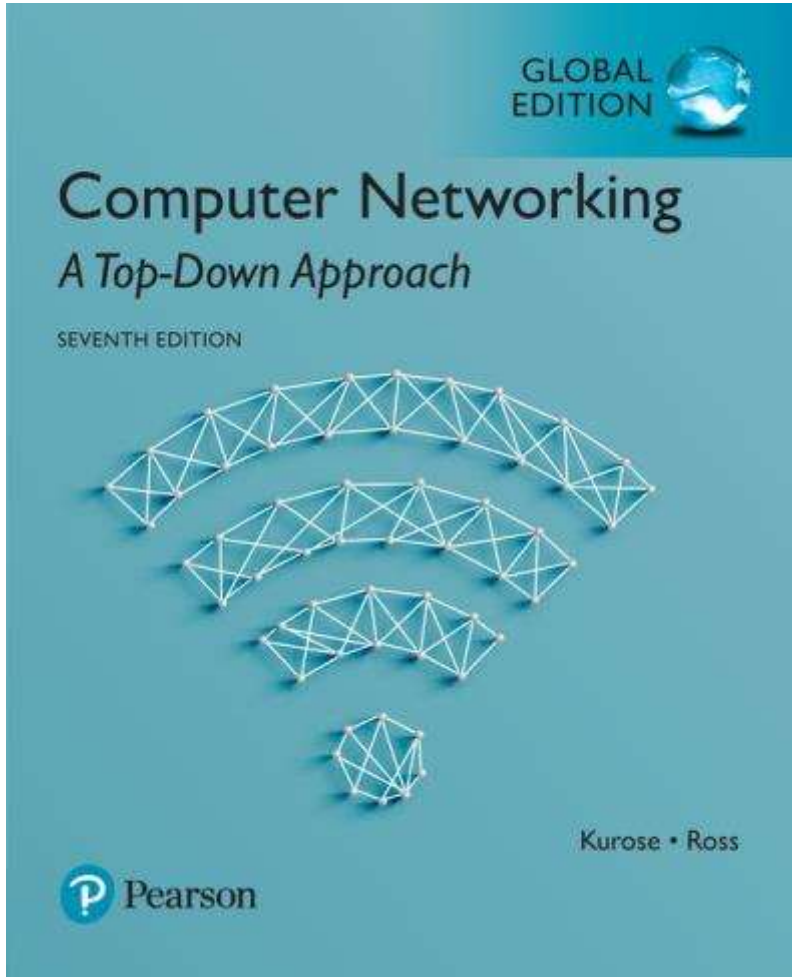
제28강
공개키 암호

Computer Networking: A Top Down Approach

컴퓨터 네트워크
(2019년 1학기)

박승철교수

한국기술교육대학교
컴퓨터공학부



Pre-study Test :

1) 다음 중 송.수신자간의 통신을 위한 공개키 암호화에 대한 설명 중 틀린 것은?
① 서로 다른 공개키와 개인키 쌍을 사용한다.
② 수신자의 공개키로 암호화할 수 있다.
③ 송신자의 개인키로 암호화할 수 있다.
④ 송신자의 공개키로 암호화할 수 있다.

2) 다음 중 공개키 암호화에 대한 설명 중 틀린 것은?
① 대용량 데이터를 효과적으로 암호화한다.
② 주로 해시값을 암호화하기 위해 사용한다.
③ 공개키 인증서가 필요하다.
④ 디지털 서명에 사용된다.

3) 다음 중 RSA 키 생성 알고리즘에 대한 설명 중 틀린 것은?
① 큰 수의 소인수 분해의 어려운 점을 활용한다.
② 큰 소수 2개를 선택하여 활용한다.
③ 오일러 2차 정리(Euler's Theorem)에 기초하여 개발되었다.
④ 공개키는 오일러 함수값(Ø(n)) 보다 작은 임의의 수를 선택한다.

4) RSA 키 생성 알고리즘에서 선택한 소수 2개가 5와 7일 경우 공개키를 (35, 5)로 선택할 때 개인키는?
① (35, 5)
② (35, 6)
③ (35, 7)
④ (35, 8)

5) RSA에서 공개키로부터 개인키를 보호할 수 있는 이유는?
① 큰 수 n에 대한 이산대수 문제의 어려움
② 큰 수 n에 대한 소인수분해 문제의 어려움
③ 큰 수 n에 대한 지수함수 문제의 어려움
④ 큰 수 n에 대한 지수 모듈로 문제의 어려움

6) 다음 중 공개키 암호화가 아닌 것은?
① SEED
② RSA
③ Diffie-Hellman
④ Elliptic Curve Crytography

Public Key(공개키) Cryptography

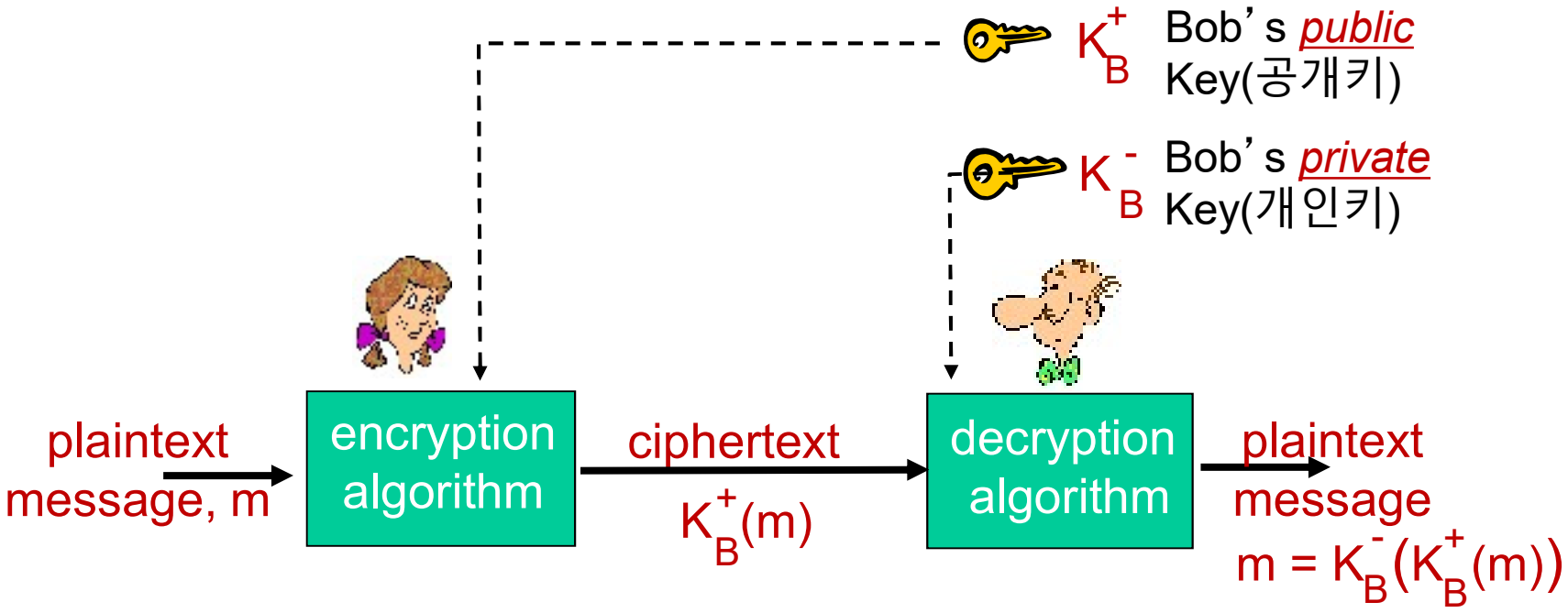
symmetric key crypto

- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never "met")?

public key crypto

- radically different approach [Diffie-Hellman76, RSA78]
- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Public key cryptography



Public key encryption algorithms

requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

$$K_B^-(K_B^+(m)) = m$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

Prerequisite: modular arithmetic

- x mod n = remainder of x when divide by n
- facts:

  [(a mod n) + (b mod n)] mod n = (a+b) mod n

  [(a mod n) - (b mod n)] mod n = (a-b) mod n

  [(a mod n) * (b mod n)] mod n = (a*b) mod n
- thus

  $(a \bmod n)^d \bmod n = a^d \bmod n$
- example: x=14, n=10, d=2:

  $(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$

  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

RSA: Creating public/private key pair

1. 두 개의 서로 다른 큰 소수 p와 q를 선택한다.
2. p와 q를 곱하여 n을 계산한다(n = p x q).
3. Ø(n)의 특징2와 특징3을 사용하여 Ø(n)을 계산한다
   (Ø(n) = Ø(p x q) = Ø(p) x Ø(q) = (p-1)(q-1)).
4. 1 < e < Ø(n)의 관계를 만족하고 Ø(n)과 서로 소의 관계인 소수 e를 선택한다.
5. e에 대한 mod Ø(n) 곱셈 역원 d를 계산한다. e가 소수이므로 역원 d가 반드시 존재하고 de = 1 mod Ø(n)와 같이 쉽게 계산된다.
   즉, (de - 1) = 0 mod Ø(n)이다.
   d는 Ø(n)보다 작으면서 가능하면 큰 수가 선택된다.
6. (n, e)를 공개키로 선택한다.
7. (n, d)를 개인키로 선택한다.

문제 : 공개키로부터 개인키 추론이 불가능한 이유는 ?

RSA: Creating public/private key pair

- p=5, q=7인 경우
  - n=35, Ø(n)=24
- e=5 선택, 5는 24와 서로소이므로 선택가능
  - 개인키 d는 24보다 작으면서 de = 1 mod 24인 값으로 선택
  - (de – 1)을 24로 나누어 나머지가 0이 되는 가장 큰 수를 d로 선택
  - (23 x 5 – 1) / 24 =
  - (22 x 5 – 1) / 24 =
  - (21 x 5 – 1) / 24 =
  - …
  - (6 x 5 – 1) / 24 =
  - (5 x 5 – 1) / 24 = 1, 나머지 = 0
  - 따라서 d = 5
- 공개키 = (35,5), 개인키 = (35,5)

RSA: encryption, decryption

1. to encrypt message m ($<n$), compute

$c = m^e \bmod n$

2. to decrypt received bit pattern, c, compute

$m = c^d \bmod n$

magic happens!

$$m = \underbrace{(m^e \bmod n)}_{c}{}^d \bmod n$$

공개키에 의한 암호화, 개인키에 의한 복호화

$c = m^e \bmod n$

$$
\begin{aligned}
m &= c^d \bmod n \\
&= (m^e \bmod n)^d \bmod n \\
&= m^{ed} \bmod n \\
&= m^{k\emptyset(n)+1} \bmod n && \text{/* } de = 1 \bmod \emptyset(n) \text{ */} \\
&= (m^{\emptyset(n)} \bmod n)^k (m \bmod n) \\
&= (1)(m \bmod n) && \text{/* 오일러의 2차정리 */} \\
&= m && \text{/* } m < n \text{ */}
\end{aligned}
$$

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

result is the same!

개인키에 의한 암호화, 공개키에 의한 복호화

$c = m^d \bmod n$

$$
\begin{aligned}
m &= c^e \bmod n \\
&= (m^d \bmod n)^e \bmod n \\
&= m^{de} \bmod n \\
&= m^{k\emptyset(n)+1} \bmod n && /* \ de = 1 \bmod \emptyset(n) \ */ \\
&= (m^{\emptyset(n)} \bmod n)^k (m \bmod n) \\
&= (1)(m \bmod n) && /* \ \text{오일러의 2차정리} \ */ \\
&= m && /* \ m < n \ */
\end{aligned}
$$

RSA in practice: session keys

- exponentiation in RSA is computationally intensive(계산이 복잡 → 작은 길이 정보 암호화에 사용 : 대칭키, 해시값)
- DES is at least 100 times faster than RSA
- use public key crypto to establish secure connection, then establish second key – symmetric session key – for encrypting data

session key, K_S

- Bob and Alice use RSA to exchange a symmetric key K_S
- (공개키로 일회성 대칭키를 암호화하여 전달)
- once both have K_S, they use symmetric key cryptography

After-study Test :

1) 다음 중 송.수신자간의 통신을 위한 공개키 암호화에 대한 설명 중 틀린 것은?
① 서로 다른 공개키와 개인키 쌍을 사용한다.
② 수신자의 공개키로 암호화할 수 있다.
③ 송신자의 개인키로 암호화할 수 있다.
④ 송신자의 공개키로 암호화할 수 있다.

2) 다음 중 공개키 암호화에 대한 설명 중 틀린 것은?
① 대용량 데이터를 효과적으로 암호화한다.
② 주로 해시값을 암호화하기 위해 사용한다.
③ 공개키 인증서가 필요하다.
④ 디지털 서명에 사용된다.

3) 다음 중 RSA 키 생성 알고리즘에 대한 설명 중 틀린 것은?
① 큰 수의 소인수 분해의 어려운 점을 활용한다.
② 큰 소수 2개를 선택하여 활용한다.
③ 오일러 2차 정리(Euler's Theorem)에 기초하여 개발되었다.
④ 공개키는 오일러 함수값(Ø(n)) 보다 작은 임의의 수를 선택한다.

4) RSA 키 생성 알고리즘에서 선택한 소수 2개가 5와 7일 경우 공개키를 (35, 5)로 선택할 때 개인키는?
① (35, 5)
② (35, 6)
③ (35, 7)
④ (35, 8)

5) RSA에서 공개키로부터 개인키를 보호할 수 있는 이유는?
① 큰 수 n에 대한 이산대수 문제의 어려움
② 큰 수 n에 대한 소인수분해 문제의 어려움
③ 큰 수 n에 대한 지수함수 문제의 어려움
④ 큰 수 n에 대한 지수 모듈로 문제의 어려움

6) 다음 중 공개키 암호화가 아닌 것은?
① SEED
② RSA
③ Diffie-Hellman
④ Elliptic Curve Crytography