서명하였다는 사실
3. "전자서명생성정보"란 전자서명을 생성하기 위하여 이용하는 전자적 정보를 말한다.
4. "전자서명수단"이란 전자서명을 하기 위하여 이용하는 전자적 수단을 말한다.
5. "전자서명인증"이란 전자서명생성정보가 가입자에게 유일하게 속한다는 사실을 확인하고 이를 증명하는 행위를 말한다.
6. "인증서"란 전자서명생성정보가 가입자에게 유일하게 속한다는 사실 등을 확인하고 이를 증명하는 전자적 정보를 말한다.
7. "전자서명인증업무"란 전자서명인증, 전자서명인증 관련 기록의 관리 등 전자서명인증서비스를 제공하는 업무를 말한다.
8. "전자서명인증사업자"란 전자서명인증업무를 하는 자를 말한다.
9. "가입자"란 전자서명생성정보에 대하여 전자서명인증사업자로부터 전자서명인증을 받은 자를 말한다.
10. "이용자"란 전자서명인증사업자가 제공하는 전자서명인증서비스를 이용하는 자를 말한다.

# 디지털 서명 서비스 모델

- 부인 방지를 위한 신뢰받는 센터(TTP)의 활용




**Note**

제3의 신뢰자 **(TTP: Trusted Third Party)** 를 이용하면 부인봉쇄를 할 수 있음

# 제4절 디지털 서명 활용 예

## 4.1 보안 공지

## 4.2 소프트웨어 다운로드

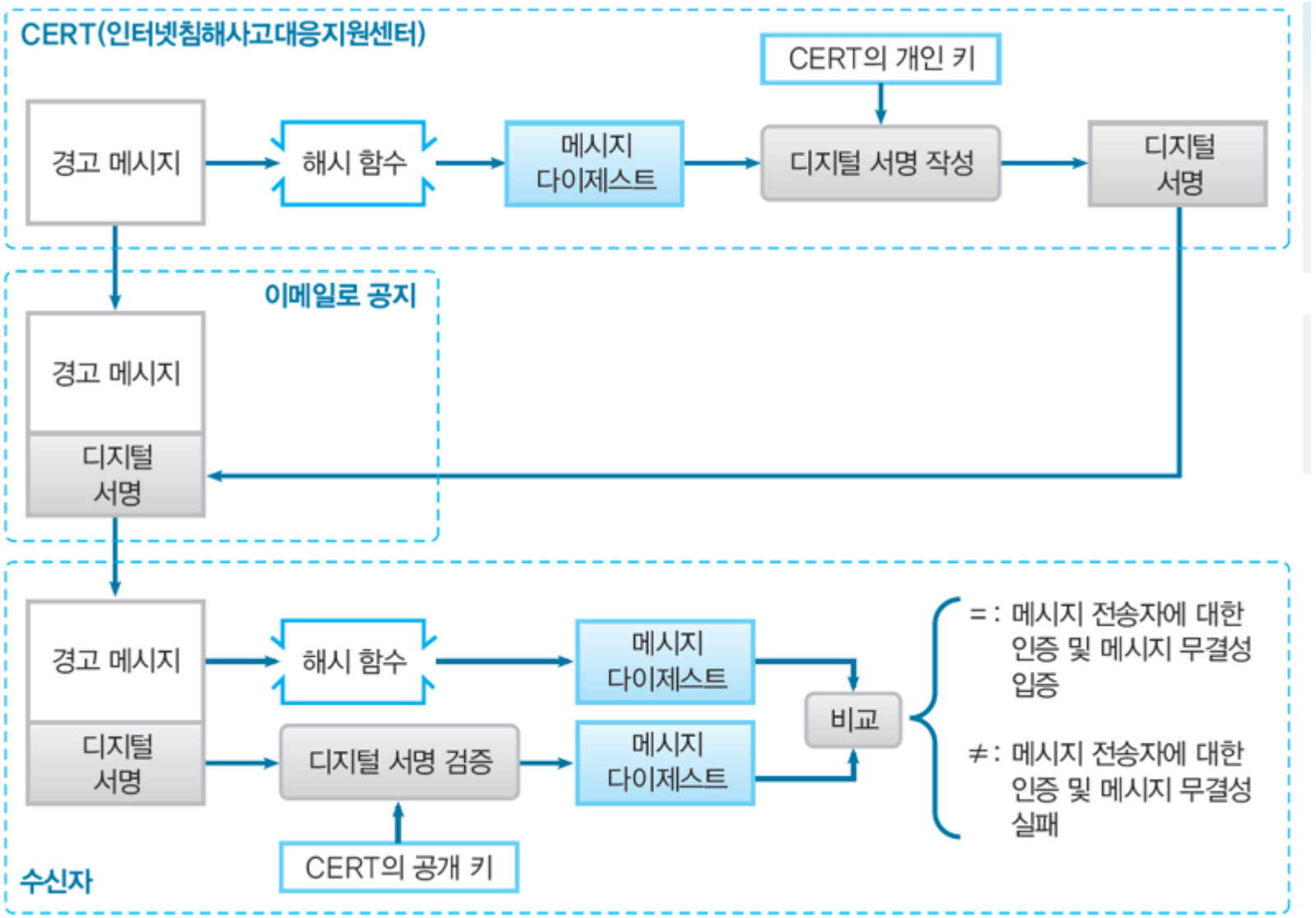
## 4.3 공개 키 인증서

## 4.4 SSL/TLS

# 4.1 보안 공지

- 클리어서명(clearsign)
  - 메시지를 암호화하지 않고 서명만 한 것

# 경고 메시지에 대한 디지털 서명 활용



# 4.2 소프트웨어 다운로드

- 소프트웨어의 작성자가 소프트웨어에 디지털 서명을 작성
- 소프트웨어 사용자가 다운로드한 후에 서명을 검증
  - 적극적 공격자 맬로리에 의한 내용 조작을 검출하는 것이 가능

# 4.3 공개 키 인증서

- 디지털 서명을 검증하려면 바른 공개키가 필요
- 자신이 입수한 공개 키가 바른 공개 키인지 어떤지를 검증하기 위해서 공개키를 메시지로 간주하고 그것에 디지털 서명을 한 것
- 공개키 인증서
  - 공개키에 디지털 서명을 붙인 것

# 4.4 SSL/TLS

- SSL/TLS에서는 서버가 올바른 것이라는 것을 인증하기 위해서 서버 인증서를 이용

- 이것은 서버의 공개키에 디지털 서명을 한 것

# 제5절 RSA에 의한 디지털 서명

## 5.1 RSA에 의한 서명 작성

## 5.2 RSA에 의한 서명 검증

## 5.3 자세한 RSA 서명

# 5.1 RSA에 의한 서명 작성

- 서명 = $(\text{메시지})^{D} \bmod N$
- D와 N은 서명자의 개인 키

# 5.2 RSA에 의한 서명 검증

- 서명으로부터 얻어진 메시지 = $(\text{서명})^{E} \bmod N$
- E와 N은 서명자의 공개 키

# RSA 서명 작성과 검증

| | | |
|---|---|---|
| 키 쌍 | 공개 키 | (E, N) |
| | 개인 키 | (D, N) |
| 서명의 작성 | | 서명 = (메시지)$^{D}$ mod N<br>(메시지를 D 제곱해서 N으로 나눈 나머지) |
| 서명의 검증 | | 서명으로부터 얻어진 메시지 = (서명)$^{E}$ mod N<br>(서명을 E제곱해서 N으로 나눈 나머지)<br>「서명으로부터 얻어진 메시지」와 「메시지를 비교한다 |

# 5.3 자세한 RSA 서명

- 공개 키:E = 5, N = 323
- 개인 키:D = 29,N = 323
- N이 323이므로 메시지는 0 ~ 322 범위의 정수에서 고른다.
- 여기서는 123이라는 메시지에 서명을 해 보자.

# 서명 작성

- 메시지$^D$ mod N = $123^{29}$ mod 323= 157
- 서명은 157
- 수신자에게 전달할 것
  - (메시지, 서명) = (123, 157)

# 서명 검증

- (메시지, 서명) = (123, 157) 수신
- 공개 키(E, N) = (5, 323)을 사용해서 서명으로부터 얻어진 메시지를 계산
- $서명^{E} \bmod N = 157^{5} \bmod 323 = 123$
- 이 메시지 123은 분명히 송신자가 보낸 메시지 123과 일치
- 서명 검증에 성공

# 제6절 다른 디지털 서명

6.1 ElGamal 디지털 서명

6.2 Schnorr 디지털 서명

6.3 DSA

6.4 ECDSA

6.5 Rabin 방식

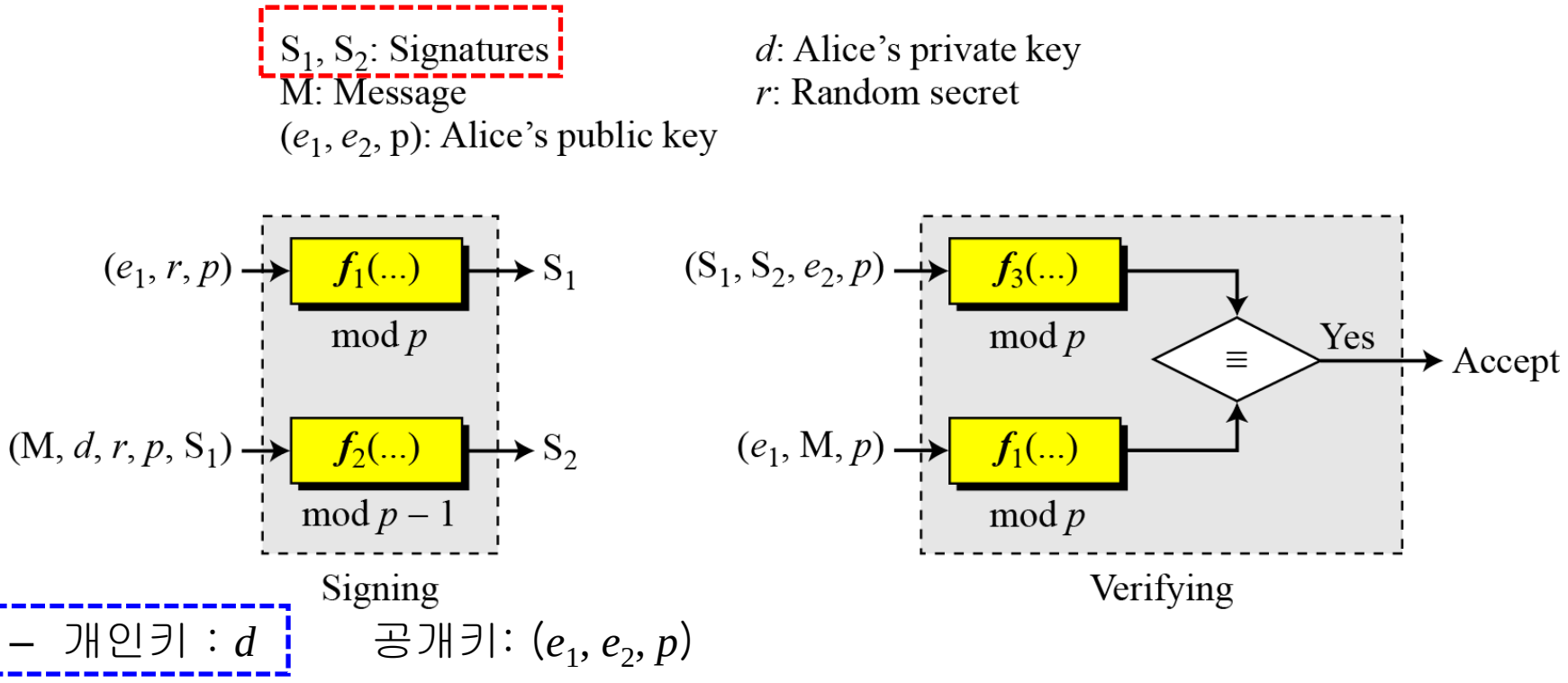
6.6 전자서명의 표준 (DSS)

6.7 기타 디지털 서명

# 6.1 ElGamal 디지털 서명

- ElGamal 방식은 Taher ElGamal에 의한 공개키 알고리즘으로 mod N 으로 이산대수를 구하는 것이 곤란하다는 것을 이용
- ElGamal 방식은 공개키 암호화와 디지털 서명에 이용
- 암호 소프트웨어 GnuPG에서도 알고리즘의 하나로 사용

- ElGamal 디지털 서명 구조에 대한 일반적 아이디어

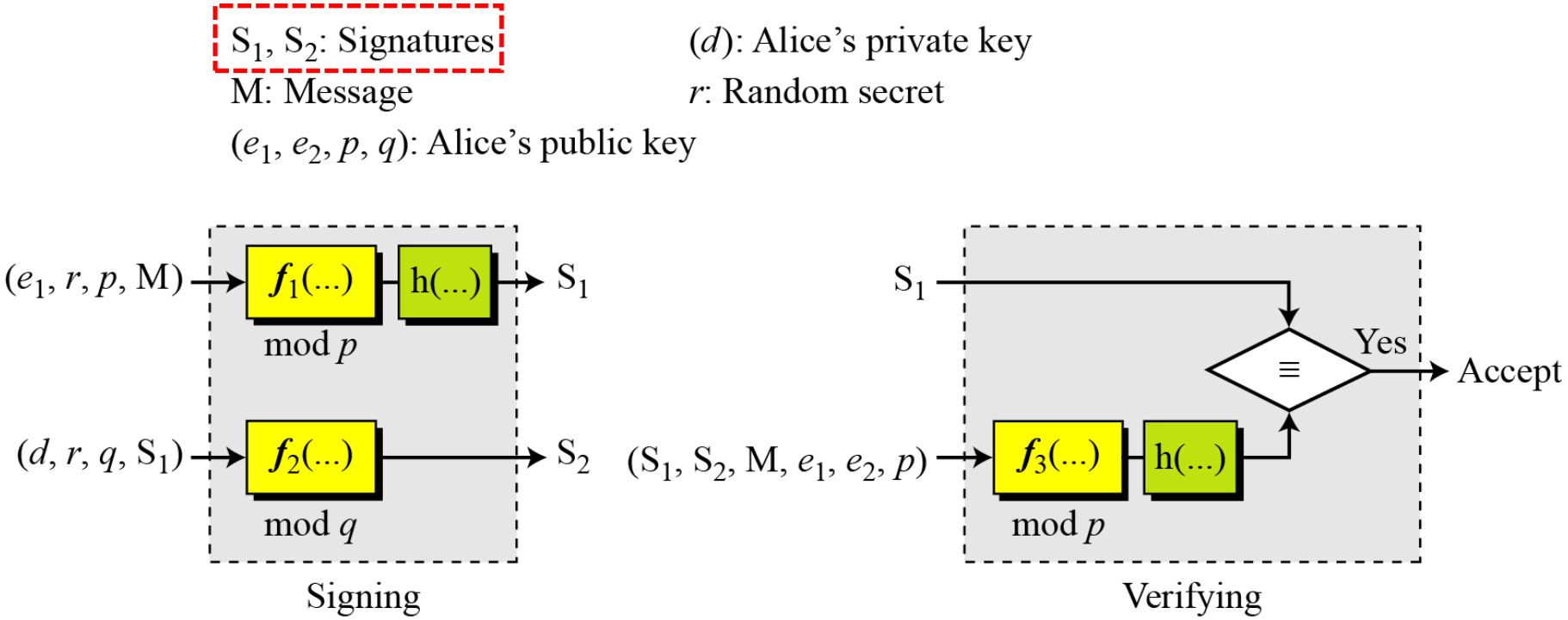



– 개인키 : $d$

공개키: $(e_1, e_2, p)$

# 6.2 Schnorr 디지털 서명

- Schnorr 디지털 서명 구조에 대한 일반적 아이디어




– 개인키 : $d$ 공개키: $(e_1, e_2, p, q)$

# 6.3 DSA

- DSA(Digital Signature Algorithm)
  - 디지털 서명 알고리즘의 일종
  - NIST(National Institute of Standards and Technology)가 1991년에 제정한 디지털 서명 규격(DSS)

- Schnorr의 알고리즘과 ElGamal 방식의 변형으로 디지털 서명에만 이용

# DSA 키 생성

**전역적 공개키 (Global Public Key) 구성요소**

p $512 \le L \le 1024$에 대해 $2^{L-1} < p < 2^L$을 만족하는 소수로 $L$은 64의 배수; 즉, 512와 1024비트 사이에서 64 비트씩 증가하는 비트 길이

q $2^{159} < q < 2^{160}$를 만족하는 $(p-1)$의 소수인 제수 (divisor)
즉, 160 비트 길이

g $= h^{(p-1)/q} \bmod p$ 여기서, $h$는 $h^{(p-1)/q} \bmod p > 1$을 만족하는 $1 < h < (p-1)$ 인 임의의 정수

**사용자 개인키**

x $0 < x < q$인 난수 또는 의사 난수 정수

**사용자 공개키**

y $= g^x \bmod p$

# DSA 서명 생성 및 검증

[서명 생성]

- r = ($g^k$ mod $p$) mod $q$
- $s$ = [$k^{-1}$(H($M$)+$xr$)] mod $q$
- Signature = $(r, s)$

메시지 $M$과 $(r,s)$를 전송

[서명 검증]

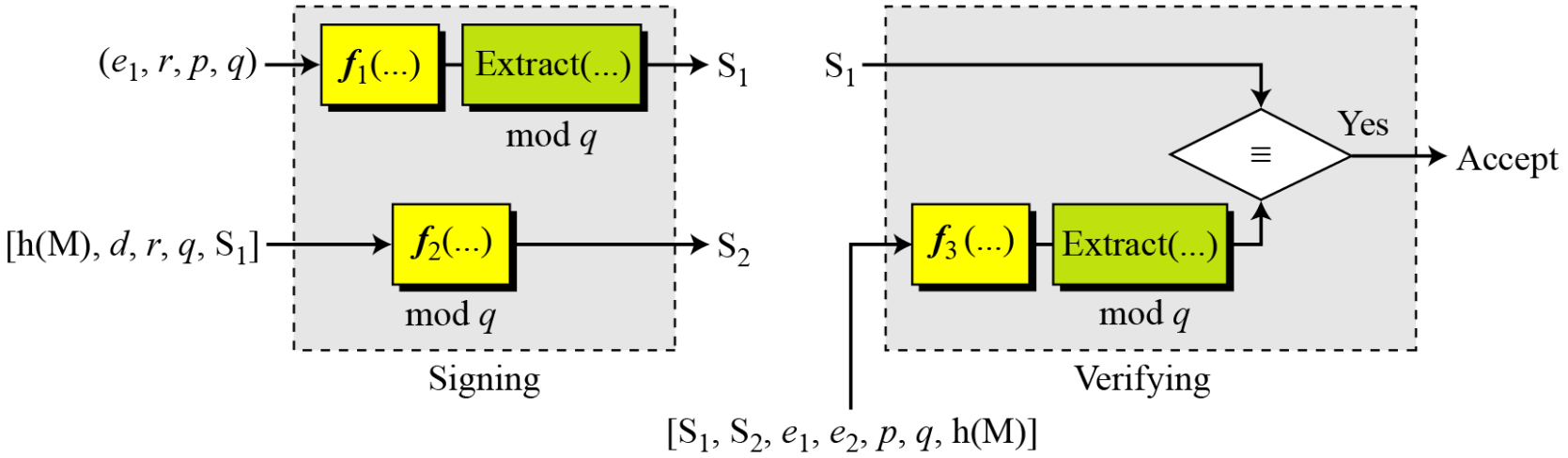
- $w$ = ($s'$)$^{-1}$mod $q$
- u1 = [(H(M')$w$] mod $q$
- u2 = ($r'$)$w$ mod $q$
- $v$ = [($g^{u1}y^{u2}$)mod $p$] mod $q$
- $v$=$r'$ 인지 검증

# 6.4 ECDSA

- ECDSA(Elliptic Curve Digital Signature Algorithm)
- 타원 곡선 암호(ECC: Elliptic Curve Cipher)를 사용한 디지털 서명알고리즘

- ECDSS 구조에 대한 일반적 아이디어

$S_1$, $S_2$: Signatures
M: Message
$(a, b, p, q, e_1, e_2)$: Alice's public key
$d$: Alice's private key
$r$: Random secret



# 6.5 Rabin 방식

- Rabin 방식
  M. O. Rabin에 의한 공개 키 알고리즘으로 mod N으로 제곱근을 구하는 것이 곤란하다는 것을 이용

- Rabin 방식
  공개키 암호화와 디지털 서명에 이용

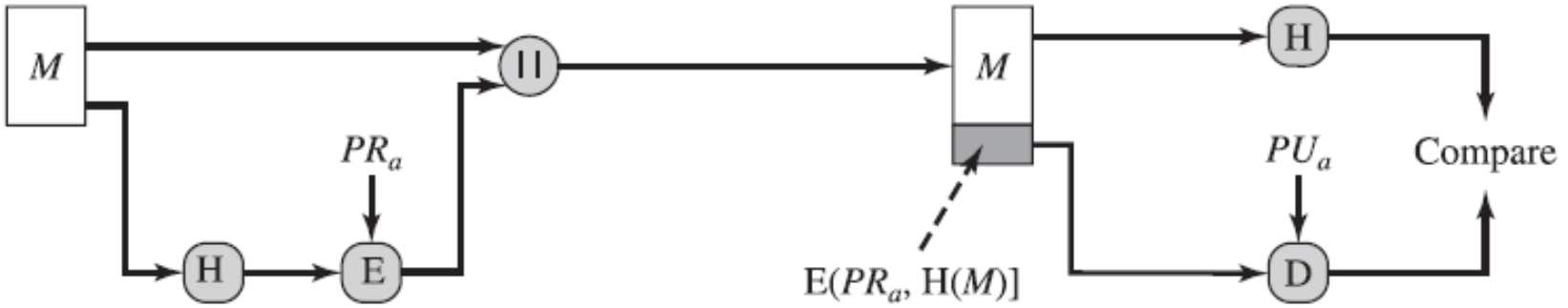
# 6.6 전자서명의 표준 (DSS)

- **전자서명표준**
  - DSS (Digital Signature Standard)
  - NIST (National Institute of Standards and Technology) 가 제안한 것으로 FIPS PUB 186임
  - SHA 이용

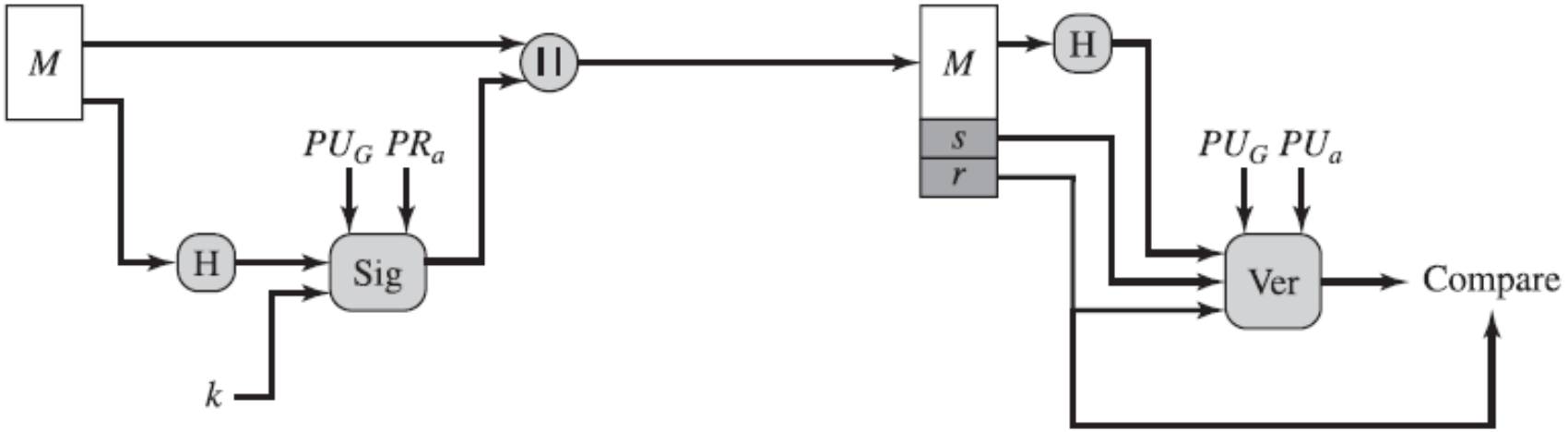
- **전자서명알고리즘**
  - DSA (Digital Signature Algorithm)
  - DSS에서 이용되는 알고리즘

- 1991년 제안, 1993년 개정, 보조적 추가 개정 1996년
- 2000년 확장 버전이 FIPS 186-2로 발표됨
- 2009년 FIPS 186-3으로 개정됨
  - RSA 및 타원곡선암호학 (ECC)에 기반한 전자서명 알고리즘을 포함

# 전자서명의 두가지 접근 방법

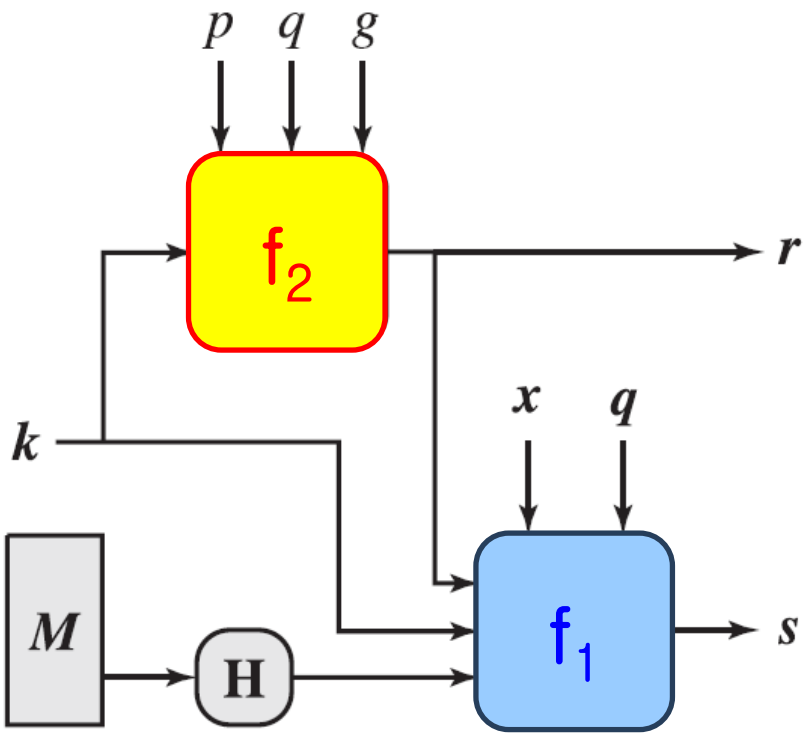



(a) RSA 방법




(b) DSS 방법

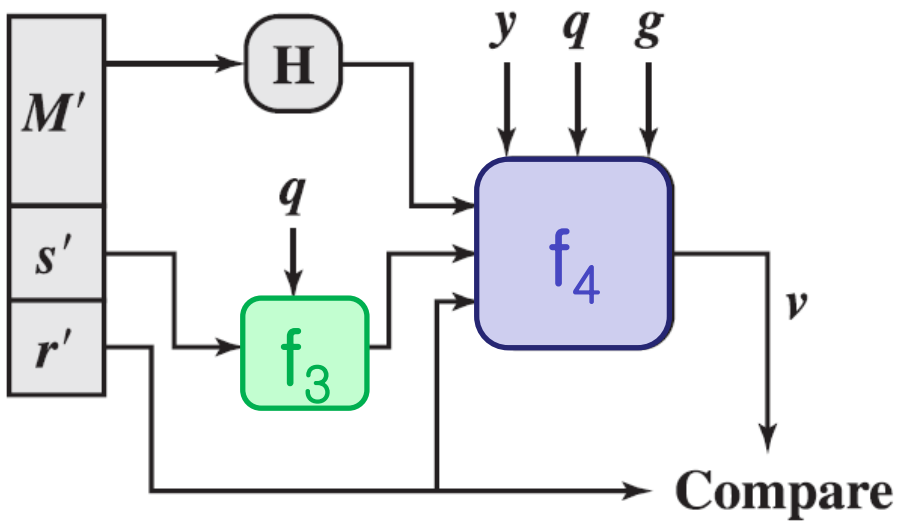
# DSS 서명과 검증




$s = f_1(H(M), k, x, r, q) = (k^{-1} (H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

(a) 서명




$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q} y^{r'w \bmod q}) \bmod p) \bmod q$

(b) 확인

# 6.7 기타 디지털 서명

- RSA : Rivest & Shamir & Adleman 제안, 1978
- Schnorr : 1989년 Schnorr가 개인식별방식과 함께 제안
- GQ : 1998년 Guillou 와 Quisquater가 제안
- SIGN : 1991년 후지오까, 오까모도, 미야구찌가 제안
- Fiat Shamir : Fiat, Shamir, 1987
- KCDSA : 1999, TTA표준
- EC-KCDSA : 2001, KCDSA의 변형, TTA표준

# 제7절 디지털 서명에 대한 공격

**7.1 중간자 공격**

**7.2 일방향 해시 함수에 대한 공격**

**7.3 디지털 서명을 사용한 공개 키 암호 공격**

**7.4 잠재적 위조**

**7.5 기타 공격**

# 7.1 중간자 공격

- **중간자 공격**
  - 디지털 서명에도 위협이 되는 공격
  - 중간자 공격을 막으려면 입수한 공개 키가 정확한 상대의 것인지 아닌지를 확인하는 것이 필요

- **핑거프린트(fingerprint)**
  - 공개 키를 취급하는 소프트웨어는 공개 키의 해시 값을 표시하는 수단을 준비
  - 이 해시 값 ➔ 핑거프린트

# 7.2 일방향 해시 함수에 대한 공격

- 디지털 서명에서 사용하는 일방향 해시 함수는 충돌내성을 가져야만 한다
- 만약 충돌내성이 없으면 디지털 서명을 한 해시 값과 같은 해시 값을 갖는, 다른 메시지를 만들 수 있다

# 7.3 디지털 서명을 사용한 공개 키 암호 공격

- 공격자의 교묘한 속임수

밥 씨, 안녕하세요.

저는 암호기술을 연구하고 있는 맬로리라고 합니다.
저는 지금 디지털 서명에 관한 실험을 하고 있는데,
첨부된 데이터에 당신의 서명을 붙여서 회신해 주시면 감사하겠습니다.
첨부된 데이터는 랜덤한 데이터(**사실은 도청한 암호문**)이기 때문에 문제는 발생하지 않습니다.
협조해 주셔서 감사합니다.

From 맬로리

# 밥의 어리석은 행동

- 밥은 맬로리한테 온 메일을 읽고 첨부 데이터를 보니 확실히 랜덤한 데이터인 것 같다고 여긴다
  - 하지만 실은 이것은 앨리스가 밥의 공개 키로 암호화한 암호문

- 밥은 순수한 마음으로 첨부 데이터에 서명을 한다

# 맬로리의 목적 달성

서명 = (첨부 데이터)$^{D}$ mod N (RSA의 서명 작성)

= (암호문)$^{D}$ mod N (첨부 데이터는 실은 암호문이기 때문에)

= (메시지)$^{E}$)$^{D}$ mod N (복호화 처리가 행하여졌기 때문에)

= 메시지

- 의미를 모르는 메시지에는 절대로 디지털 서명을 하지 말아야한다

# 7.4 잠재적 위조

- **개인 키가 없는 공격자가 의미가 있는 메시지를 만들고, 그에 대한 바른 디지털 서명을 만들 수 있다면**
  - 그 디지털 서명 알고리즘은 안전하지 않음

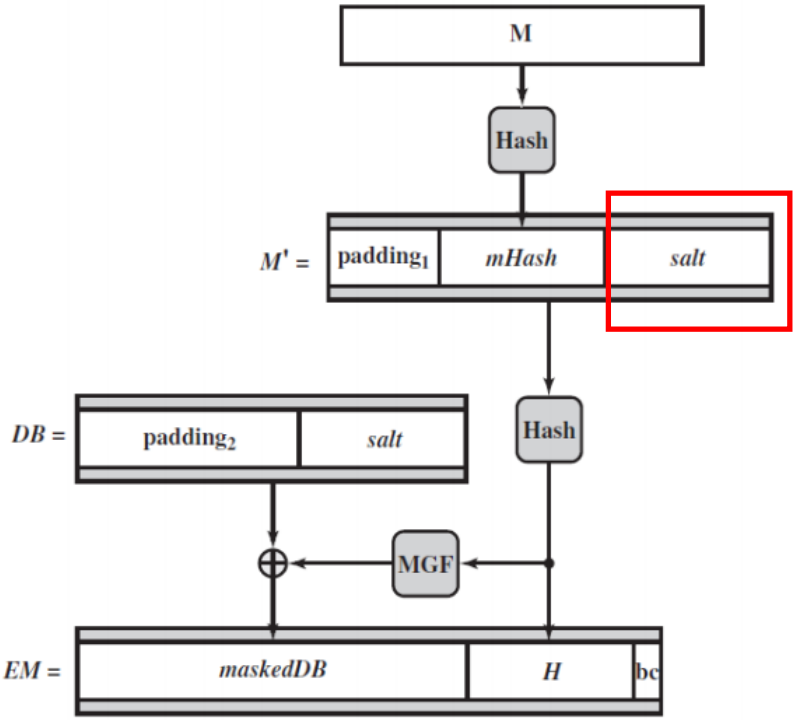
- **의미가 없는 메시지(예, 랜덤한 비트 열)라고 하더라도,**
  - 만약 올바른 디지털 서명을 만들 수 있다면(즉, 검증을 통과할 수 있는 디지털 서명이 된다면), 디지털 서명 알고리즘에 대한 위협

# RSA에서의 잠재적 위조

- 메시지를 RSA로 복호화하는 디지털 서명 알고리즘에서는 잠재적 위조가 가능
  - 랜덤한 비트 열 S를 RSA의 공개 키로 암호화한 것을 M이라고 하면, S가 M의 바른 디지털 서명이 되어 버리기 때문(이것은 앞 절에서 설명한 것의 역이다).
  - 공개 키는 공격자도 구할 수 있으므로 디지털 서명의 잠재적 위조가 가능

# RSA 잠재적 위조 대처법

- RSA-PSS(Probabilistic Signature Scheme)
  - RSA를 개량한 방법
  - 메시지에 대해서가 아니라 메시지의 해시 값에 대해서 서명하는 방법
  - 해시 값의 계산 시에는 메시지에 **솔트**를 더해 더욱 안전성을 향상



# 7.5 기타 공격

- 공개 키 암호에 대한 공격의 대부분은 디지털 서명에 대한 공격으로서도 사용가능
  - 개인 키에 대한 전사공격
  - RSA의 N을 소인수분해하는 공격

# 제8절 기타 기술과의 비교

## 8.1 메시지 인증 코드와 디지털 서명

## 8.2 하이브리드  암호 시스템과 해시 값에 대한 디지털 서명

# 8.1 메시지 인증 코드와 디지털 서명

- 대칭 암호와 공개 키 암호의 비교

| | 대칭 암호 | 공개 키 암호 |
|---|---|---|
| 송신자 | 공유 키로 암호화 | 공개 키로 암호화 |
| 수신자 | 공유 키로 복호화 | 개인 키로 복호화 |
| 키 배송 문제 | 일어난다 | 일어나지만, 공개 키의 인증이 별도로 필요 |
| 기밀성 | ○ | ○ |

- 메시지 인증 코드와 디지털 서명의 비교

| | 메시지 인증 코드 | 디지털 서명 |
|---|---|---|
| 송신자 | 공유 키로 MAC 값을 계산 | 개인 키로 서명을 작성 |
| 수신자 | 공유 키로 MAC 값을 계산 | 공개 키로 서명을 검증 |
| 키 배송 문제 | 일어난다 | 일어나지만, 공개 키의 인증이 별도로 필요 |
| 무결성 | ○ | ○ |
| 인증 | ○(통신 상대에 대해서만) | ○(제삼자에 대해서도) |
| 부인 방지 | × | ○ |

# 8.2 하이브리드 암호 시스템과 해시 값에 대한 디지털 서명

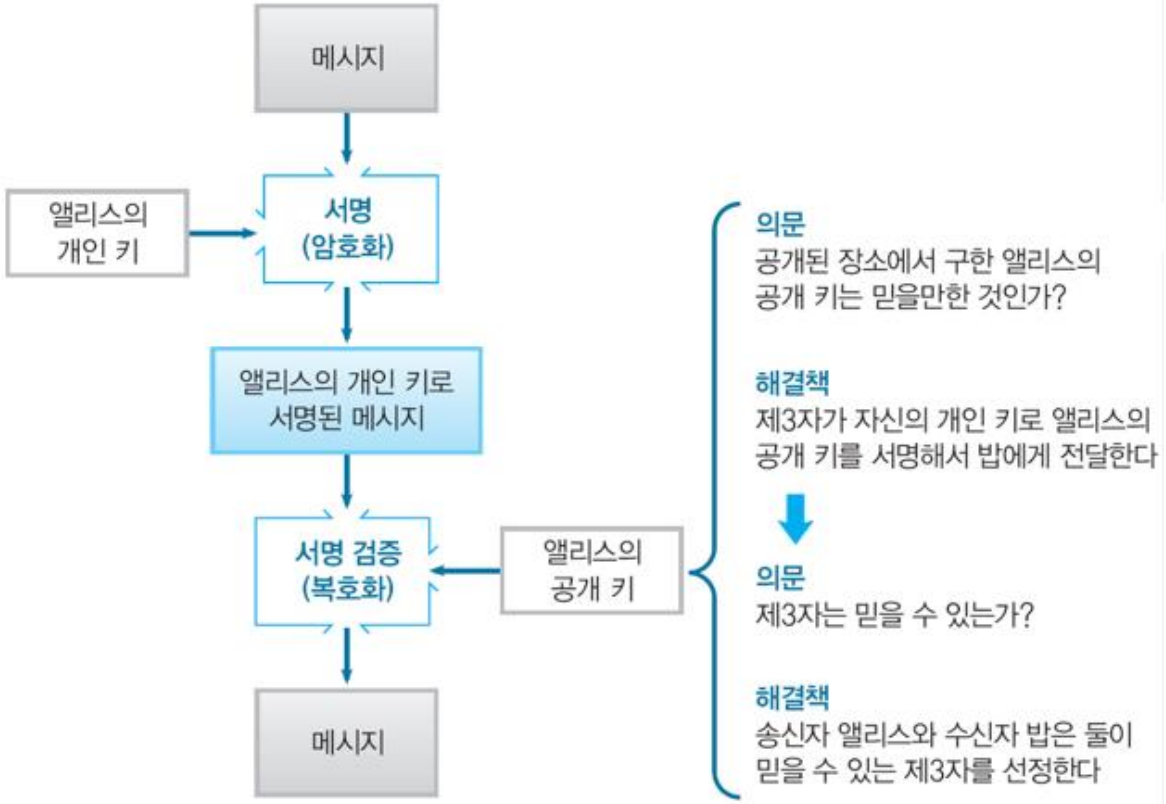
- 대칭 암호의 키는 기밀성의 핵심
- 일방향 해시 함수의 해시 값은 무결성의 핵심

# 제9절 디지털 서명으로 해결할 수 없는 문제

- 조작되어 있지 않은 공개 키를 「거짓 행세」하고 있지 않은 송신자로부터 받을 필요가 있다
  ➔ 공개키 송신자의 신뢰 문제

- 인증서
  - 바른 공개 키를 입수하기 위해 고안된 것이 **공개 키 기반 구조** (Public Key Infrastructure: PKI)
  - 공개 키 암호 및 디지털 서명의 기술을 사회적인 기반 구조로 만들어 가는 것

# 기술적인 방법만으로는 해결할 수 없는 문제



# 10절 전자서명 서비스

| 구분 | 인증사업자 (서비스명) | 서비스 개요 |
|---|---|---|
| 적용 중인 서비스 | **금융결제원** (브라우저인증서) | ○ **별도 프로그램 설치 없이** 웹 표준(HTML5)을 지원하는 클라우드 연동 브라우저 인증 서비스<br>○ **금융(10개 은행), 정부 및 공공기관(19개 사이트)** 적용 |
| | **카카오페이** (카카오페이인증) | ○ 모바일 메신저 카카오톡 기반의 인증 서비스로 **PKI 전자서명, 블록체인** 및 **생체인식 FIDO** 기술 적용<br>○ **국민연금공단, KB증권, BC카드, DB손해보험** 등 적용 |
| | **한국전자인증** (클라우드사인) | ○ **클라우드기반**의 **생체인증** 간편 전자서명서비스<br>○ **국세청 홈텍스, 서울시e바로, KB손해보험 등** 적용 |
| | **예티소프트** (VestPin) | ○ 순수 **HTML5** 기술 적용 **간편인증/간편서명** 솔루션<br>○ **카드사**(S,K,B 등) 및 **보험사**(K,M,H 등)에 적용 |
| | **아톤** (ATON mPKI) | ○ SW 보안매체(SE) 적용 **간편 전자서명 솔루션**<br>○ **IBK기업은행, 케이뱅크** 및 **PASS인증서** 등에 적용 |
| | **한국정보인증** (싸인오케이) | ○ WebTrust/AATL 인증기반의 전용인증서 및 **블록체인 기술을 적용**한 온라인 전자계약서비스<br>○ 연봉계약/대출약정서/렌탈/개인정보활용동의서 등에 적용 |

| | | |
|---|---|---|
| 적용 중인 서비스 | **위즈베라**<br>(PINsign) | ○ PIN 기반 간편 전자서명(사설) 서비스<br>○ KB모바일인증서(기적용) 및 의료분야(EMR) 적용 예정 |
| | **SKT**<br>(모바일 전자증명) | ○ **블록체인** 및 전자서명 기술 활용 **모바일 전자증명**서비스<br>○ 금융, 통신, 대학 등 전자증명 용도로 적용 예정 |
| | **시큐브**<br>(시큐사인) | ○ **생체수기서명** 인증 기술을 적용한 전자서명서비스<br>○ 전자계약, 지불결제, 본인인증(신원확인) 분야에 적용 예정 |
| | **라온시큐어**<br>(OmniOne) | ○ **블록체인 기술을 활용**한 전자서명 인증서비스<br>○ **병무청** 민원포탈시스템 적용 예정 |
| | **코스콤**<br>(OpenPass) | ○ 간편비밀번호, 생체, 패턴 등 편리한 인증서비스<br>○ **공인사설인증, FIDO, OTP 등 다양한 통합인증서비스** 제공 |
| | **아이콘루프**<br>(my-ID) | ○ **블록체인 기술을 활용**한 전자서명 인증서비스<br>○ 증권사(9개), 은행(2개) 등 25개 사이트 적용예정 |

# 연습문제 풀이

1. 디지털 서명을 했을 때 막을 수 있는 것들로만 묶인 것은?
① 데이터 변경, 거짓 행세, 부인 방지
② 비밀 유지, 데이터 변경, 도청 방지
③ 도청 방지, 거짓 행세, 부인 방지
④ 부인 방지, 데이터 변경, 도청 방지
⑤ 데이터 변경, 도청 방지, 비밀 유지

1. 디지털 서명을 했을 때 막을 수 있는 것들로만 묶인 것은?

① 데이터 변경, 거짓 행세, 부인 방지

② 비밀 유지, 데이터 변경, 도청 방지

③ 도청 방지, 거짓 행세, 부인 방지

④ 부인 방지, 데이터 변경, 도청 방지

⑤ 데이터 변경, 도청 방지, 비밀 유지

2. 송신자가 공개 키 암호를 이용하여 평문을 암호화한 다음 수신자에게 보낼 때 사용하는 키와 평문에 서명을 하여 보낼 때 사용하는 키를 순서대로 묶은 것은?

① 수신자의 공개 키, 송신자의 개인 키
② 송신자의 개인 키, 수신자의 개인 키
③ 수신자의 공개 키, 송신자의 공개 키
④ 송신자의 개인 키, 수신자의 공개 키

2. 송신자가 공개 키 암호를 이용하여 평문을 암호화한 다음 수신자에게 보낼 때 사용하는 키와 평문에 서명을 하여 보낼 때 사용하는 키를 순서대로 묶은 것은?

① 수신자의 공개 키, 송신자의 개인 키
② 송신자의 개인 키, 수신자의 개인 키
③ 수신자의 공개 키, 송신자의 공개 키
④ 송신자의 개인 키, 수신자의 공개 키

3. 디지털 서명을 할 때 메시지에 직접 서명하는 방법이 있는데 이 방법이 잘 사용되지 않는 이유는?
① 암호화가 되지 않기 때문에
② 무결성이 보장되지 않기 때문에
③ 서명에 시간이 많이 걸리기 때문에
④ 메시지 내용의 일부만 서명되기 때문에
⑤ 부인방지 효과가 떨어지기 때문에

3. 디지털 서명을 할 때 메시지에 직접 서명하는 방법이 있는데 이 방법이 잘 사용되지 않는 이유는?

① 암호화가 되지 않기 때문에

② 무결성이 보장되지 않기 때문에

③ 서명에 시간이 많이 걸리기 때문에

④ 메시지 내용의 일부만 서명되기 때문에

⑤ 부인방지 효과가 떨어지기 때문에

4. 공개 키에 신뢰받는 제 3자의 디지털 서명을 붙인 것을 _____(이)라 한다.

① 제 3자 인증서
② 클리어 서명
③ 디지털 서명
④ 공개 키 인증서
⑤ 인증 서명

4. 공개 키에 신뢰받는 제 3자의 디지털 서명을 붙인 것을 _____(이)라 한다.

① 제 3자 인증서
② 클리어 서명
③ 디지털 서명
④ 공개 키 인증서
⑤ 인증 서명

5. 다음 중에서 디지털 서명에 사용되는 방법이 아닌 것은?

① RSA

② Diffie-Hellman

③ DSA

④ ElGamal

⑤ Rabin

5. 다음 중에서 디지털 서명에 사용되는 방법이 아닌 것은?

① RSA

② Diffie-Hellman

③ DSA

④ ElGamal

⑤ Rabin

6. 디지털 서명에 대한 공격에 사용할 수 있는 공격방법은?

① 중간자 공격

② 재전송 공격

③ 생일 공격

④ 트래픽분석 공격

⑤ 선택평문 공격

6. 디지털 서명에 대한 공격에 사용할 수 있는 공격방법은?

① 중간자 공격

② 재전송 공격

③ 생일 공격

④ 트래픽분석 공격

⑤ 선택평문 공격

7. 메시지 인증 코드와 디지털 서명에 대한 유사점과 다른 점으로 묶인 것은?

① 무결성이 보장된다, 메시지 인증코드는 제 3자의 인증이 불가능하다.

② 기밀성이 보장된다. 메시지 인증코드는 제 3자의 인증이 불가능하다.

③ 부인 방지가 된다. 디지털 서명은 제 3자의 인증이 불가능하다.

④ 무결성이 보장된다. 디지털 서명은 제 3자의 인증이 불가능하다.

⑤ 기밀성이 보장된다. 디지털 서명은 제 3자의 인증이 가능하다.

7. 메시지 인증 코드와 디지털 서명에 대한 유사점과 다른 점으로 묶인 것은?

① 무결성이 보장된다, 메시지 인증코드는 제 3자의 인증이 불가능하다.

② 기밀성이 보장된다. 메시지 인증코드는 제 3자의 인증이 불가능하다.

③ 부인 방지가 된다. 디지털 서명은 제 3자의 인증이 불가능하다.

④ 무결성이 보장된다. 디지털 서명은 제 3자의 인증이 불가능하다.

⑤ 기밀성이 보장된다. 디지털 서명은 제 3자의 인증이 가능하다.

8. 디지털 서명을 하여 수신자에게 전송할 경우에 중간의 공격자가 서명된 메시지를 도청한 다음에 내용을 읽어볼 수 있기 때문에 디지털 서명으로 _____(을)를 보장하지는 못한다.

① 인증
② 부인방지
③ 기밀성
④ 무결성
⑤ 가용성

8. 디지털 서명을 하여 수신자에게 전송할 경우에 중간의 공격자가 서명된 메시지를 도청한 다음에 내용을 읽어볼 수 있기 때문에 디지털 서명으로 _____(을)를 보장하지는 못한다.

① 인증
② 부인방지
③ 기밀성
④ 무결성
⑤ 가용성

9. 디지털 서명 알고리즘에서 잠재적 위조를 방지하기 위해 RSA를 개량한 방법을 무엇이라고 하는가?

① RSA-AES
② RSA-ECC
③ RSA-PSS
④ RSA-DSA
⑤ RSA-HASH

9. 디지털 서명 알고리즘에서 잠재적 위조를 방지하기 위해 RSA를 개량한 방법을 무엇이라고 하는가?

① RSA-AES
② RSA-ECC
③ RSA-PSS
④ RSA-DSA
⑤ RSA-HASH

# 참고문헌


- 암호학과 네트워크 보안, Behrouz A. Forouzan 지음, 이재광외 3인 역, 한티미디어
- 컴퓨터 보안과 암호, WILLIAM STALLINGS 지음, 최용락외 2인 역, 그린출판사
- 전자서명법, [시행 2022. 10. 20.] [법률 제18479호, 2021. 10. 19., 일부개정], https://www.law.go.kr/%EB%B2%95%EB%A0%B9/%EC%A0%84%EC%9E%90%EC%84%9C%EB%AA%85%EB%B2%95

# Q & A