

Network security Essentials

- Chapter 3 공개키 암호와 메시지 인증(2) -

박재형(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

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 \neq 복호화키
암호 방식	기호(문자, 비트) 대체 치환	수학적 함수 응용
장점	<ul style="list-style-type: none">계산 속도 빠름알고리즘이 다양	<ul style="list-style-type: none">암호 키 사전 공유 불필요통신 대상의 추가의 용이
단점	키 분배 및 관리의 어려움	계산 속도 느림
대표적인 예	DES, 3DES, AES	RSA

공개키 암호 원리

- 공개키 암호
- 구조

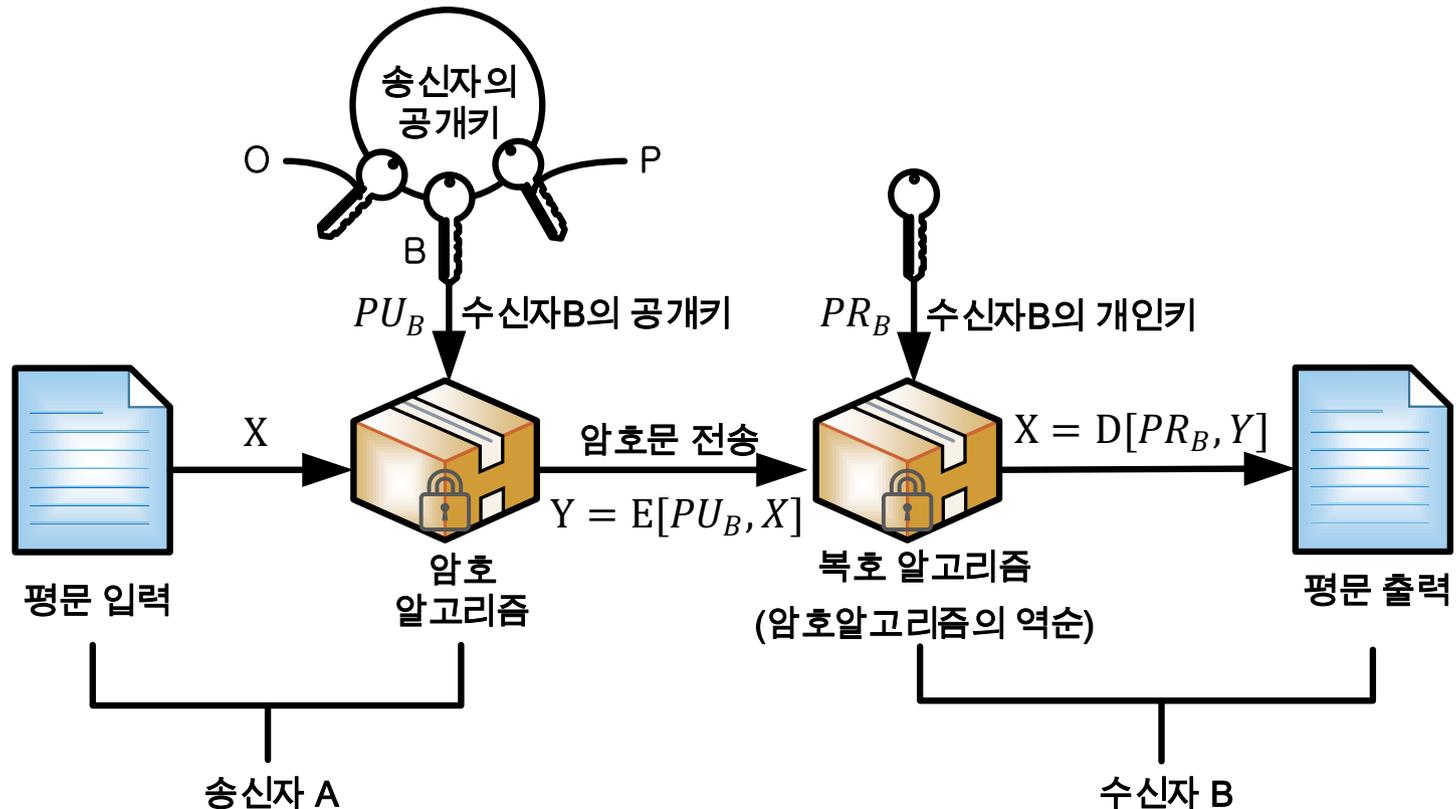
용어	의미
평문	사람이 읽을 수 있는 메시지나 데이터로서 알고리즘의 입력으로 사용
암호 알고리즘	평문을 암호화하기 위해 사용하는 알고리즘
공개키와 개인키	한 쌍의 키(공개키, 개인키)로 이루어져있고 한 개는 암호화에 사용되고 다른 한 개는 복호화에 사용
암호문	출력으로 나오는 암호화된 메시지이며 평문과 한쌍의 키에 의한 생성
복호 알고리즘	평문을 암호화 할 때 사용한 키에 대응하는 키를 이용하여 암호문을 평문으로 변환하는 알고리즘

공개키 암호 원리

- 공개키 암호

- 구조

- 공개키에 의한 암호화

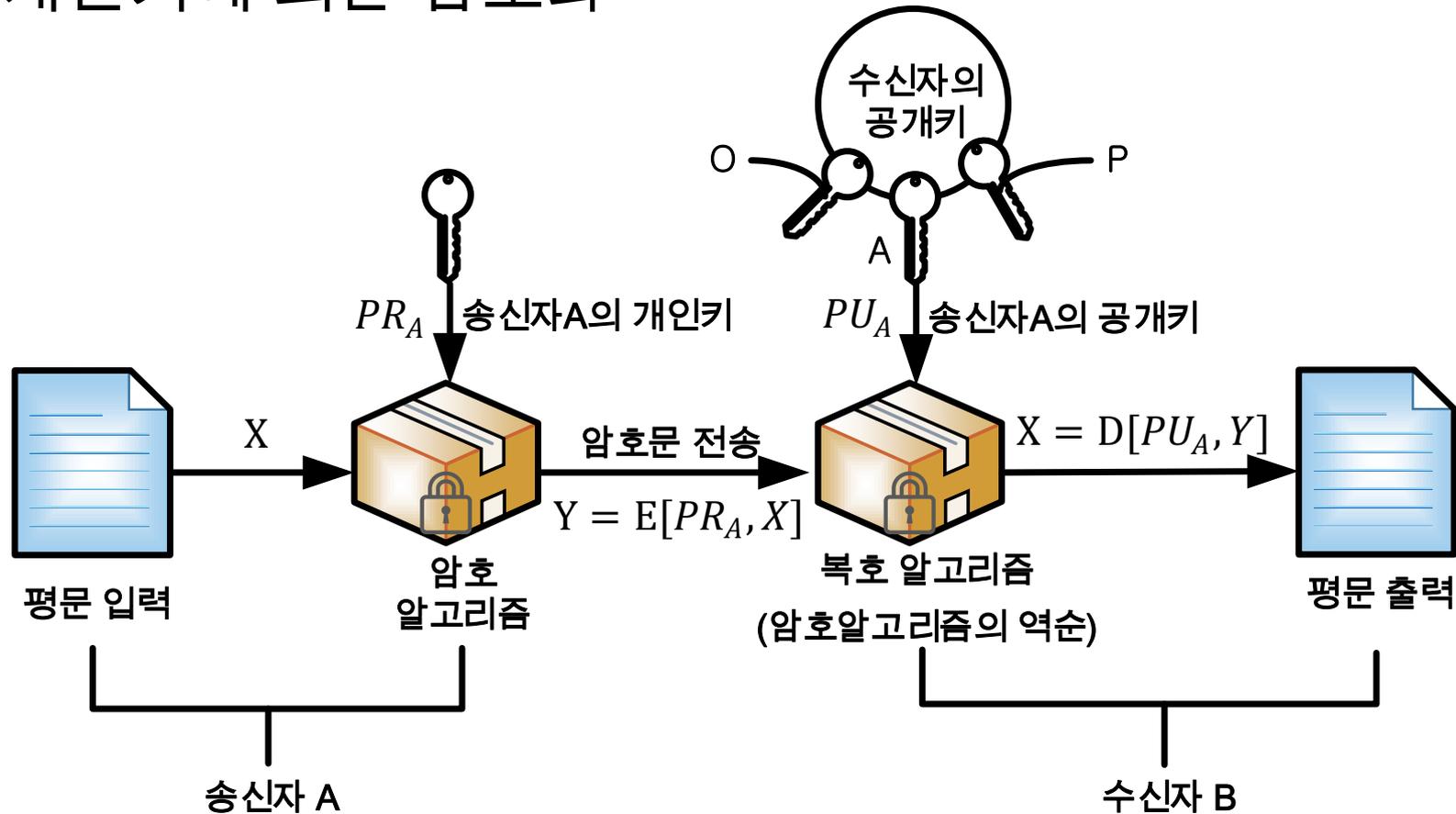


공개키 암호 원리

- 공개키 암호

- 구조

- 개인키에 의한 암호화



공개키 암호 원리

• 공개키 암호

• 요건

- 한 쌍의 키를 생성하는 것이 쉬워야 함
- 송신자는 암호문을 쉽게 구할 수 있어야 함
 - $C = E(PU, M)$
- 수신자는 암호문을 복호화하는 것이 쉬워야 함
 - $M = D(PR, C) = D[PR, E(PU, M)]$
- 공개키를 가지고 개인키를 예측할 수 없어야 함
- 공개키와 암호문을 알고 있더라도 해독이 불가능해야 함
- 암호/복호화에 키가 하나씩 사용되어야 함
 - $E[PU, D(PR, M)], D[PR, E(PU, M)]$
- 평문은 키 값보다 작아야 함
 - $C = M^e \text{ mod } n (M < n)$

공개키 암호 원리

- 공개키 암호
- 응용
 - 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

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, <i>modulus</i> 로 사용
e	공개키의 인자 값 (공개 값)
d	개인키의 인자 값 (비밀 값)
$PU = \{e, n\}$	공개키
$PU = \{d, n\}$	개인키

공개키 암호 알고리즘

- RSA 알고리즘

- 키생성 과정

1. n 생성

- 소수 p 와 q 선택
- $n = p \times q$ (p, q 는 소수, $p \neq q$)

2. $\phi(n)$ 생성

- 키쌍을 생성하기 위해 보조적으로 사용하는 수
- $\phi(n) = (p - 1)(q - 1)$

3. 정수 e 를 선택 {공개키 = $(e, \phi(n))$ }

- $\gcd(\phi(n), e) = 1; [1 < e < \phi(n)]$
 - e 와 $\phi(n)$ 의 최대 공약수 gcd (greatest common divisor)는 1
 - d 의 존재를 보증하기 위해 필요한 조건

공개키 암호 알고리즘

- RSA 알고리즘

- 키생성 과정

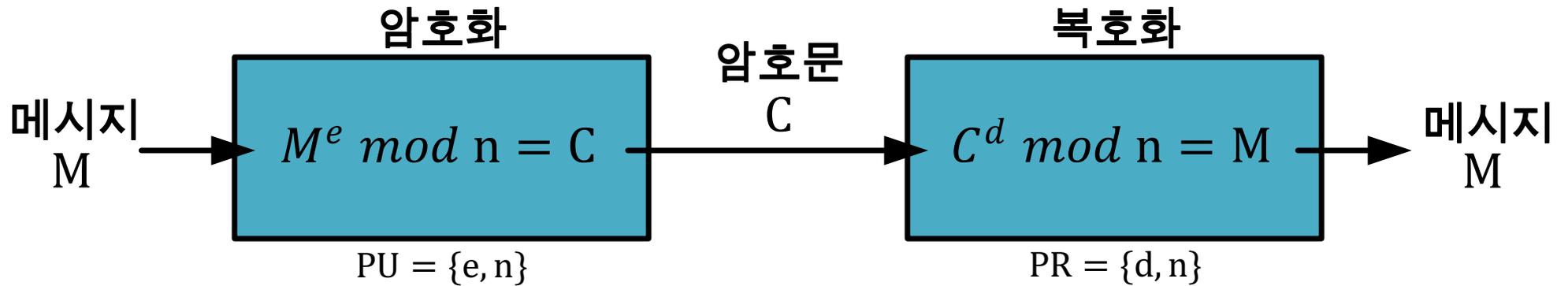
- 4. d 생성 {개인키 = $(d, \phi(n))$ }

- d 는 e 로부터 계산
 - $d \times e \bmod \phi(n) = 1; [1 < d < \phi(n)]$
 - 암호문을 복호화하면 원래의 평문으로 돌아가는 것을 보장

공개키 암호 알고리즘

- RSA 알고리즘

- 구조



- 암호화

- $C = M^e \pmod n$
 - 메시지 M 을 공개키 $PU = \{e, n\}$ 로 암호화

- 복호화

- $M = C^d \pmod n$
 - 암호문 C 를 공개키 $PR = \{d, n\}$ 로 복호화

공개키 암호 알고리즘

• RSA 알고리즘

• 키생성의 예

1. 두 소수 $p = 17$ 과 $q = 19$ 을 생성
2. $n = pq = 17 \times 19 = 323$ 을 계산
3. $\phi(n) = (p - 1)(q - 1) = 16 \times 18 = 144$ 을 계산
4. $\gcd(e, \phi(n)) = \gcd(e, \phi(n)) = 1$ 이 되는 $e = 5$ 선택
 - e.g., $e = 5, 7, 11, 13, 17, 19, 23, \dots$
5. 공개키는 $(e, n) = (5, 323)$
6. d 는 $e \times d \bmod \phi(n) = 1$ 을 만족해야 함
7. $e \times d \bmod \phi(n) = 5 \times 29 \bmod 144$
 $= 145 \bmod 144 = 1$
 $\therefore d = 29$
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 $n = 123^5 \text{ mod } 323 = 225$
 - 암호문은 225

- 복호화

- e.g., 암호문 225를 복호화
 - 복호화에서 개인키 $d = 29, n = 323$ 사용
 - 암호문^d mod $n = 225^{29} \text{ mod } 323 = 123$
 - 평문은 123

공개키 암호 알고리즘

- RSA 알고리즘

- 장점

- 대칭키 암호화보다 더 키 분배 성능이 뛰어남
 - 인증과 부인 봉쇄를 제공
 - 개인키의 사용

- 단점

- 대칭키 암호화보다 느리게 동작
 - bit수가 많고 수행시간이 김
 - 수학적으로 집약적인 작업

공개키 암호 알고리즘

- RSA 알고리즘

- 보안

- 전수조사 공격

- 가능한 모든 경우의 개인키를 시도해보는 공격

- 대응책

- e 와 d 의 비트 크기를 최소 512 bits가 되도록 함
 - 10진수로 약 154자리 수

- 소인수 분해 공격

- n 을 소인수분해 한 p 와 q 를 구하여 키 값을 얻는 공격

- 대응책

- n 의 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 p 의 원시근

- 자신의 거듭 제곱을 이용하여 1부터 $p - 1$ 까지의 정수를 생성해 낼 수 있는 수

- $\alpha \bmod p, \alpha^2 \bmod p, \dots, \alpha^{p-1} \bmod p$

- 이산대수 (Discrete logarithm)

- p 보다 작은 임의의 정수 b 와 p 의 원시근 α 에 대해서

- $b = \alpha^i \bmod p$ ($0 \leq i \leq p - 1$)

- 밀수 α 를 갖는 b 의 지수 (Exponent) i 를 이산대수라고 함

- 표기: $\log_{\alpha, p}(b)$

공개키 암호 알고리즘

• Diffie-Hellman 알고리즘

• 비밀키 생성 과정

• 공개되는 값

- 소수 q , 원시근 α , 공개값 Y_A, Y_B
 - 원시근: 거듭제곱을 이용해 생성한 수

1. 통신 양측은 임의의 개인값 X_A, X_B 선택

2. 공개값 Y_A, Y_B 계산

$$Y_A = \alpha^{X_A} \text{ mod } q$$

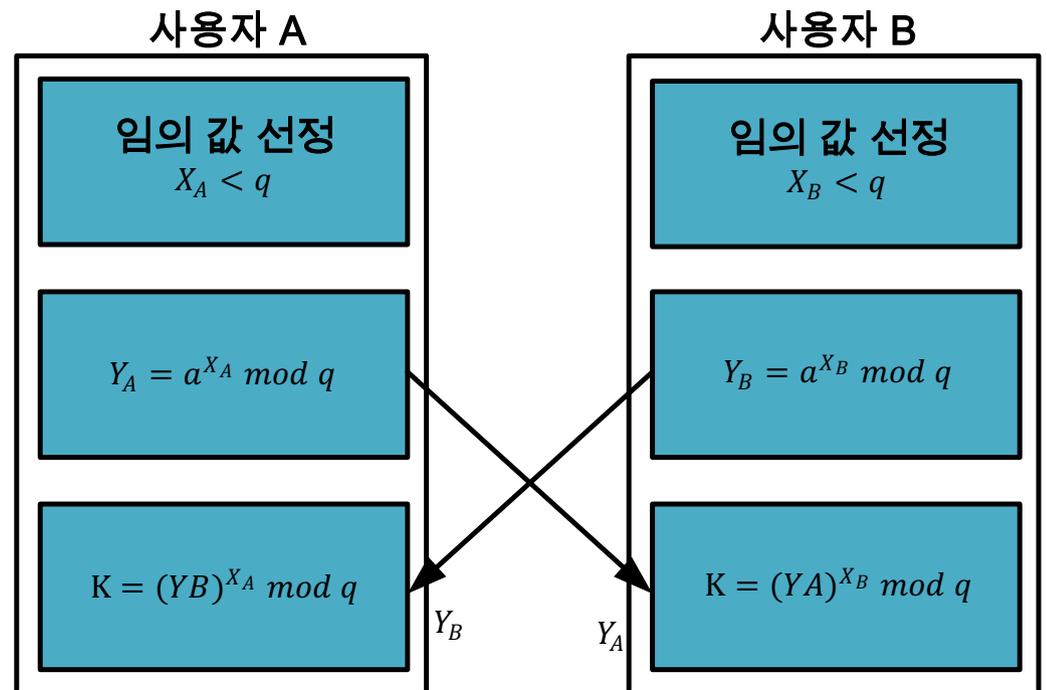
$$Y_B = \alpha^{X_B} \text{ mod } q$$

3. 계산한 공개 값 전송

4. 비밀키 생성

$$K = (Y_B)^{X_A} \text{ mod } q$$

$$= (Y_A)^{X_B} \text{ mod } q$$



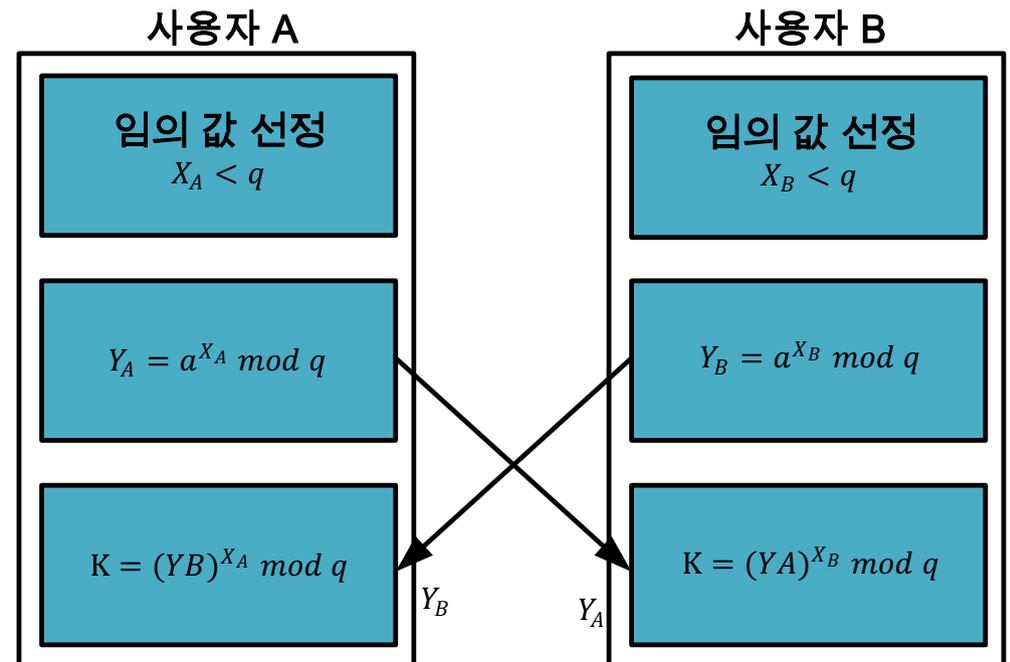
공개키 암호 알고리즘

• Diffie-Hellman 알고리즘

• 비밀키 생성 과정

• 동일한 비밀키

- $K = (Y_B)^{X_A} \text{ mod } q$
 $= (\alpha^{X_B} \text{ mod } q)^{X_A} \text{ mod } q$
 $= (\alpha^{X_B})^{X_A} \text{ mod } q$
 $= a^{X_B X_A} \text{ mod } q$
 $= (\alpha^{X_A})^{X_B} \text{ mod } q$
 $= (\alpha^{X_A} \text{ mod } q)^{X_B} \text{ mod } q$
 $= (Y_A)^{X_B} \text{ mod } q$



공개키 암호 알고리즘

- Diffie-Hellman 알고리즘

- 이산대수 문제 (Discrete logarithms problem)

- 예제

- $7^x \bmod 13 = 8$ 이 되는 x 값은 무엇인가?

- $7^0 \bmod 13 = 1$

- $7^1 \bmod 13 = 7$

- $7^2 \bmod 13 = 10$

- $7^3 \bmod 13 = 5$

- $7^4 \bmod 13 = 9$

- $7^5 \bmod 13 = 11$

- $7^6 \bmod 13 = 12$

- $7^7 \bmod 13 = 6$

- $7^8 \bmod 13 = 3$

- $7^9 \bmod 13 = 8$

- $\therefore x = 9$

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘

- 소극적 공격

- 이산대수 공격 (Discrete Logarithm Attack)

- 공격자가 M_1, M_2 를 가로챘다고 가정, $M_1 = g^a \text{ mod } q$ 에서 a 를 구하고 $M_2 = g^b \text{ mod } q$ 에서 b 를 구한다면 공유키 $K = g^{ab} \text{ mod } q$ 해독 가능

- 대처하기 위해 소수 q 는 최소 십진수로 300자리 이상의 수가 되어야 함

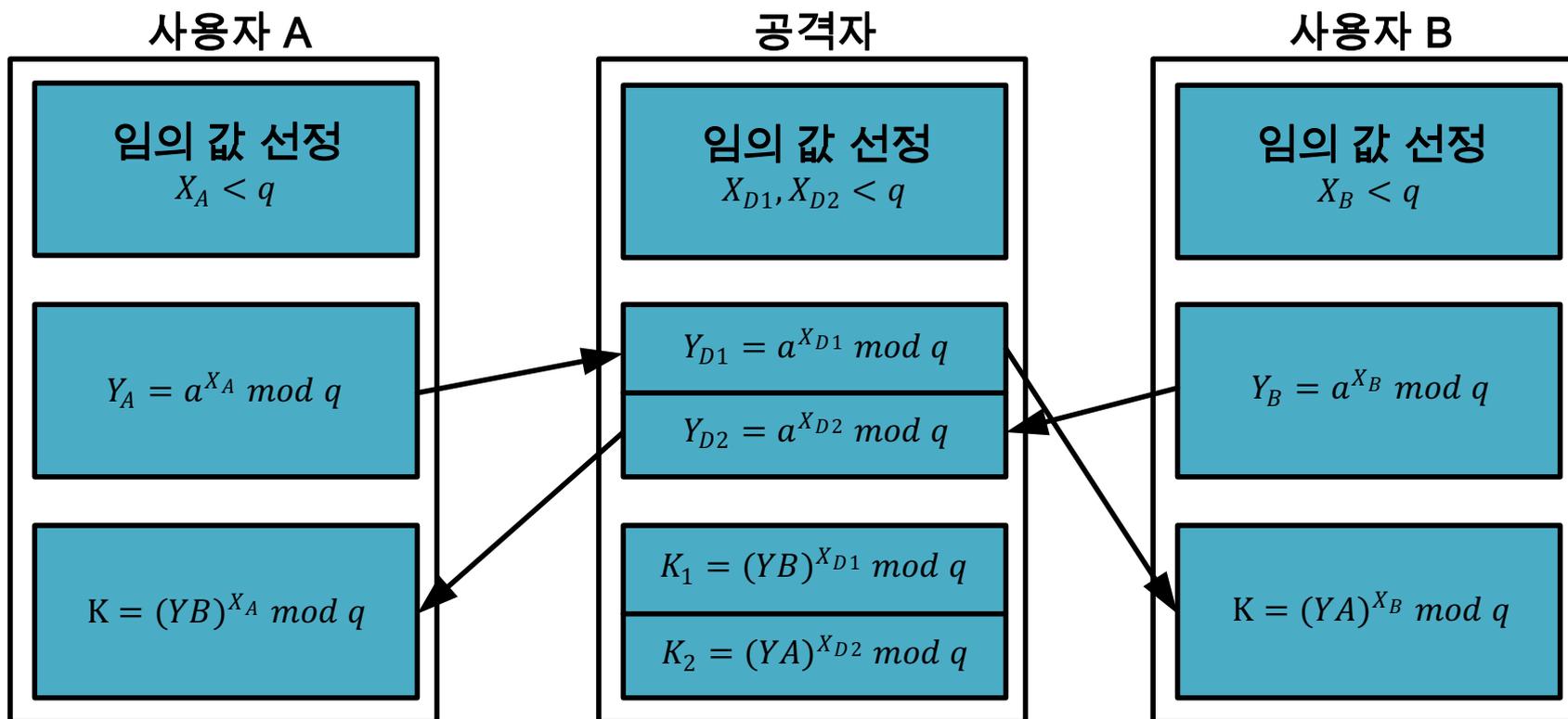
- 적극적 공격

- 중간자 공격 (Man-in-the-Middle Attack)

- 기밀성에 대한 매우 유효한 공격 방법
 - 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 비교 표

구분	RSA	Diffie-Hellman
수학적 배경	소인수 분해 문제	이산대수 문제
키 분배 방법	키전송	키 합의
응용 분야	암/복호화, 디지털 서명, 키 교환	키 교환
장점	여러 라이브러리 존재	키 분배에 최적화 키는 필요시 에만 생성, 저장 불필요
단점	느린 계산 속도	위조에 취약

공개키 암호 알고리즘

- 기타 알고리즘
 - 디지털 서명 표준 (DSS: Digital Signature Standard)
 - DSS는 디지털 서명의 표준이고 이 표준안에서 특정 알고리즘(DSA, HMAC, RSA)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

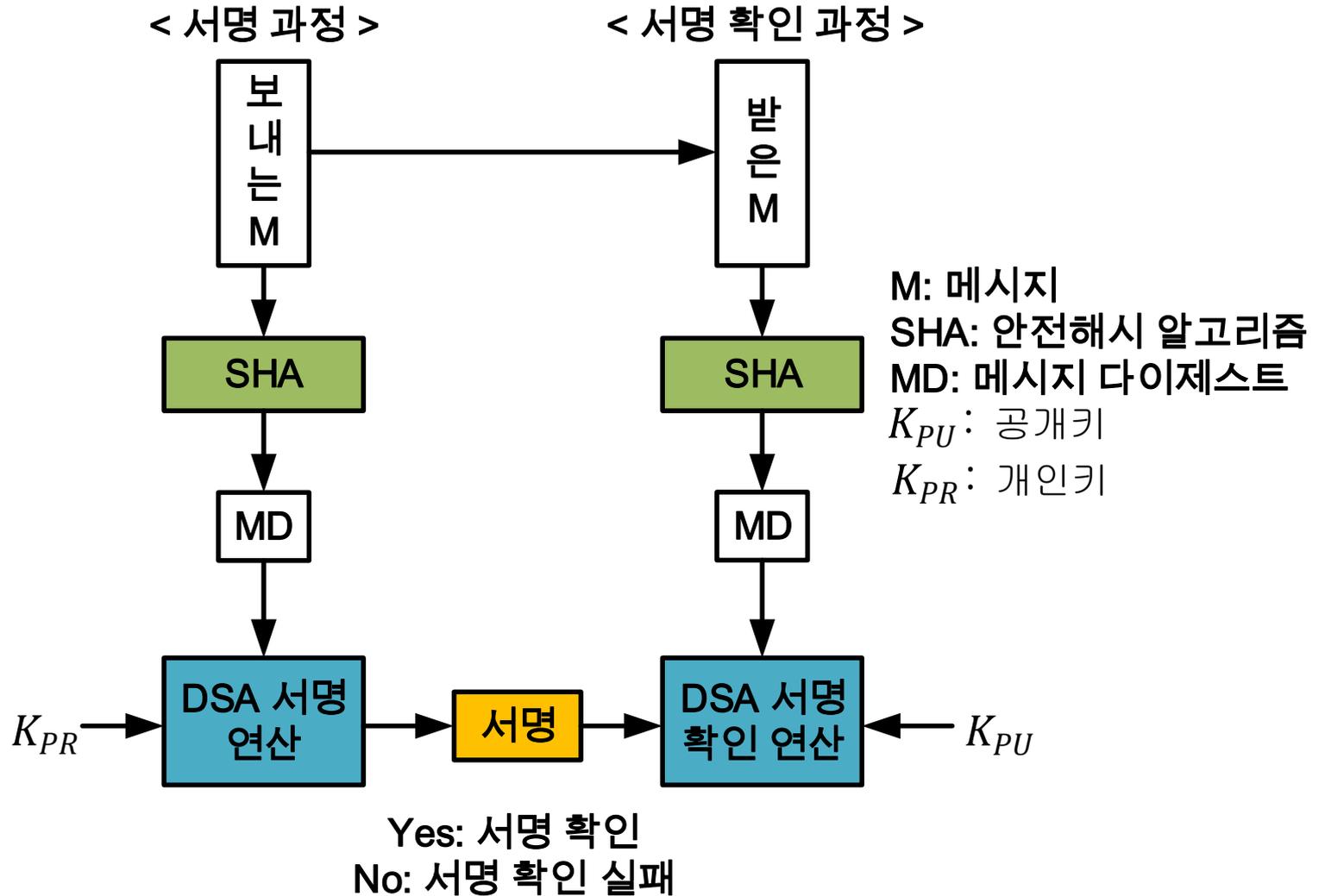
- 송신자가 자신의 개인키로 암호화 한 메시지를 수신자가 송신자의 공개키로 복호화
- 디지털 서명만 제공하는 기법으로 인증자 블록을 만듦
 - 인증자
 - 메시지의 기능을 대신하는 작은 블록
 - 인증자를 개인키로 암호화하여 메시지의 출처, 무결성, 순서를 확인해주는 서명 생성
- 공개키로 복호화하기 때문에 기밀성을 보장하지 않음

공개키 암호 알고리즘

- 기타 알고리즘

- DSA

- 과정



공개키 암호 알고리즘

- 기타 알고리즘

- 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)

- 정의

- 1985년 Neal Koblitz와 Victor Miller가 독립적으로 제안한 타원 곡선 이론 기반 공개키 암호 방식

- 특징

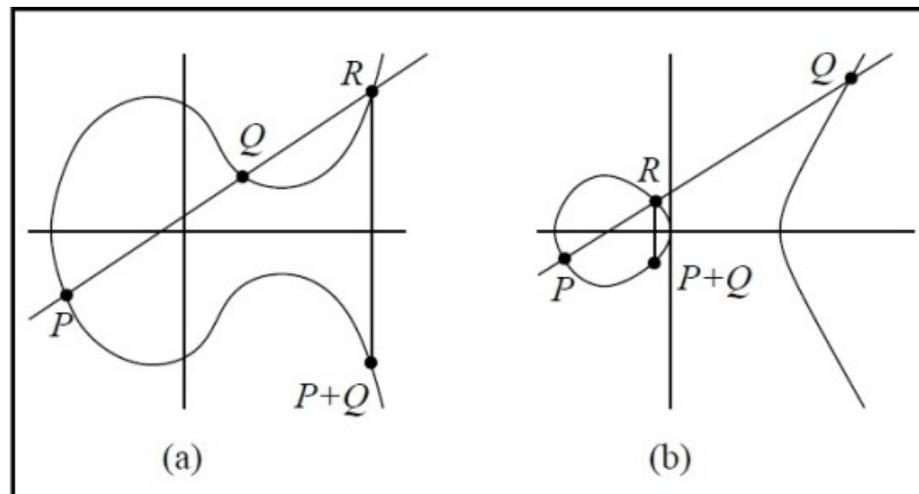
- 고정된 실수인 a, b 가 있을 때, 방정식 $y^2 = x^3 + ax + b$ 를 만족하는 (x, y) 들의 집합을 사용
 - RSA보다 짧은 키 길이를 사용하면서 비슷한 수준의 안전성을 제공
 - 하드웨어와 소프트웨어로 구현하기가 용이 함

공개키 암호 알고리즘

- 기타 알고리즘

- 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)

- $y^2 = x^3 + ax + b$ 를 만족하는 P 보다 작은 소수를 개인키로 생성(d)
- 타원곡선의 더하기연산을 통하여 공개키 생성(f)
 - G : 타원곡선상 임의의 점
 $d \times G = f$



공개키 암호 알고리즘

- 기타 알고리즘

- 타원 곡선 암호 (ECC, Elliptic Curve Cryptography)

- 예제

- x, y 가 유한 할 때

- P 가 23일 때(개인키)

- $y^2 = x^3 + x$ over Z_{23} 에서 x 가 11이면(공개키),
 $y^2 \bmod 23 = (1331 + 11) \bmod 23 = 1342 \bmod 23 = 8$
 $y^2 \bmod 23 = 8$

- 식을 만족하는 y 는 10과 13임

- P 가 커지면 y 가 가지는 값이 많아짐

Thanks!

박재형 (jaehyoung@pel.smuc.ac.kr)