2020/02/17 @ 상명대학교 2020 보안 기초 세미나

Network security Essentials

- Chapter 3 공개키 암호와 메시지 인증(2) -

박 재 형(jaehyoung@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

- 공개키 암호 원리
- 공개키 암호 알고리즘

공개키 암호 원리

- 공개키 암호
  - 정의
    - 서로 다른 두개의 키(공개키, 개인키)를 이용하여 만들어진 암호방식
      - 비대칭키 방식
  - 특징
    - 서로 다른 한쌍의 키를 사용하며 하나는 암호화 다른 하나는 복호화에 사용
      - 한쌍의 키 : 공개키, 개인키
      - 공개키로 암호화시 개인키로 복호화, 개인키로 암호화시 공개키로 복호화
    - 키 분배가 필요 없는 방식
      - 통신할때 “공개키로 암호화를 했으니 개인키로 복호화 해라”라는

공개키 암호 원리

- 대칭키 암호 방식과 공개키 암호방식 비교

구분	대칭키 암호 방식	공개키 암호 방식
키	대칭키(비밀키)	비대칭키(공개키, 개인키)
암호키 관계	암호화키 = 복호화키	암호화키 ≠ 복호화키
암호 방식	기호(문자, 비트) 대체 치환	수학적 함수 응용
장점	• 계산 속도 빠름 • 알고리즘이 다양	• 암호 키 사전 공유 불필요 • 통신 대상의 추가의 용이
단점	키 교환 및 관리의 어려움	계산 속도 느림
대표적인 예	DES, 3DES, AES	RSA

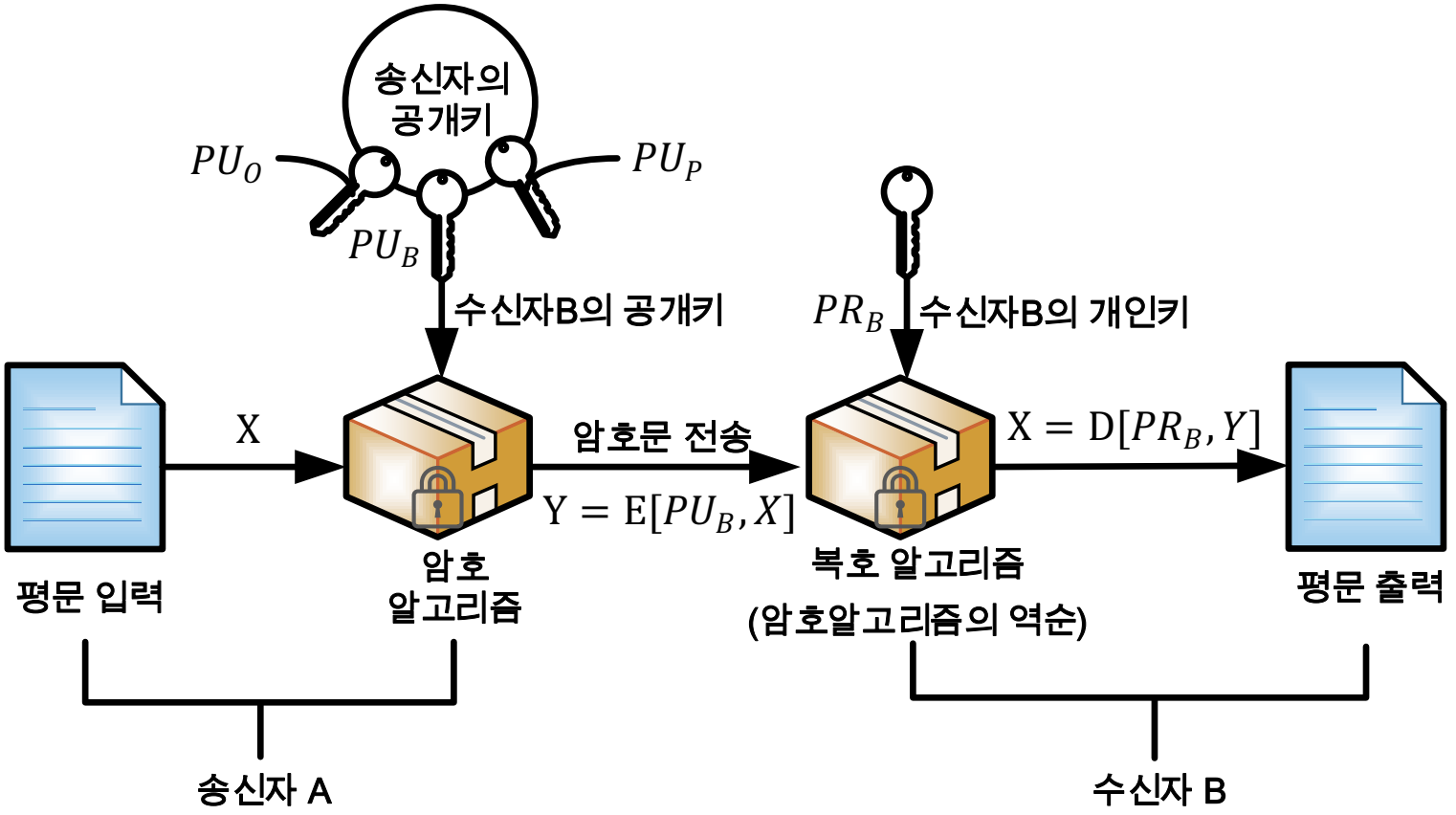
공개키 암호 원리

- 공개키 암호
  - 구조

용어	의미
X	평문으로 사람이 읽을 수 있는 메시지나 데이터로서 알고리즘의 입력으로 사용
암호 알고리즘	평문을 암호화하기 위해 사용하는 알고리즘
PR_B, PU_B	수신자 B의 한 쌍의 키(개인키, 공개키)로 한 개는 암호화에 사용되고 다른 한 개는 복호화에 사용
PR_A, PU_A	수신자 B의 한 쌍의 키(개인키, 공개키)로 한 개는 암호화에 사용되고 다른 한 개는 복호화에 사용
Y	출력으로 나오는 암호화된 메시지이며 평문과 한쌍의 키에 의한 생성
복호알고리즘	평문을 암호화 할 때 사용한 키에 대응하는 키를 이용하여 암호문을 평문으로 변환하는 알고리즘

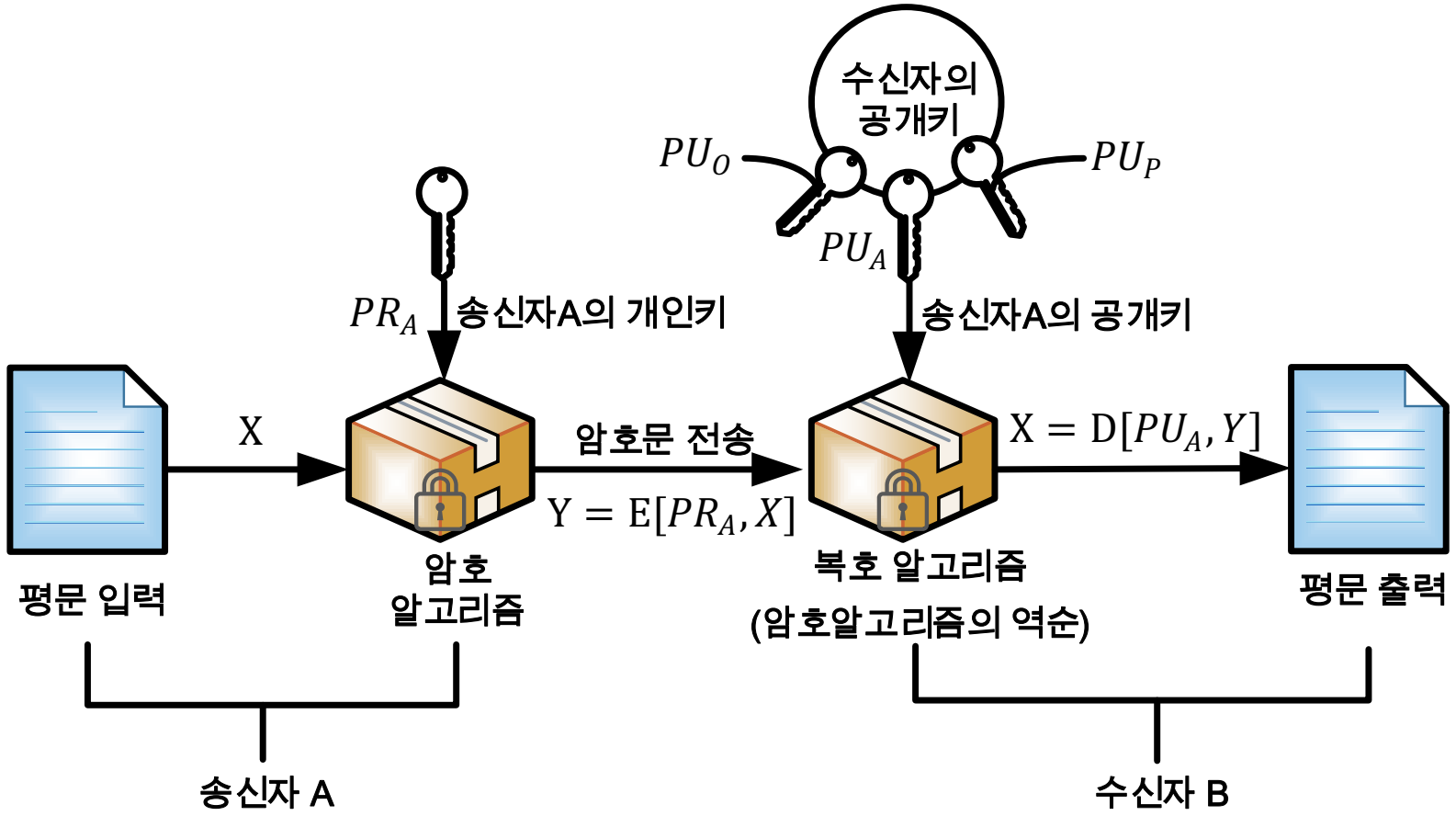
공개키 암호 원리

- 공개키 암호
  - 구조
    - 공개키에 의한 암호화



공개키 암호 원리

- 공개키 암호
  - 구조
    - 개인키에 의한 암호화



공개키 암호 원리

- 공개키 암호
  - 요건
    - 한 쌍의 키를 생성할 때 컴퓨터 계산 시간을 고려해야 함
    - 송신자가 암호화 할 때 기밀성, 가용성이 보장되어야 함
      - $C = E(PU, M)$
    - 수신자는복호화 할 때 기밀성, 가용성이 보장되어야 함
      - $M = D(PR, C) = D[PR, E(PU, M)]$
    - 공개키를 가지고 개인키를 예측할 수 없어야 함
    - 공격자가 공개키와 암호문을 알고 있더라도 해독이 불가능해야 함
    - 2개의 키 중 하나를 암호화에 사용 한다면, 다른 하나는 복호화에 사용 할 수 있어야 함

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 알고리즘
  - 정의
    - 공개키 암호 시스템의 소인수분해기반으로 한 알고리즘
      - 최소의 전자 서명
      - 1977년 MIT (Massachusetts Institute of Technology)에서 Ron Rivest와 Adi Shamir, Len Adlman이 개발한 공개키 암호 알고리즘
  - 특징
    - 공개키 암호 알고리즘의 하나
    - 소인수분해의 기반으로 이루어짐
    - 평문, 암호문, 키 모두 숫자로 이루어짐

공개키 암호 알고리즘

- RSA 알고리즘
  - 표기법

표기법	설명
M	메시지
C	암호문
p, q	키를 생성하기 위해 선택하는 소수
$n = (p \times q)$	암/복호화에 이용되는 키의 인자 값, $modulus$로 사용
e	공개키의 인자 값 (공개 값)
d	개인키의 인자 값 (비밀 값)
$PU_i = \{e, n\}$	i의 공개키
$PR_i = \{d, n\}$	i의 개인키

공개키 암호 알고리즘

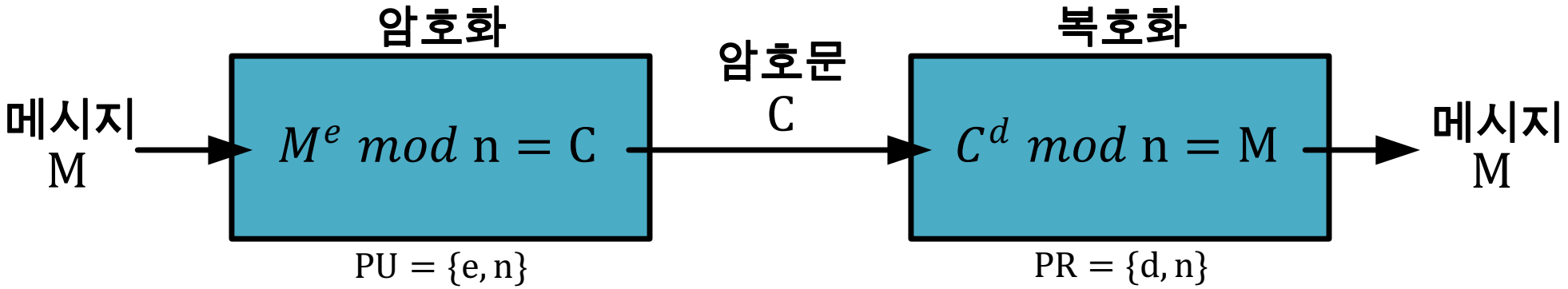
- RSA 알고리즘
  - 키생성 과정
    1. n 생성
       - 소수 p와q 선택
       - $n = p \times q$ $(p, q$는 소수$, p \neq q)$
    2. $\emptyset(n)$ 생성
       - 키쌍을 생성하기 위해 보조적으로 사용하는 수
       - $\emptyset(n)$: 오일러 함수로서 양의 정수 중 n과 서로소인 개수
       - $\emptyset(n) = (p-1,)(q-1)$
    3. 정수 e를 선택 {공개키 = $(e, \emptyset(n))$}
       - $\gcd(\emptyset(n), e) = 1;\ [1 < \mathrm{e} < \emptyset(n)]$
         - gcd (greatest common divisor): e 와 $\emptyset(n)$ 의 서로소
         - 서로소: 두수의 최대공약수가 1인 수
           - d의 존재를 보증

공개키 암호 알고리즘

- RSA 알고리즘
  - 키생성 과정
    4. d 생성 {개인키 = $(d, \emptyset(n))$}
       - d는 e로부터 계산
       - $d \times e \; mod \; \emptyset(n) = 1; \; [1 < d < \emptyset(n)]$
         - 암호문을 복호화하면 원래의 평문으로 돌아가는 것을 보장

공개키 암호 알고리즘

- RSA 알고리즘
  - 구조



    - 암호화
      - $C = M^e \ mod \ n$
      - 메시지 M을 공개키 $PU = \{e, n\}$로 암호화
    - 복호화
      - $\mathrm{M} = C^d \ mod \ n$
      - 암호문 C를 개인키 $\mathrm{P}R = \{d, n\}$로 복호화

공개키 암호 알고리즘

- RSA 알고리즘
  - 키생성의 예
    1. 두 소수 $p = 17$ 과 $q = 19$을 생성
    2. $n = pq = 17\times19 = 323$을 계산
    3. $\emptyset(n) = (p-1)(q-1) = 16\times18 = 144$를 계산
    4. $\gcd\big(e, \emptyset(n)\big) = \gcd\big(e, \emptyset(n)\big) = 1$이 되는 $e = 5$ 선택
       - e.g., $e = 5, 7, 11, 13, 17, 19, 23, \cdots$
    5. 공개키는 $(e, n) = (5, 323)$
    6. d는 $e\times d \; mod \; \emptyset(n) = 1$을 만족해야 함
    7. $e\times d \; mod \; \emptyset(n) = 5\times d \; mod \; 144$
       $= 145 \; mod \; 144 = 1$
       $\therefore d = 29$
    8. 개인키는 $(d, n) = (29, 323)$

공개키 암호 알고리즘

- RSA 알고리즘
  - 암호화
    - e.g., 평문은 $n < 323$인 수,평문 123을 암호화
      - 평문$^{e}\ mod\ n = 123^{5}\ mod\ 323 = 225$
      - 암호문은 225
  - 복호화
    - e.g.,암호문 225를 복호화
      - 복호화에서 개인키 $d = 29, n = 323$ 사용
      - 암호문$^{d}\ mod\ n = \ 225^{29}\ mod\ 323 = 123$
      - 평문은 123

공개키 암호 알고리즘

- RSA 알고리즘
  - 장점
    - 대칭키 암호화보다 더 키 교환 성능이 뛰어남
    - 인증과 부인 봉쇄를 제공
  - 단점
    - 대칭키 암호화보다 느리게 동작
    - bit수가 많고 수행시간이 김
    - 수학적으로 많은 연산 작업

공개키 암호 알고리즘

- RSA 알고리즘
  - 보안
    - 전수조사 공격
      - 가능한 모든 경우의 개인키를 시도해보는 공격
      - 대응책
        - e 와 d의 비트 크기를 최소 512 bits가 되도록 함
          - 10진수로 약 154자리 수
    - 소인수 분해 공격
      - n을 소인수분해 한 p와q를 구하여 키 값을 얻는 공격
      - 대응책
        - n의 bits 크기를 최소 1024 bits가 되도록 함
          - 10진수로 약 300자리수

공개키 암호 알고리즘

- Diffie-Hellman 키교환
  - 정의
    - 상호간에 대칭키 비밀키를 교환하는 알고리즘
  - 특징
    - 공개키를 교환하여 양측이 사용할 비밀키 생성
    - 대칭키를 공유하는데 사용됨
    - 이산대수(이산로그)방식 이용

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 이산대수 문제 (Discrete logarithms problem)
    - 원시근 (Primitive root) α
      - 소수 q의 원시근
        - 자신의 거급 제곱을 이용하여 1부터 $q-1$까지의 정수를 생성해 낼 수 있는 수
        - $\alpha \bmod q,\ \alpha^2 \bmod q,\ \cdots,\ \alpha^{p-1} \bmod q$
      - 예시
        - $q = 7$에서
        - $3^1 \bmod 7 = 3$
        - $3^2 \bmod 7 = 9 \bmod 7 = 2$
        - $3^3 \bmod 7 = 27 \bmod 7 = 6$
        - $3^4 \bmod 7 = 81 \bmod 7 = 4$
        - $3^5 \bmod 7 = 243 \bmod 7 = 5$
        - $3^6 \bmod 7 = 729 \bmod 7 = 1$
        - 3은 q의 원시근

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 이산대수 문제 (Discrete logarithms problem)
    - 이산대수 (Discrete logarithm)
      - $a^x = b\ (1 < a, x), (b$는 정수 $)$일 때 만족하는 x를 가리킴

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 비밀키 생성 과정
    - 공개되는 값
      - 소수 q, 원시근 α, 공개값 Y_A, Y_B

1. 통신 양측은 임의의 개인값 X_A, X_B 선택
2. 공개값 Y_A, Y_B 계산

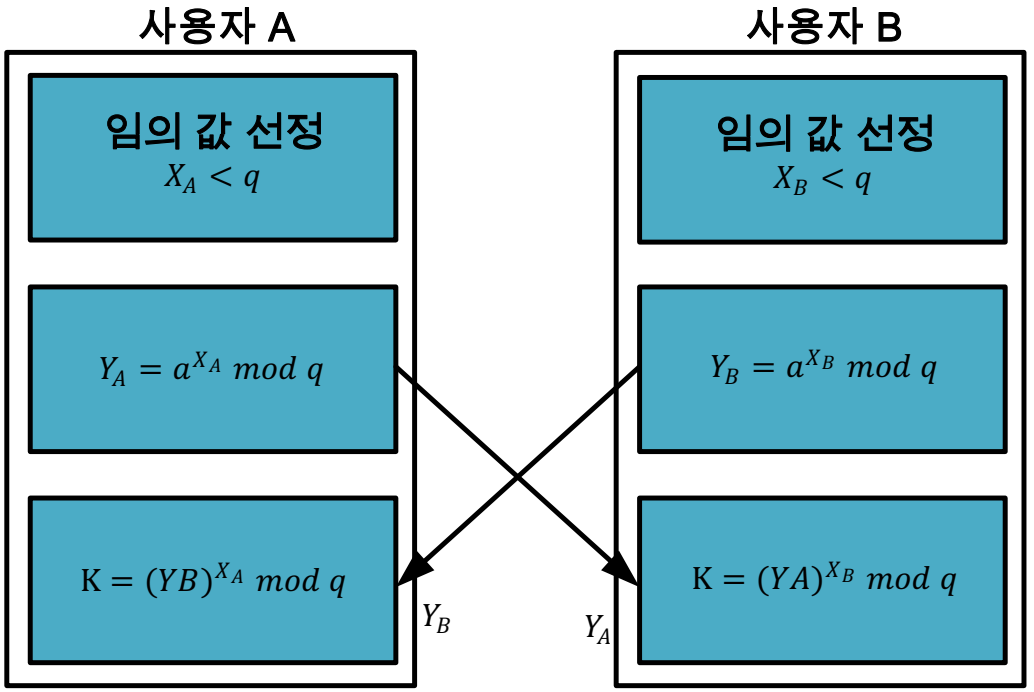
$$Y_A = \alpha^{X_A} \bmod q$$

$$Y_B = \alpha^{X_B} \bmod q$$

3. 계산한 공개 값 전송
4. 비밀키 생성

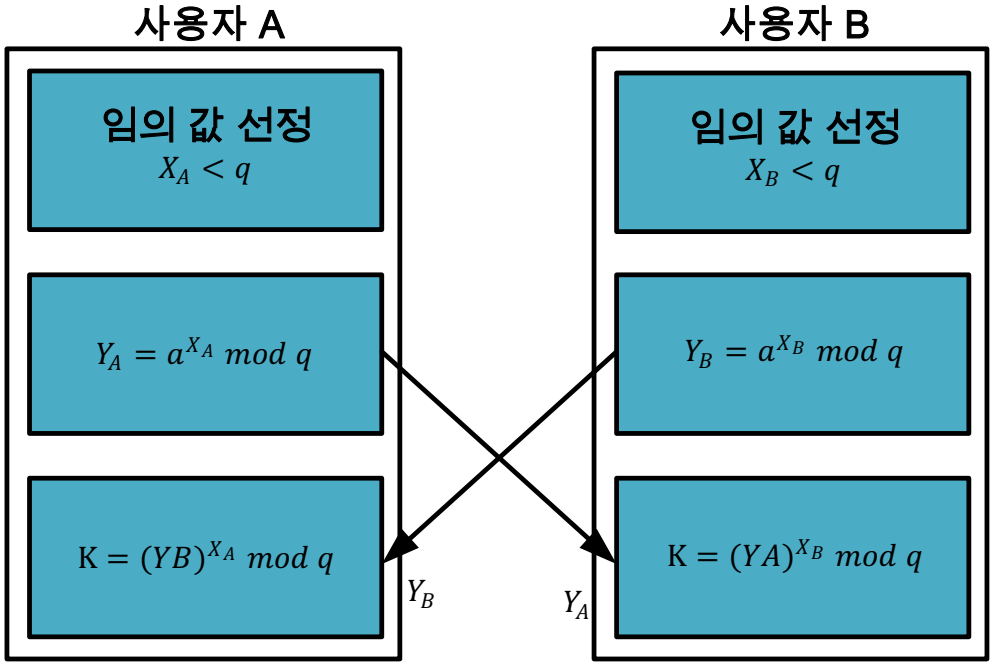
$$\mathrm{K} = (Y_B)^{X_A} \bmod q$$

$$= (Y_A)^{X_B} \bmod q$$



공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 비밀키 생성 과정
    - 동일한 비밀키
      - $K = (Y_B)^{X_A} \ mod \ q$
        $= (\alpha^{X_B} \ mod \ q)^{X_A} \ mod \ q$
        $= (\alpha^{X_B})^{X_A} \ mod \ q$
        $= a^{X_B X_A} \ mod \ q$
        $= (\alpha^{X_A})^{X_B} \ mod \ q$
        $= (\alpha^{X_A} \ mod \ q)^{X_B} \ mod \ q$
        $= (Y_A)^{X_B} \ mod \ q$



공개키 암호 알고리즘

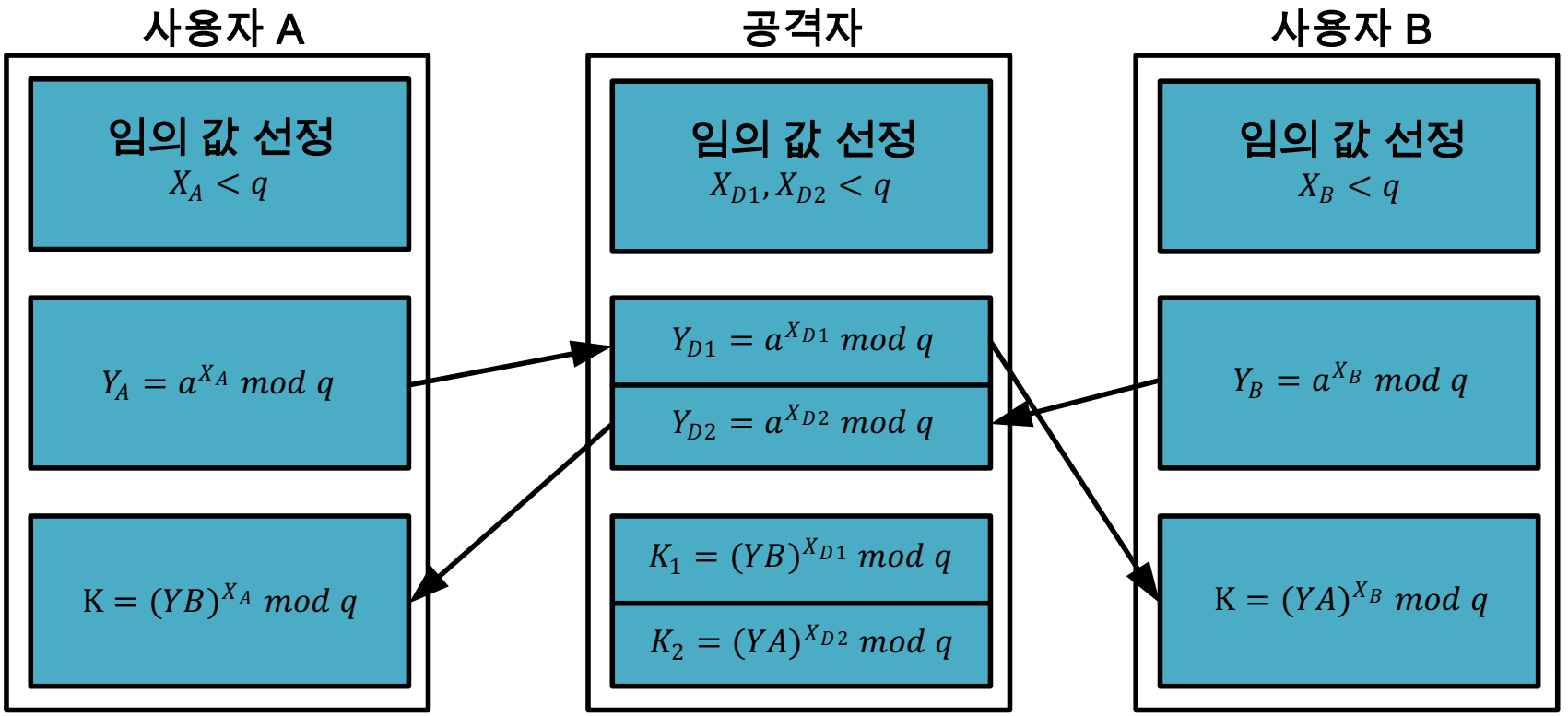
- Diffie-Hellman 알고리즘
  - 이산대수 문제 (Discrete logarithms problem)
    - 예제
      - $7^x \ mod \ 13 = 8$이 되는 x값은 무엇인가?
        - $7^0 \ mod \ 13 = 1$
        - $7^1 \ mod \ 13 = 7$
        - $7^2 \ mod \ 13 = 10$
        - $7^3 \ mod \ 13 = 5$
        - $7^4 \ mod \ 13 = 9$
        - $7^5 \ mod \ 13 = 11$
        - $7^6 \ mod \ 13 = 12$
        - $7^7 \ mod \ 13 = 6$
        - $7^8 \ mod \ 13 = 3$
        - $7^9 \ mod \ 13 = 8$
        - $\therefore x = 9$

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 소극적 공격
    - 이산대수 공격 (Discrete Logarithm Attack)
      - 공격자가 M_1, M_2를 가로 챘다고 가정, $M_1 = a^{X_A} \, mod \, q$에서 X_A를 구하고 $M_2 = X^{X_B} \, mod \, q$에서 X_B를 구한다면 공유키 $\mathrm{K} = a^{X_A X_B} \, mod \, q$해독 가능
        - 대처하기 위해 소수 q는 최소 십진수로 300자리 이상의 수가 되어야 함
  - 적극적 공격
    - 중간자 공격 (Man-in-the-Middle Attack)
      - 공격자가 송신자 수신자 사이에서 송/수신자 행세하는 공격
        - 두 통신자 사이에 인증을 제공하지 못함

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 중간자 공격



공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - RSA와 Diffie-Hellman 알고리즘 구분 표

구분	RSA	Diffie-Hellman
수학적 배경	소인수 분해 문제	이산대수 문제
키 분배 방법	키 전달	키 합의
응용 분야	암/복호화, 디지털 서명, 키 교환	키 교환
장점	여러 라이브러리 존재 (응용 범위가 좋음)	키 분배에 최적화,필요시 생성
단점	느린 계산 속도	중간자 공격에 취약

- 라이브러리: 소프트웨어 개발에 사용되는 하부 프로그램

공개키 암호 알고리즘

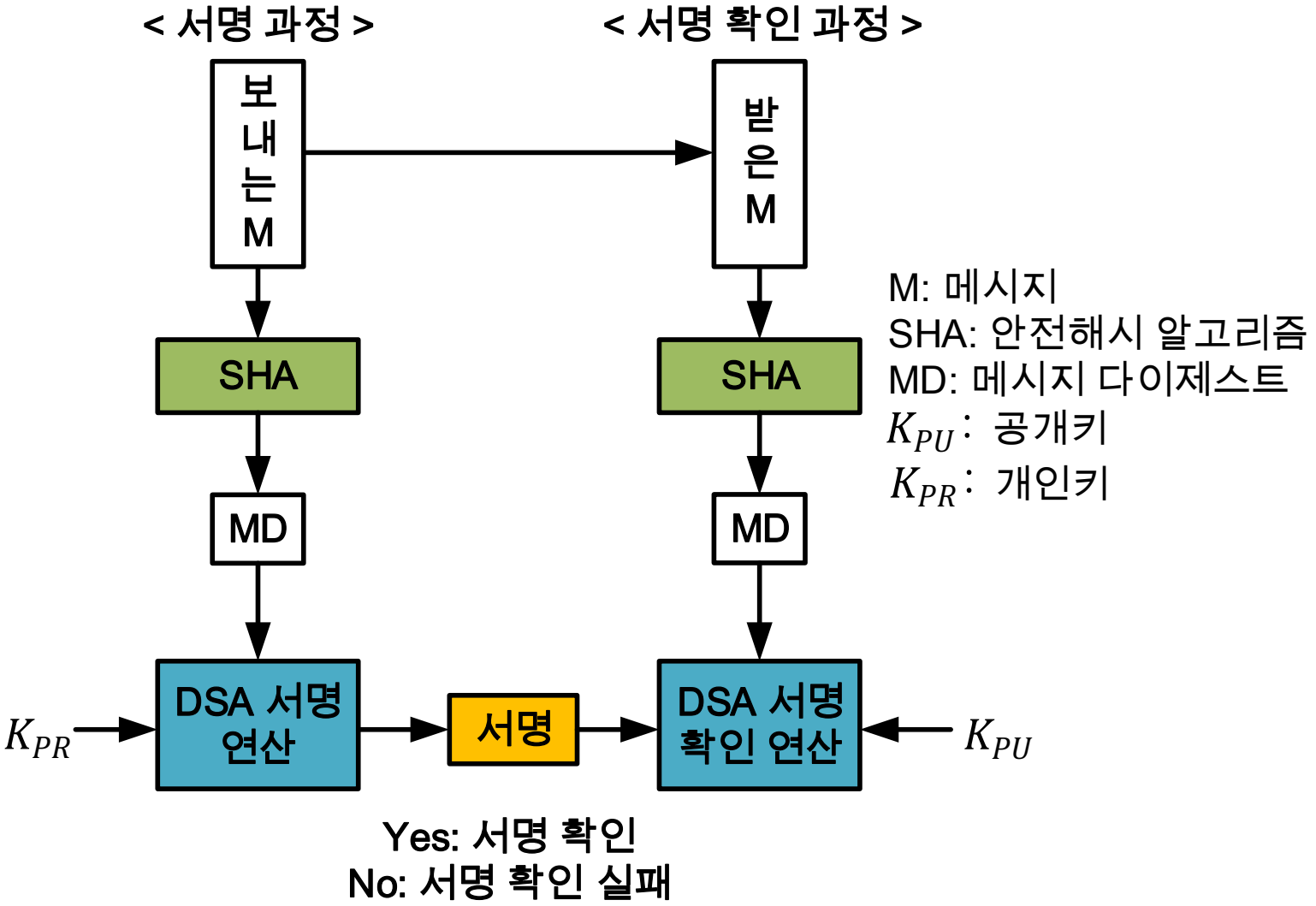
- 기타알고리즘
  - 디지털 서명 표준 (DSS: Digital Signature Standard)
    - DSS는 디지털 서명의 표준이고 이 표준안에서 특정 알고리즘DSA(HMAC, RSA 등)사용
      - DSA: 오직 디지털 서명 기능만 제공하도록 설계한 알고리즘
        - 암호, 키교환 사용 불가

공개키 암호 알고리즘

- 기타 알고리즘
  - 디지털 서명 (Digital Signature)
    - 송신자의 신원을 증명하는 인증 기법
    - 특징
      - 송신자가 송신할 메시지에 서명하는 방법을 사용
        - 서명은 개인키로 암호화 됨
      - 공개키로 복호화하기 때문에 기밀성을 보장하지 않음
        - 메시지는 평문상태로 전달
      - 무결성, 인증, 부인봉쇄 를 제공

공개키 암호 알고리즘

- 기타알고리즘
  - DSA
    - 과정

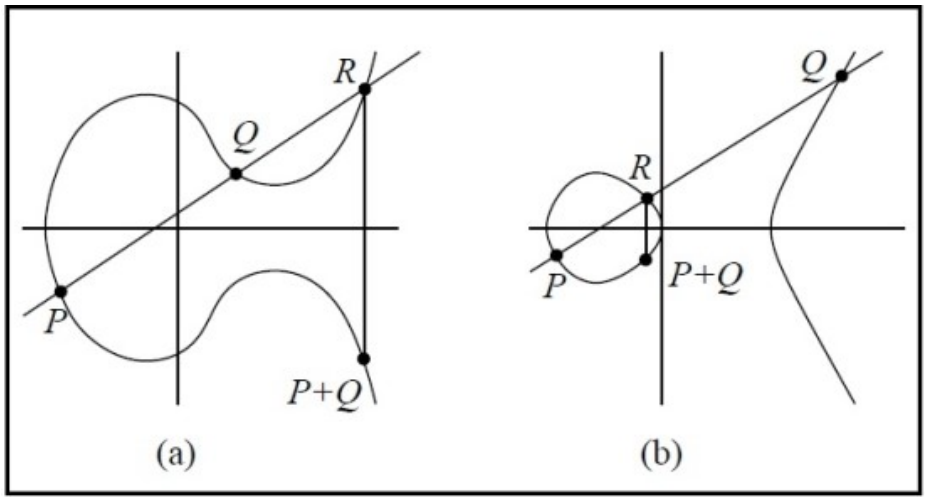


공개키 암호 알고리즘

- 기타알고리즘
  - 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)
    - 정의
      - 1985년 Neal Koblitz와 Victor Miller가 독립적으로 제안한 타원 곡선 이론 기반 공개키 암호 방식
    - 특징
      - 고정된 실수인 a, b가 있을 때, 방성식 $y^2 = x^3 + ax + b$를 만족하는 (x, y)들의 집합을 사용
      - RSA보다 짧은 키 길이를 사용하면서 비슷한 수준의 안전성을 제공
      - 하드웨어와 소프트웨어로 구현하기가 용이 함

공개키 암호 알고리즘

- 기타알고리즘
  - 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)
    - $y^2 = x^3 + ax + b$를 만족하는 P보다 작은 소수를 개인키로 생성(d)
    - 타원곡선의 더하기연산을 통하여 공개키 생성(f)
      - G: 타원곡선상 임의의 점
        d x G = f



공개키 암호 알고리즘

- 기타알고리즘
  - 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)
    - 예제
      - x, y가 유한 할 때
      - 타원곡선 군 범위가 23일 때
      - $y^2 = x^3 + x$ 에서 x가 11이면(개인키),
        $y^2 \; mod \; 23 = (1331 + 11) mod \; 23 = 1342 \; mod \; 23 = 8$
        $y^2 \; mod \; 23 = 8$
      - 식을 만족하는 y는 10과 13임(공개키)
      - 타원곡선 군의 범위가 커지면 y가 가지는 값이 많아짐
        - 군: 타원곡선상에 나타나는 유한한 임의의 점들의 집합

공개키 암호 알고리즘

- 공개키 암호
  - 공개키 알고리즘 비교표

알고리즘	암호화/복호화	디지털 서명	키 교환
RSA	O	O	O
Diffie-Hellman	X	X	O
DSS	X	O	X
ECC	O	O	O

Thanks!

박 재 형 (jaehyoung@pel.smuc.ac.kr)