

# Network Security Essentials

## - Chapter\_3 공개 키 암호와 메시지 인증(2) -

손 우 영([wooyoung@pel.sejong.ac.kr](mailto:wooyoung@pel.sejong.ac.kr))

세종대학교 프로토콜공학연구실

# 목 차

---

- 공개 키 암호 원리
- 공개 키 암호 알고리즘
- 디지털 서명

# 목 차

---

- 공개 키 암호 원리
- 공개 키 암호 알고리즘
- 디지털 서명

# 공개 키 암호 원리

- 공개 키 암호(Public Key Encryption)

- 정의

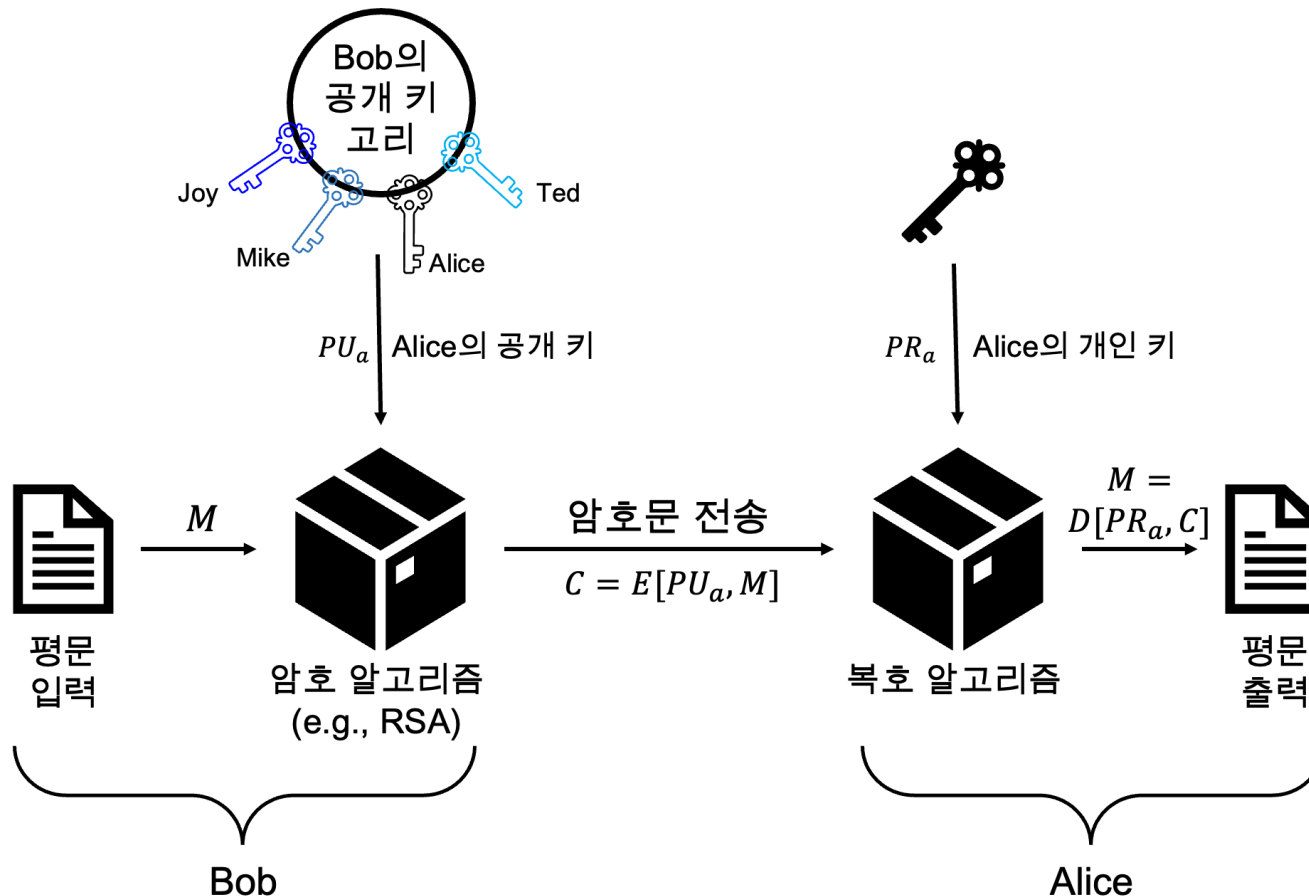
- 암호화와 복호화에 서로 다른 키(공개 키, 개인 키)를 사용하는 암호화 방식

- 구성요소

용어	기호	의미
평문	$M$	사람이 읽을 수 있는 메시지나 데이터로서 알고리즘의 입력으로 사용됨
암호문	$C$	출력으로 나오는 암호화 된 메시지이며 평문과 키에 의해 생성됨
공개 키	$PU_a$	암호화 또는 복호화에 사용되며 공개된 키
개인 키	$PR_a$	암호화 또는 복호화에 사용되며 소유자만 알고 공개되지 않는 키
암호 알고리즘	$E$	평문을 변환시켜 암호문으로 만들기 위해 사용하는 알고리즘
복호 알고리즘	$D$	평문을 암호화 할 때 사용한 키에 대응하는 키를 이용하여 암호문을 원래의 평문으로 변환할 때 사용하는 알고리즘

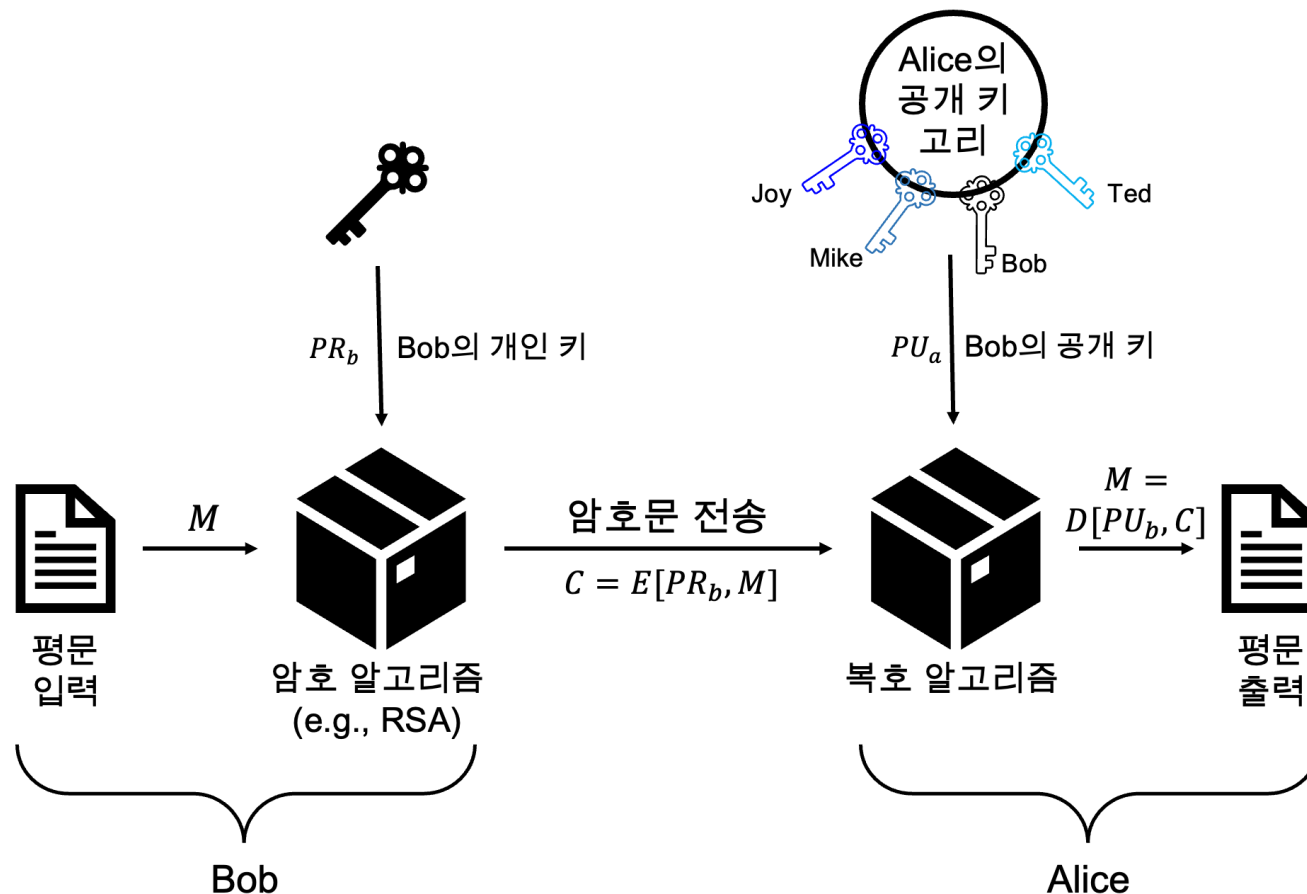
# 공개 키 암호 원리

- 공개 키 암호(Public Key Encryption)
- 암호 방식(1/2)
  - 공개 키에 의한 암호화



# 공개 키 암호 원리

- 공개 키 암호(Public Key Encryption)
- 암호 방식(2/2)
  - 개인 키에 의한 암호화



# 공개 키 암호 원리

## • 공개 키 암호화 방식과 대칭 키 암호화 방식의 비교

	공개 키 암호화 방식	대칭 키 암호화 방식
정의	암호화와 복호화에 서로 다른 키 사용	암호화와 복호화에 동일한 키 사용
키	공개 키, 개인 키	비밀 키
암호 방식	수를 이용한 수학적 함수 응용	기호(문자나 비트)를 대체/치환
장점	키 분배 및 관리 쉬움	계산 속도 빠름
단점	계산 속도 느림	키 관리의 어려움

# 공개 키 암호 원리

---

- 공개 키 암호(Public Key Encryption)

- 요건(1/2)

- 한 쌍의 키(공개 키:  $PU_a$ , 개인 키:  $PR_a$ )가 생성될 때 컴퓨터의 계산 시간을 고려해야 함
- 공개 키와 암호화될 메시지를 알고있는 송신자는 암호문을 계산적으로 쉽게 구할 수 있어야 함
  - $C = E[PU_b, M]$
- 수신자가 암호문을 자신의 개인 키를 이용해서 복호화하는 것이 계산적으로 쉬워야 함
  - $M = D[PR_b, C] = D[PR_b, E(PU_b, M)]$



# 공개 키 암호 원리

---

- 공개 키 암호(Public Key Encryption)
- 요건(2/2)
  - 공개 키를 알고 있는 공격자가 개인 키를 알아내는 것이 계산적으로 불가능해야 함
  - 공개 키와 암호문을 알고 있는 공격자가 원문을 알아내는 것이 계산적으로 불가능해야 함
  - 2개의 키 중 하나를 암호화에 사용하면 다른 하나는 복호화에 사용할 수 있어야 함
    - $M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$

# 목 차

---

- 공개 키 암호 원리
- 공개 키 암호 알고리즘
- 디지털 서명

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 정의

- 큰 정수의 소인수분해에 기반하여 수학적으로 구현한 암호화 알고리즘

- 특징

- 공개 키 암호 알고리즘
  - 수신자의 공개 키로 암호화하고 수신자의 개인 키로 복호화 함
- 큰 수를 소인수 분해하는 것이 어렵다는 것에 기반

# 공개 키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 암호 알고리즘
  - 구성

기호	의미	특징
$M$	평문	
$C$	암호문	
$p, q$	키를 생성하기 위해 선택하는 두 소수	큰 소수
$n$	암/복호화에서 모듈로(modulus)로 이용될 두 소수 $p, q$ 의 곱	공개
$e$	공개 키의 인자 값	<ul style="list-style-type: none"><li>• <math>\phi(n)</math>과 서로소</li><li>• 공개</li></ul>
$d$	개인 키의 인자 값	<ul style="list-style-type: none"><li>• <math>de \bmod \phi(n) = 1</math></li><li>• 수신자만 알아야 함</li></ul>
$PU = \{e, n\}$	공개 키	암호화 시 사용
$PR = \{d, n\}$	개인 키	복호화 시 사용

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 과정(1/3)

- 키 생성 과정(1/2)

- 1. 서로 다른 두 소수  $p, q$  선택

- 2.  $n$  계산

- $n = p \times q$

- 3.  $\phi(n)$  계산

- $\phi(n)$ : 오일러 함수로서 양의 정수 중  $n$ 과 서로소인 수의 개수

- $p, q$ 가 모두 소수이므로  $\phi(n) = (p - 1)(q - 1)$

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 과정(2/3)

- 키 생성 과정(2/2)

- 4. 정수  $e$  선택

- 정수  $e$ 는  $\phi(n)$ 와 서로소
      - $\gcd(\phi(n), e) = 1 ; 1 < e < \phi(n)$ 
        - $\gcd(\text{Greatest Common Multiple})$  : 최대공약수
      - 공개 키  $PU = \{e, n\}$  생성

- 5.  $d$  계산

- $de \bmod \phi(n) = 1$
      - 개인 키  $PR = \{d, n\}$  생성

# 공개 키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 암호 알고리즘

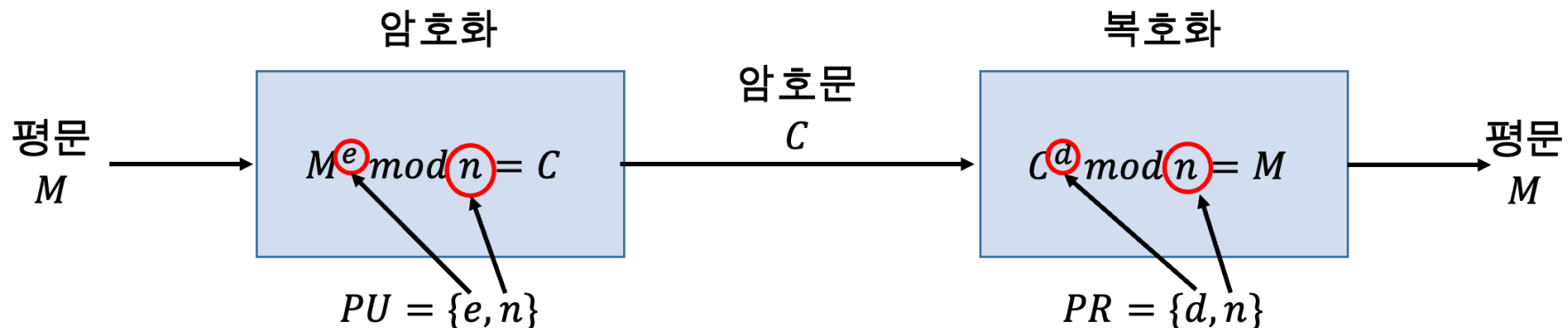
- 과정(3/3)

- 암호화

- 공개 키  $PU = \{e, n\}$  사용
    - $C = M^e \bmod n$

- 복호화

- 개인 키  $PR = \{d, n\}$  사용
    - $M = C^d \bmod n$



# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(1/6)

- 키 생성 과정(1/2)

- 1. 두 소수  $p = 17, q = 11$  선택

- 2.  $n = pq = 17 \times 11 = 187$  계산

- 3.  $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$  계산

- 4.  $\phi(n) = 160$  보다 작으면서  $\phi(n)$ 과 서로소인 수  $e$ 를 선택 ;  $e = 7$



# 공개 키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(2/6)

- 키 생성 과정(2/2)

- 5.  $d < 160$  이면서  $de \bmod 160 = 1$ 인 수  $d$ 를 결정

- 확장된 유클리드 알고리즘 이용

- $7d \bmod 160 = 1$

- $7d - 160k = 1$  를 만족하는  $d$  찾기

$d$	$-k$	$7d - 160k$
0	1	160
1	0	7
-22	0	6
23	0	1

- $\therefore n = 187, e = 7, d = 23$

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(3/6)

- 암호화 ( $M = 88$ )

- 공개 키  $PU = \{e, n\}$  사용 ;  $PU = \{7, 187\}$

- $C = M^e \bmod n$  ;  $C = 88^7 \bmod 187$

- 더 간단한 계산을 위해 이진수를 이용한 연속제곱법 사용

- 7을 이진수로 표현

- $7 = 2^0 + 2^1 + 2^2$

- $88^7 \bmod 187 = 88^{2^2+2^1+2^0} \bmod 187$

- $$= [(88^{2^0} \bmod 187) \times (88^{2^1} \bmod 187) \times (88^{2^2} \bmod 187)] \bmod 187$$

# 공개 키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(4/6)

- 암호화 ( $M = 88$ )

- modulus 연산

$88^1 \bmod 187 \equiv 88$
$88^2 \bmod 187 \equiv 77$
$88^4 \bmod 187 \equiv 77^2 \bmod 187 \equiv 132$

- $[(88^{2^0} \bmod 187) \times (88^{2^1} \bmod 187) \times (88^{2^2} \bmod 187)] \bmod 187$   
 $\equiv (88 \times 77 \times 132) \bmod 187 \equiv 11$

- 암호문  $C=11$

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(5/6)

- 복호화 ( $C = 11$ )

- 공개 키  $PR = \{d, n\}$  사용 ;  $PR = \{23, 187\}$

- $M = C^d \bmod n$ ;  $M = 11^{23} \bmod 187$

- 더 간단한 계산을 위해 이진수를 이용한 연속제곱법 사용

- 23을 이진수로 표현

- $23 = 2^0 + 2^1 + 2^2 + 2^4$

- $11^{23} \bmod 187 = 11^{2^0+2^1+2^2+2^4} \bmod 187$

- $$= [(11^{2^0} \bmod 187) \times (11^{2^1} \bmod 187) \times (11^{2^2} \bmod 187) \times (11^{2^4} \bmod 187)] \bmod 187$$

# 공개 키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 예(6/6)

- 복호화 ( $C = 11$ )

- modulus 연산

$11^1 \bmod 187 \equiv 11$
$11^2 \bmod 187 \equiv 121$
$11^4 \bmod 187 \equiv 121^2 \bmod 187 \equiv 55$
$11^8 \bmod 187 \equiv 55^2 \bmod 187 \equiv 33$
$11^{16} \bmod 187 \equiv 33^2 \bmod 187 \equiv 154$

- $[(11^{2^0} \bmod 187) \times (11^{2^1} \bmod 187) \times (11^{2^2} \bmod 187) \times (11^{2^4} \bmod 187)] \bmod 187$   
 $\equiv (11 \times 121 \times 55 \times 154) \bmod 187 \equiv 88$

- 평문  $M=88$

# 공개 키 암호 알고리즘

---

- RSA(Rivest Shamir Adleman) 암호 알고리즘
  - 보안(1/2)
    - 수학적 공격(Mathematical Attack)
      - 방법
        - 두 개의 소수 곱을 인수분해
      - 대응
        - e와 d의 비트 수가 커야 함
    - 타이밍 공격(Timing Attacks)
      - 방법
        - 복호화 알고리즘의 실행 시간의 차이 계산
      - 대응
        - 랜덤 지체 방법 사용

# 공개 키 암호 알고리즘

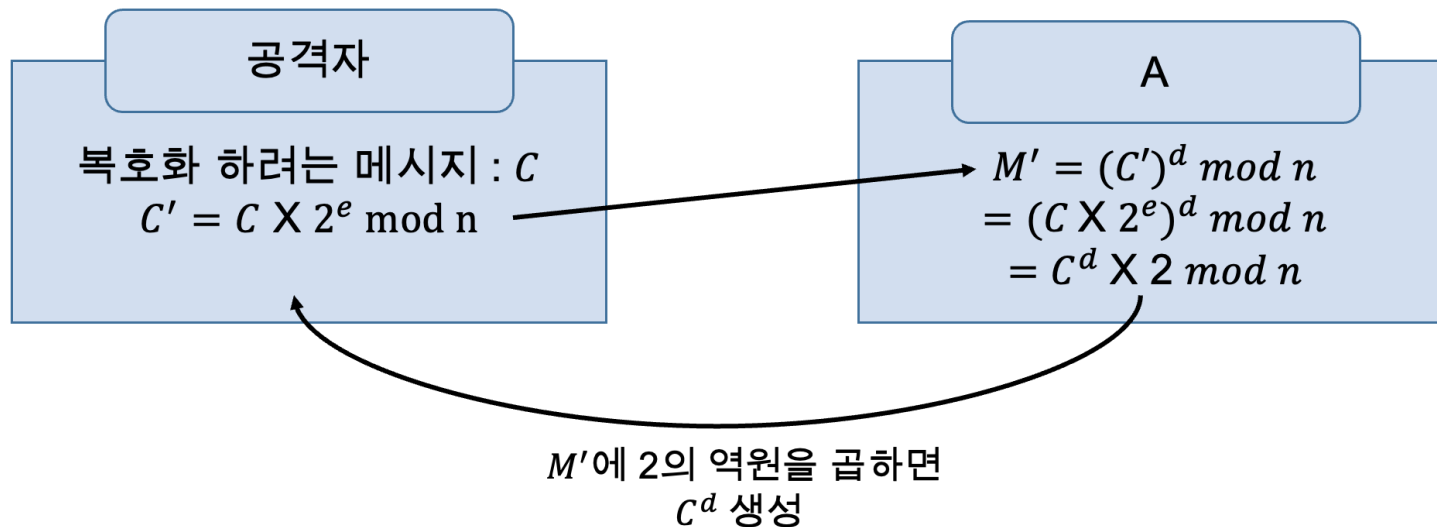
- RSA(Rivest Shamir Adleman) 암호 알고리즘

- 보안(2/2)

- 선택 암호문 공격(Chosen Cipher Attacks)

- 방법

- 암호 해독에 필요한 정보를 드러내는 블록 선택하여 RSA 알고리즘 특성 악용



- 대응

- 패딩 추가

# 공개 키 암호 알고리즘

---

- Diffie-Hellman 키 교환
  - 정의
    - 비밀 키를 교환하기 위한 알고리즘
  - 특징
    - 이산 대수 문제 계산에 착안하여 만들어짐
    - 암호화나 전자서명에 사용되지 않음
    - 자신의 개인 키와 상대방의 공개 키로 비밀 키 생성



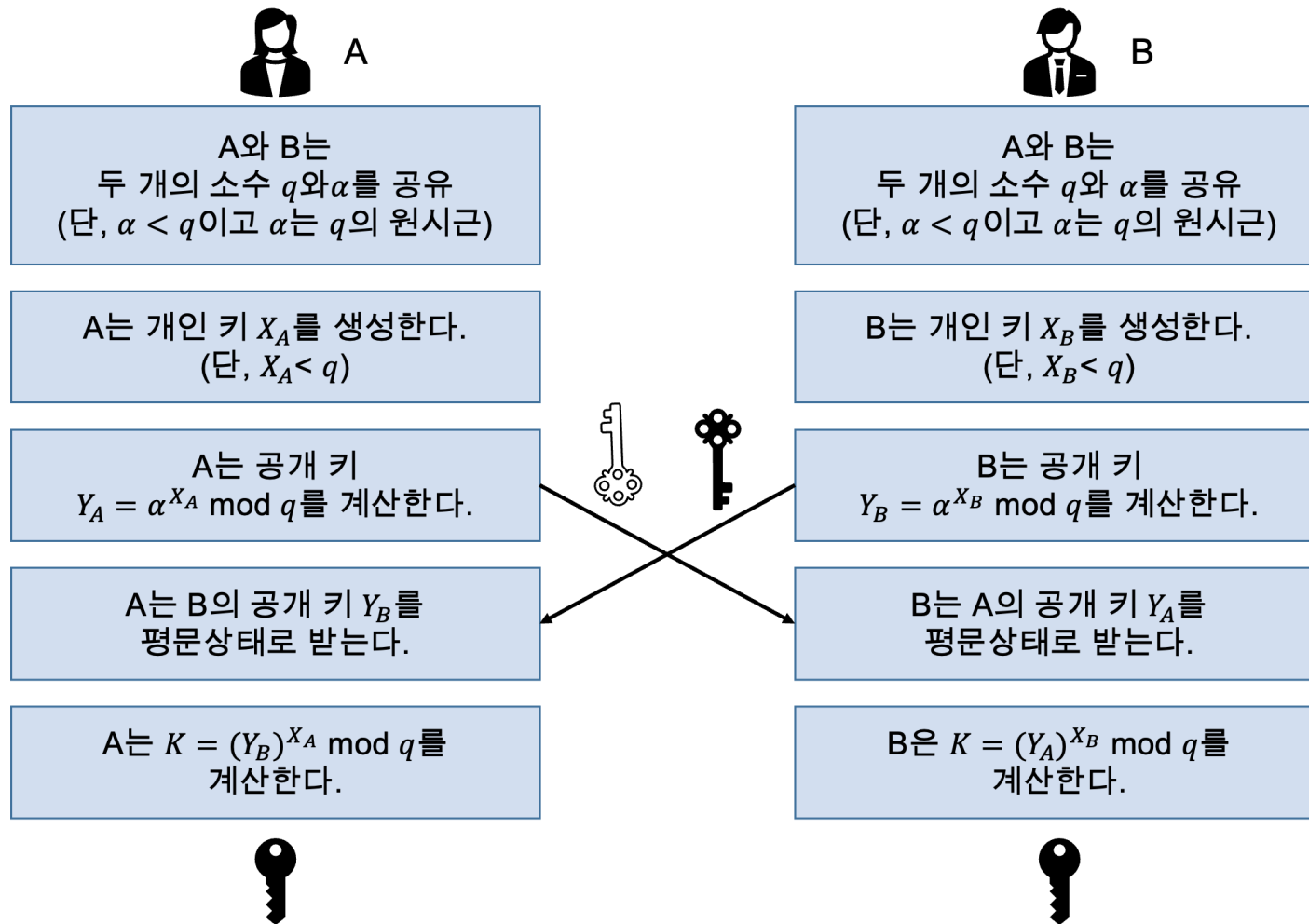
# 공개 키 암호 알고리즘

---

- Diffie-Hellman 키 교환
- 이산 대수 문제(Discrete Logarithms Problem)
  - 원시근(Primitive Root)
    - 소수  $p$ 의 원시근
      - 자신의 거듭제곱을 이용하면 1부터  $p - 1$ 까지의 정수를 모두 생성해 낼 수 있는 수
      - e.g., 어떤 수  $a$ 가 소수  $p$ 의 원시근
        - $\{a \bmod p, a^2 \bmod p, \dots, a^{p-1} \bmod p\} = \{1, 2, \dots, p - 1\}$
  - 이산 대수
    - $b = a^i \bmod p$  ( $0 \leq i \leq p - 1$ ) 일 때의  $i$

# 공개 키 암호 알고리즘

- Diffie-Hellman 키 교환
  - 키 교환 알고리즘



# 공개 키 암호 알고리즘

---

- Diffie-Hellman 키 교환

- 키 교환 알고리즘

- 동일한 비밀 키 생성

- $$\begin{aligned} K &= (Y_B)^{X_A} \bmod q \\ &= (a^{X_B} \bmod q)^{X_A} \bmod q \\ &= (a^{X_B})^{X_A} \bmod q \\ &= a^{X_A X_B} \bmod q \\ &= (a^{X_A})^{X_B} \bmod q \\ &= (a^{X_A} \bmod q)^{X_B} \bmod q \\ &= (Y_A)^{X_B} \bmod q \end{aligned}$$

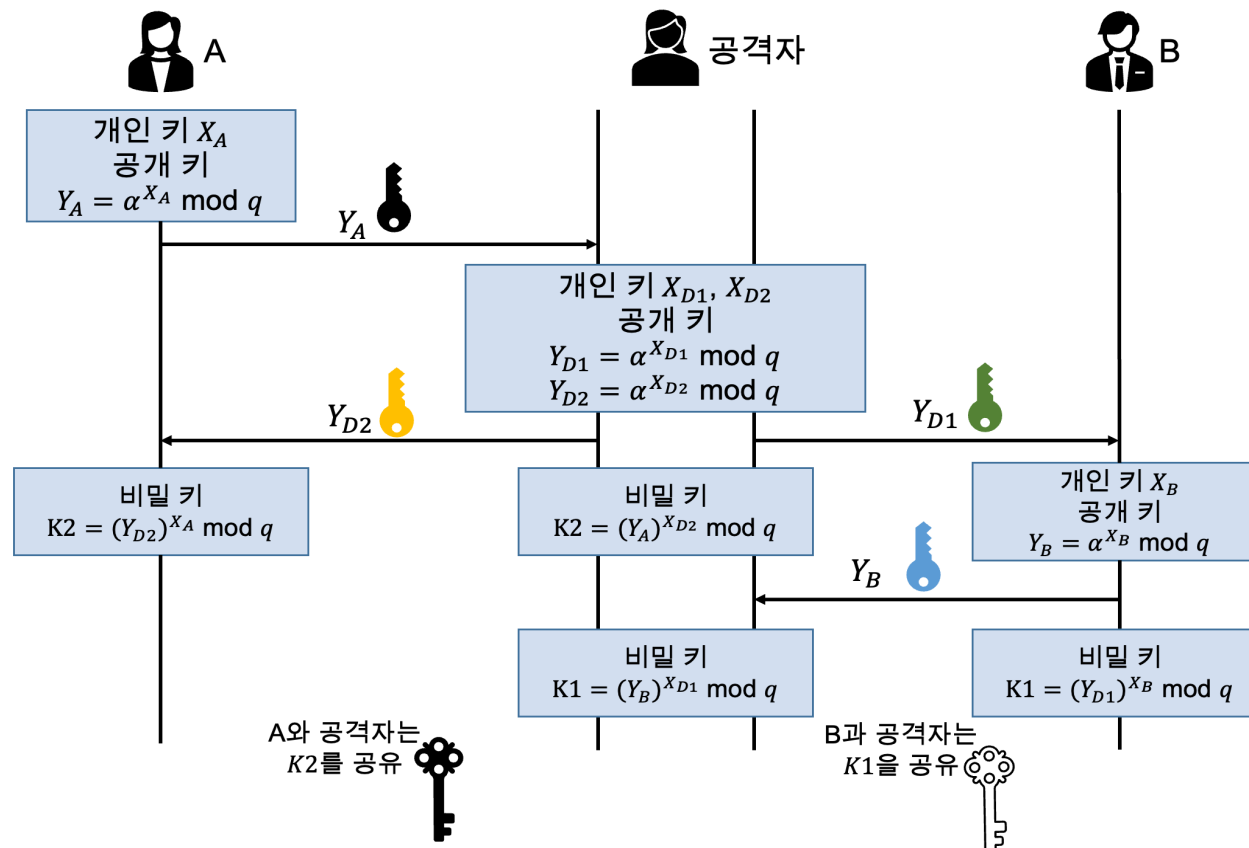
# 공개 키 암호 알고리즘

- Diffie-Hellman 키 교환

- 중간자 공격(Man-in-the-Middle Attack)

- 정의

- 공격자가 송/수신자 사이에서 전달되는 메시지를 가로채 자신과 송/수신자 사이의 키를 생성하는 공격



# 공개 키 암호 알고리즘

- 타원 곡선 암호(ECC, Elliptic Curve Cryptography)

- 정의

- 타원 곡선에 기반한 공개 키 암호 방식

- 특징

- 타원 곡선 이산 대수 문제에 기반

- 구성

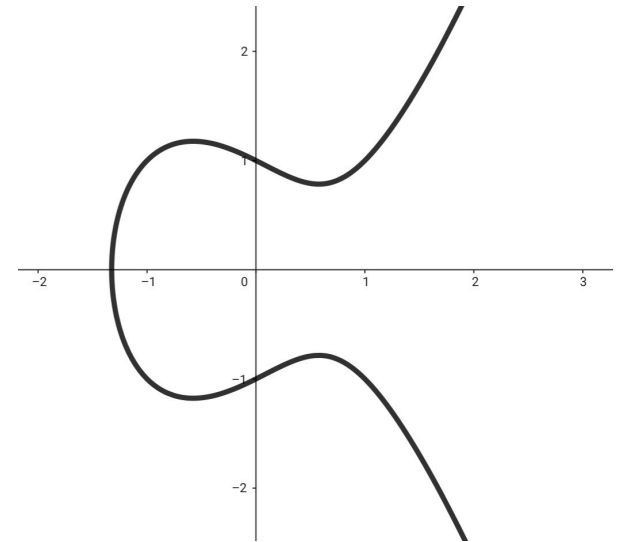
- 점  $P, Q$  : 타원 곡선( $y^2 = x^3 + ax + b$ ) 위의 점
- 정수  $k$  : 점  $P$ 를 더하여 새로운 점  $Q$ 를 계산하는 횟수

- 원리

- $k$ 와  $P$ 를 통해  $Q$ 를 계산하는 것은 쉽지만  $Q = kP$ 에서  $k$ 를 계산하는 것은 어려움

- 키

- 공개 키:  $kP$
- 개인 키:  $k$



# 목 차

---

- 공개 키 암호 원리
- 공개 키 암호 알고리즘
- 디지털 서명

# 디지털 서명

---

- 정의

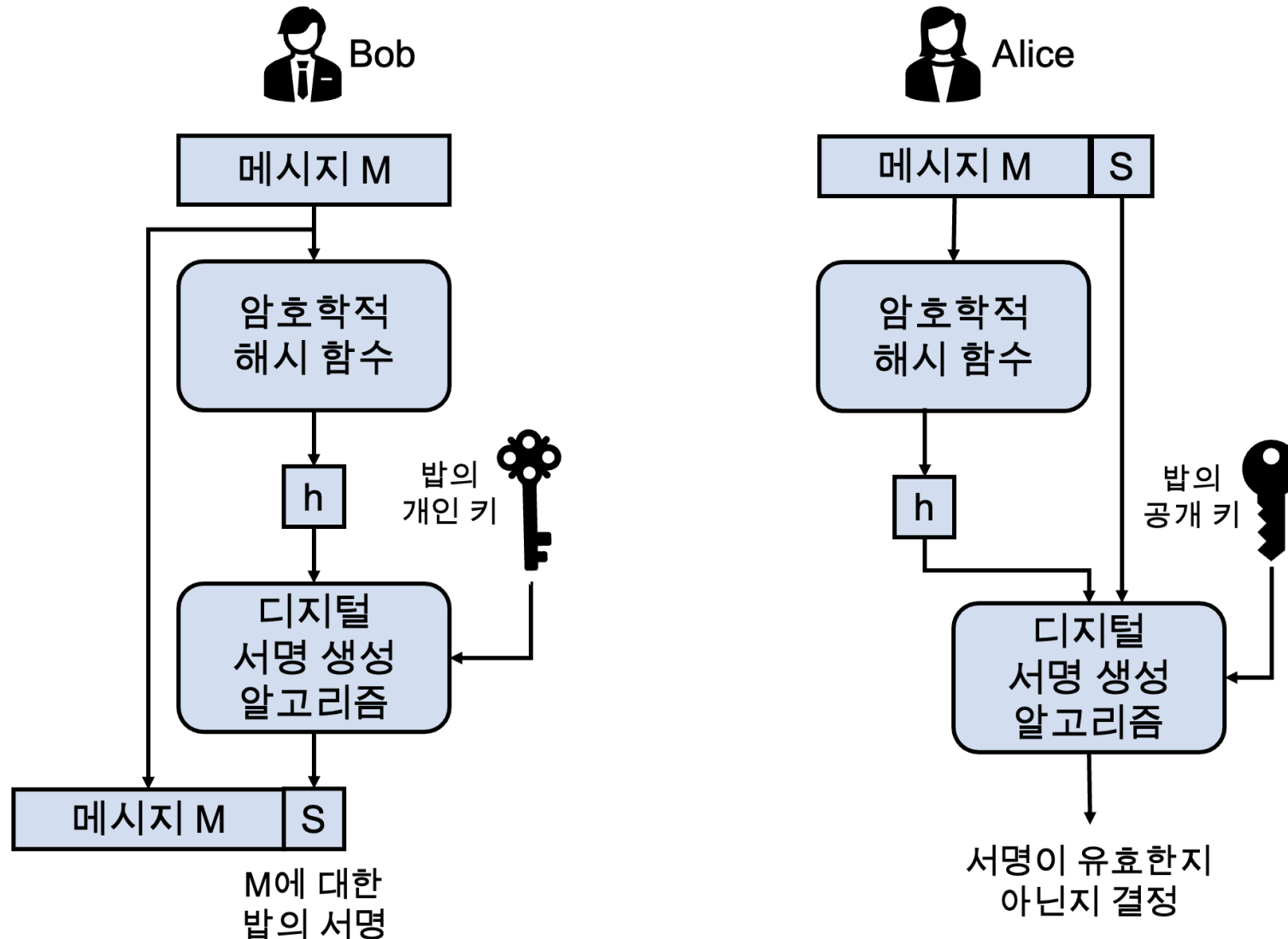
- 공개 키 암호화를 이용하여 발신자가 신분이 도용되지 않은 것을 증명하며, 데이터가 변조되지 않았음을 증명하는 것

- 특징

- 출처인증, 데이터 무결성 검증
- 해시 함수 사용
- 송신자의 개인 키로 암호화, 송신자의 공개 키로 복호화
- 기밀성 보장하지 못함

# 디지털 서명

- 생성 과정





# 디지털 서명

---

- RSA 디지털 서명 알고리즘

- 정의

- RSA 암호화를 이용한 디지털 서명

- 과정(1/2)

- 키 생성

1. 두 개의 소수  $p, q$  선택 후  $n = p \times q$  계산
2.  $\phi(n) = (p - 1)(q - 1)$  계산
3. 공개 키로  $e$  선택 후  $de \bmod \phi(n) = 1$  를 만족하는 값  $d$  계산
4. 송신자는  $d$  값을 취한 후  $n, e$  를 공개

- 서명

- 송신자는 자신의 개인 키로 계산한 값  $S = M^d \bmod n$  을 이용해서 메시지에 서명을 생성하고 그 서명과 메시지를 수신자에게 보냄

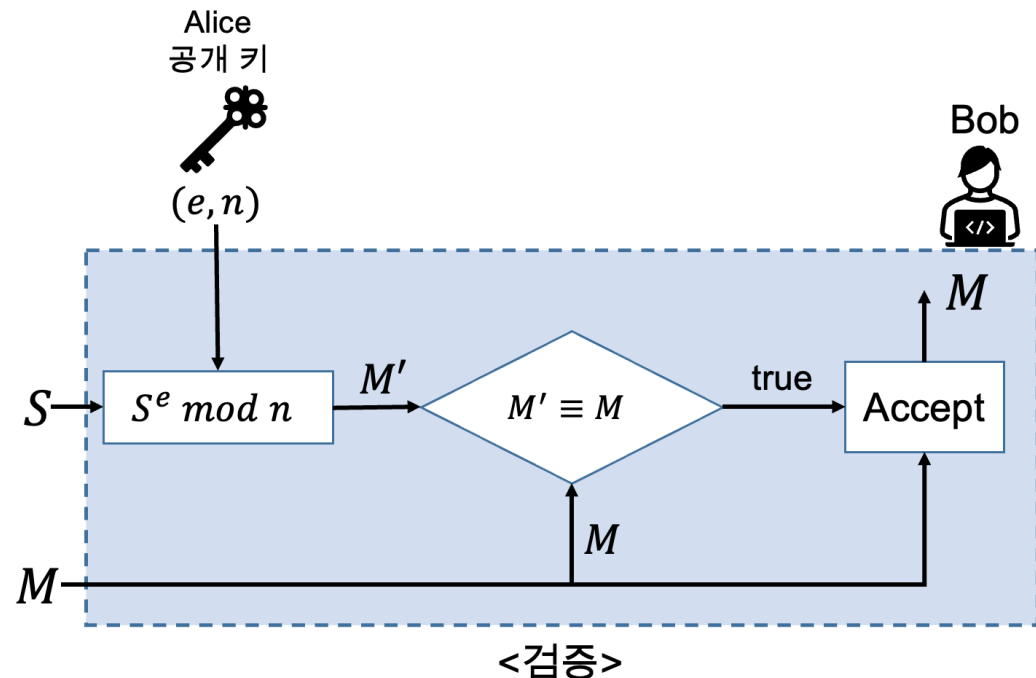
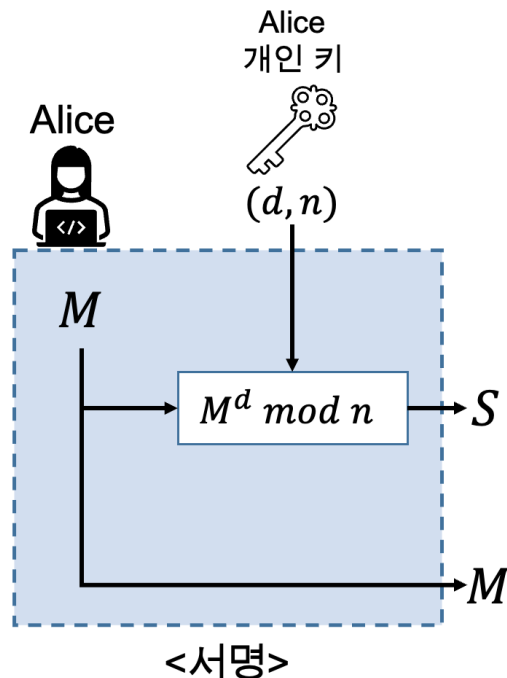
# 디지털 서명

- RSA 디지털 서명 알고리즘

- 과정(2/2)

- 검증

1. 수신자는 메시지( $M$ )와 서명( $S$ )을 수신 후 송신자의 공개 키를 적용하여 메시지  $M' = S^e \bmod n$  을 구함
2. 송신자는 이 값  $M'$ 과  $M$ 을 비교
3. 만약 두 값이 합동으로 같다면 송신자는 이 메시지를 받음



---

# Thanks!

손 우 영 ([wooyoung@pel.sejong.ac.kr](mailto:wooyoung@pel.sejong.ac.kr))