2017/08/06, 2017 보안 기초 세미나

Network Security Essentials

- 3장 공개키 암호와 메시지 인증 (2) -

최창준 (changjun@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

- 메시지 인증 코드
  - HMAC
  - 블록 암호기반 MAC

목 차

메시지 인증 코드

- 메시지 인증 코드(Message Authentication Code)
  - HMAC(Hashed MAC)
    - 해싱 기법을 적용하여 메시지의 위·변조를 방지하는 기법
    - 응용
      - IP 보안, 전송 계층 보안(TLS, Transport Layer Security), 안전한 전자 결제(SET, Secure Electronic Transaction)와 같은 인터넷 프로토콜에서 사용됨

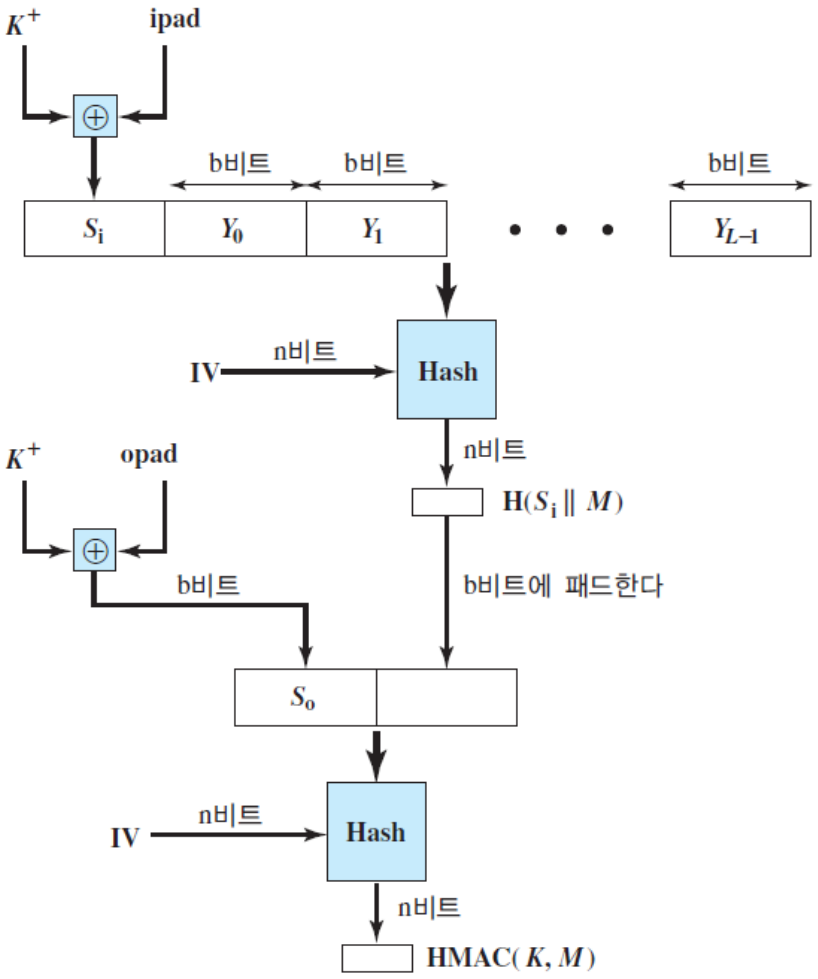
메시지 인증 코드

- HMAC
  - 표기법

표기법	설명
H	해시 함수
M	HMAC의 입력 메시지
Y_i	M의 i번째 블록
L	M의 블록 수
b	블록의 비트 수
n	내장된 해시 함수에 의해 생성된 해시 코드의 길이
K	비밀키, 키의 길이가 b보다 길면 n 비트 키를 생성하는 해시 함수에 입력으로 사용
K^+	K의 왼쪽에 0을 붙여서 길이가 b 비트가 되도록 한 것
ipad	00110110(16 진수 36)을 b/8번 반복한 2진 수열
opad	01011100(16 진수 5C)을 b/8번 반복한 2진 수열

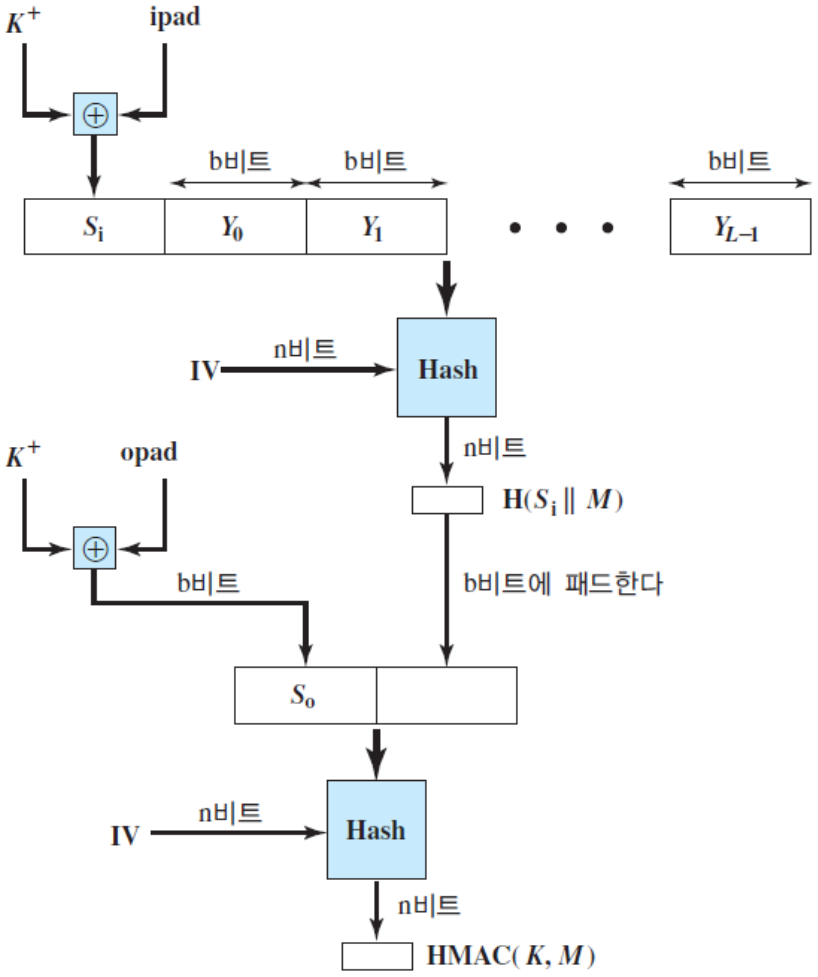
메시지 인증 코드

- HMAC
  - 구조 (1/2)
    1. 메시지를 길이가 b 비트인 블록으로 분리
    2. 비밀 키 K의 왼쪽에 0으로 된 열을 패딩하여 b 비트의 K^+ 생성
    3. K^+와 상수 ipad(input pad)를 XOR 연산하여 b 비트 S_i 블록 생성
    4. S_i를 메시지 맨 앞에 붙이고 n 비트 IV와 해시 함수에 입력
    5. 출력 값으로 생성된 n 비트 다이제스트를 중간 HMAC이라고 부름



메시지 인증 코드

- HMAC
  - 구조 (2/2)
    6. n 비트로 된 중간 HMAC의 왼쪽에 0으로 이루어진 열을 패딩하여 b 비트 블록 생성
    7. 단계 2와 단계 3을 다른 상수인 opad(output pad)로 반복하여 b 비트 S_o 블록 생성
    8. 단계 7에서 얻은 결과를 단계 6에서 얻은 블록 앞에 붙임
    9. 단계 8의 결과에 동일한 해시 함수를 적용하여 최종 n 비트 HMAC을 생성



메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드(CMAC, Cipher-based Message Authentication Code)
    - 대칭키 암호를 N번 사용하여 N개의 평문 블록으로부터 하나의 MAC을 생성
    - 운용모드용으로 AES와 3DES를 사용
      - AES
        - 암호 블록 길이 b : 128 비트
        - 키 길이 k : 128, 192 또는 256 비트
      - 3DES
        - 암호 블록 길이 b : 64 비트
        - 키 길이 k : 112 또는 168 비트
    - 메시지는 n개 블록으로 나뉨

메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 계산식과 용어

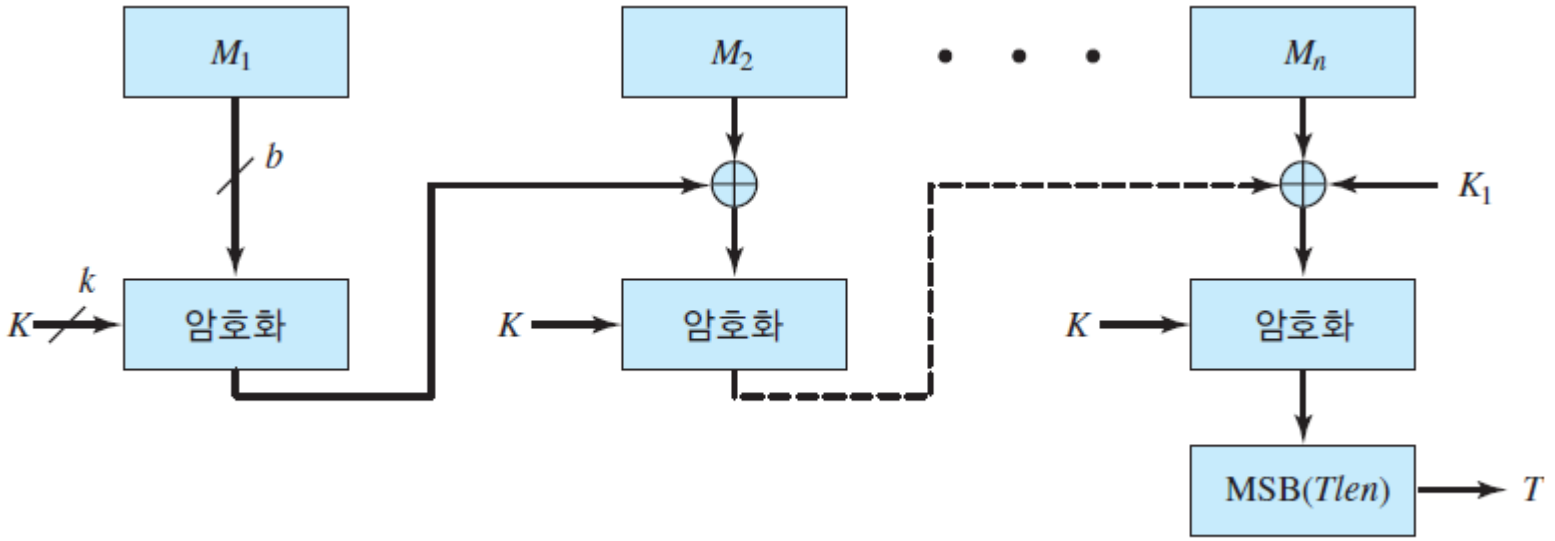
계산식
$C_1 = E(K, M_1)$
$C_2 = E(K, [M_2 \oplus C_1])$
$C_3 = E(K, [M_3 \oplus C_2])$
• • •
$C_n = E(K, [M_n \oplus C_{n-1} \oplus K_1])$
$T = MSB_{Tlen}(C_n)$

용어	정의
T	메시지 인증 코드, “태그(Tag)”
Tlen	T의 비트 길이
MSB_s (X)	비트열 X의 왼쪽부터 S개 비트
K_1	결과로 나온 암호문을 한 비트 왼쪽으로 이동시킨 값
K_2	K_1을 한 비트 왼쪽으로 이동시킨 값

- MSB(Most Significant Bit) : 최상위 비트

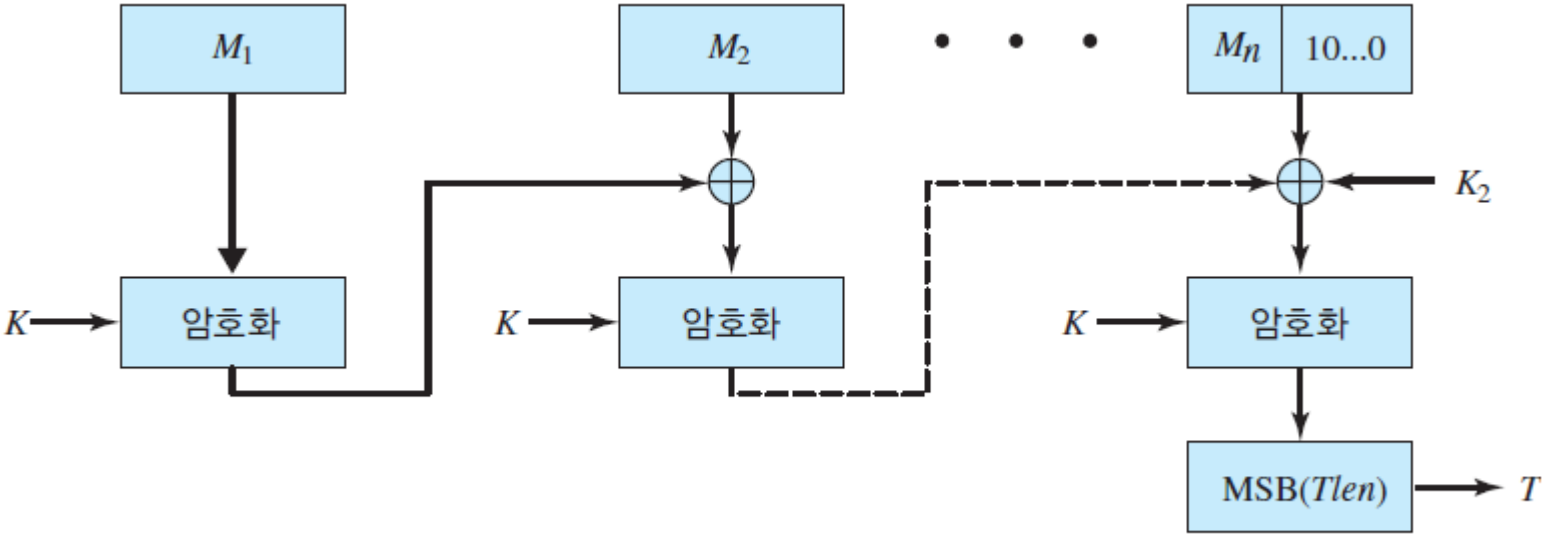
메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 메시지 길이가 블록 길이의 정수배일 때
      - k 비트 키와 b 비트 서브키 K_1을 사용



메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 기반 메시지 인증 코드
    - 메시지 길이가 블록 길이의 정수배가 아닐 때
      - 마지막 블록에 패딩을 붙여 블록의 길이가 b 비트가 되게 함
      - k 비트 키와 b 비트 서브키 K_2를 사용

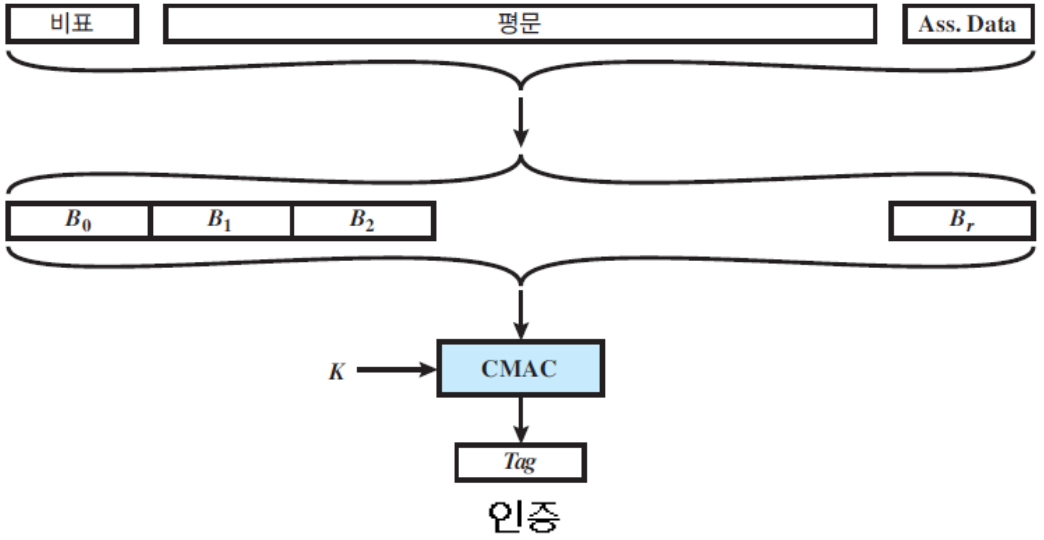


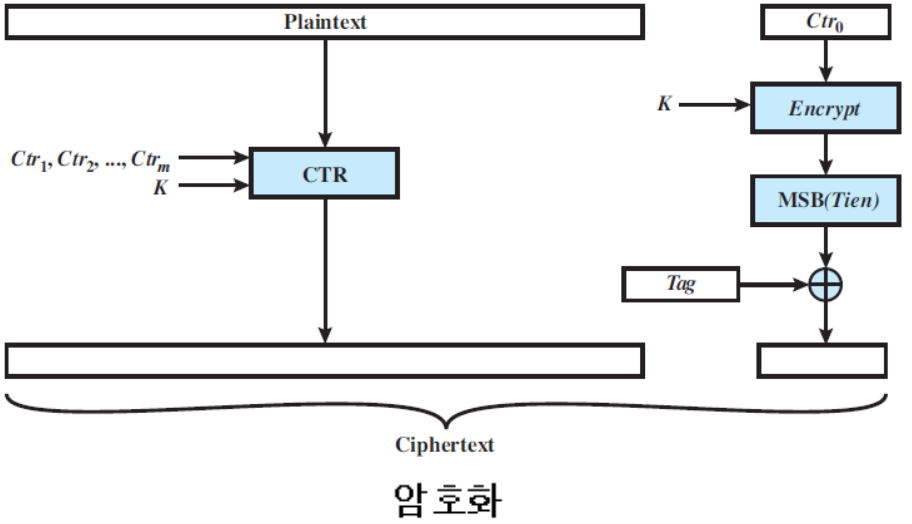
메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 블록 체인 카운터 MAC(CCM, Counter with Cipher block chaining-Message authentication code)
    - 인증된 암호화(Authentication encryption)모드라고도 함
      - 통신상 기밀성과 인증(무결성)을 동시에 보호하는 암호 시스템을 설명할 때 사용되는 용어
    - 핵심 알고리즘 요소
      - AES 암호 알고리즘
      - CTR 운용 모드
      - CMAC 인증 알고리즘

메시지 인증 코드

- 블록 암호 기반 MAC
  - 암호 블록 체인 카운터
    - 암호화와 인증에 동일한 키를 사용
    - 인증 과정에 입력되는 내용
      - 인증하고 암호화할 데이터 (평문 메시지 데이터 블록 P)
      - 인증은 하지만 암호화는 하지 않는 데이터 (유관 데이터 A)
      - 프로토콜 연관이 있는 동안 모든 순간에 달라지는 유일 값 (유관 데이터에 할당되는 비표 N)





공개키 암호

- 공개키 암호(Public-key encryption)
  - 정의
    - 암·복호화할 때 서로 다른 두 개의 키를 사용하는 암호 방식
      - 공개키(Public key)
        - 외부에 공개 되는 키
      - 개인키(Private key)
        - 외부에 공개해서는 안 되는 키
  - 특징
    - 수학적 함수에 의해 만들어짐
    - 기밀성, 키 분배, 인증 분야에서 성능이 뛰어남
    - 공개키로 암호화한 암호문은 공개키와 대응되는 개인키가 아니면 복호화할 수 없음

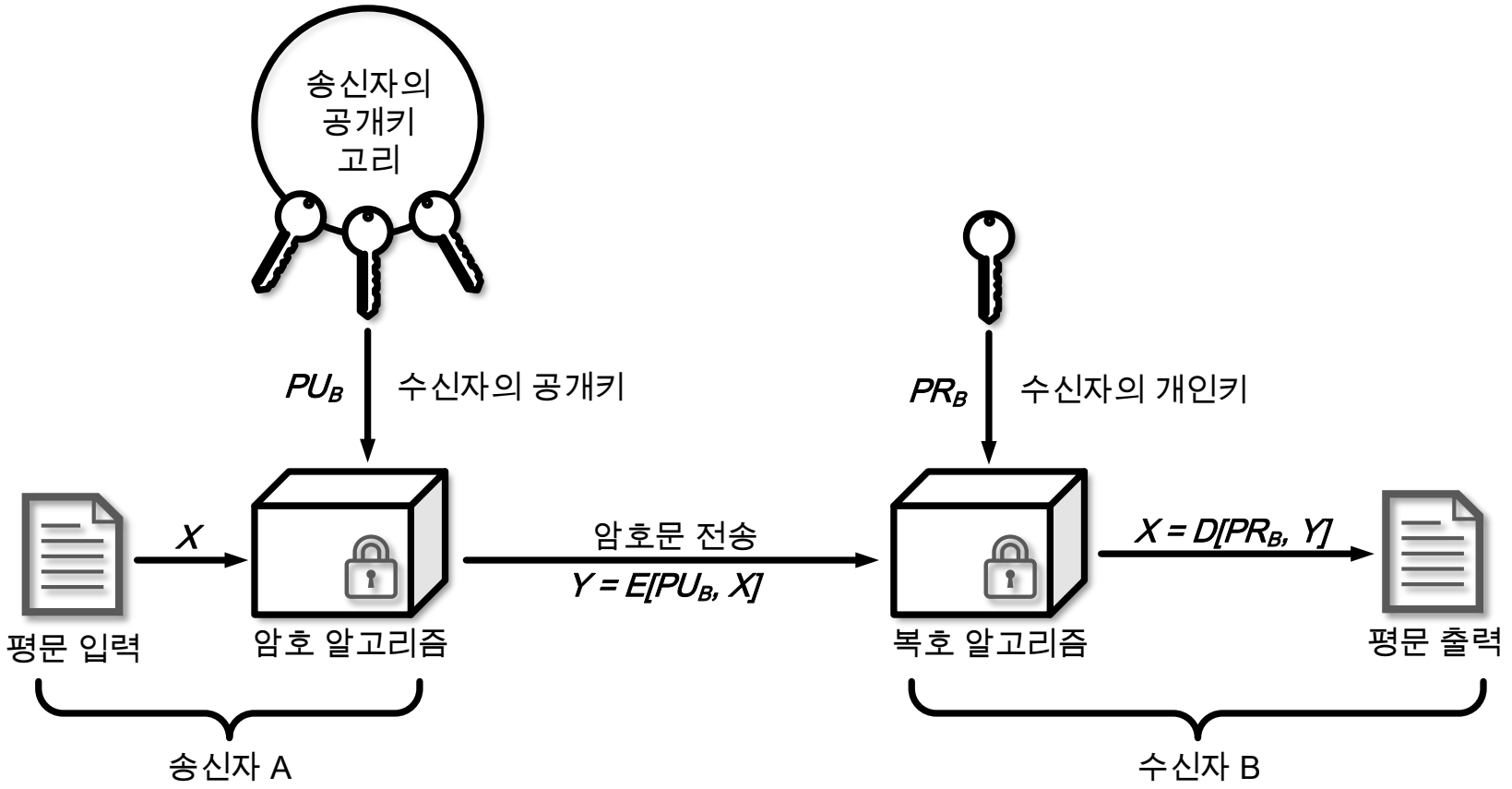
공개키 암호

- 공개키 암호
  - 대칭키와 공개키 암호 방식 비교 표

구분	대칭키 암호 방식	공개키 암호 방식
키	대칭키(비밀키)	비대칭키(공개키, 개인키)
키의 개수	$\frac{N(N-1)}{2}$	$2N$
암호화 키의 관계	암호화 키 = 복호화 키	암호화 키 ≠ 복호화 키
장점	암·복호화 계산 속도가 빠름	암호 키 사전 공유 불필요 키 분배 및 관리가 용이함
단점	키 분배 및 관리가 어려움	암·복호화 계산 속도가 느림
대표 알고리즘	DES	RSA

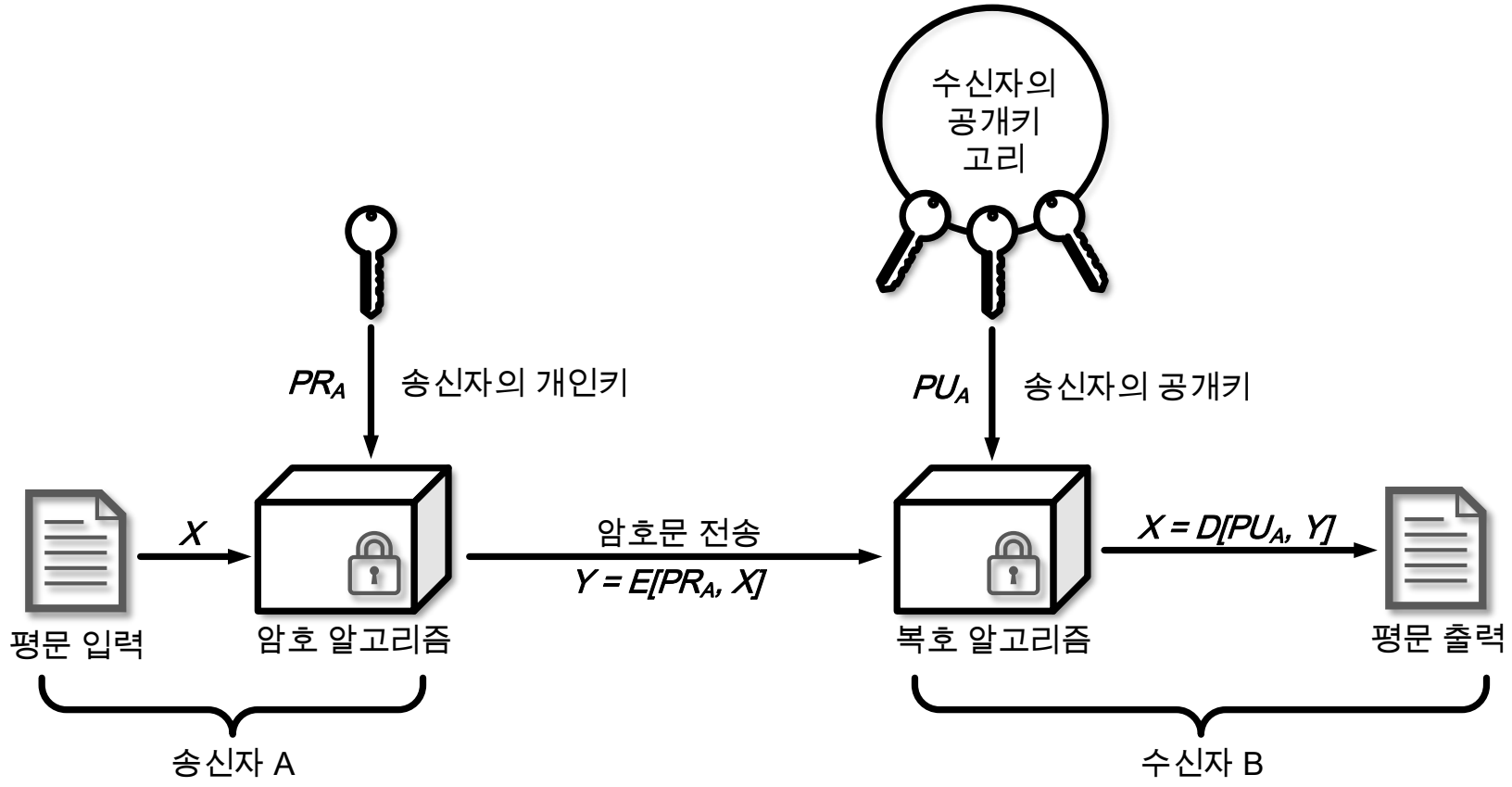
공개키 암호

- 공개키 암호
  - 구조
    - 공개키에 의한 암호화



공개키 암호

- 공개키 암호
  - 구조
    - 개인키에 의한 암호화



공개키 암호

- 공개키 암호
  - 응용
    - 암호화/복호화(Encryption/Decryption)
      - 송신자는 수신자의 공개키로 메시지 암호화
    - 디지털 서명(Digital signature)
      - 송신자는 자신의 개인키로 메시지에 서명
    - 키 교환(Key exchange)
      - 키 합의(Key Agreement) 방식
        - 통신 양측의 공개키와 개인키를 사용하여 비밀키 생성
        - e.g., Diffie-Hellman 알고리즘
      - 키 전송(Key Transport) 방식
        - 통신 양측 중 한쪽이 비밀키를 생성하여 상대방의 공개키로 암호화한 후 전송
        - e.g., RSA 알고리즘

공개키 암호

- 공개키 암호
  - 응용
    - 공개키 알고리즘 비교 표

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

- 공개키 암호
  - 요건
    - 한 쌍의 키(공개키, 개인키)를 생성하는 것이 쉬워야 함
    - 송신자는 암호문을 쉽게 구할 수 있어야 함
      - $C = E(PU, M)$
    - 수신자는 암호문을 복호화하는 것이 쉬워야 함
      - $M = D(PR, C) = D[PR, E(PU, M)]$
    - 키 쌍 중에 하나의 키를 알고 있는 경우 다른 키를 예측할 수 없어야 함
    - 공개키와 암호문을 알고 있더라도 해독이 불가능해야 함
    - 암·복호화에 키가 하나씩 사용되어야 함
      - $M = D[PU, E(PR, M)] = D[PR, E(PU, M)]$
    - 평문은 키 값보다 작아야 함
      - $C = M^e \ mod \ n$

공개키 암호 알고리즘

- RSA(Rivest Shamir Adleman) 알고리즘
  - 정의
    - 1977년 MIT의 Ron Rivest와 Adi Shamir, Len Adlman이 개발한 공개키 암호 알고리즘
  - 특징
    - 소인수분해 문제의 어려움을 기반으로 한 알고리즘
    - 해독하는 데 시간이 오래 걸리기 때문에 널리 사용됨
      - e.g., 공인 인증서 등
    - 평문 메시지와 암호문, 키 값을 모두 숫자로 취급함

공개키 암호 알고리즘

- RSA 알고리즘
  - 표기법

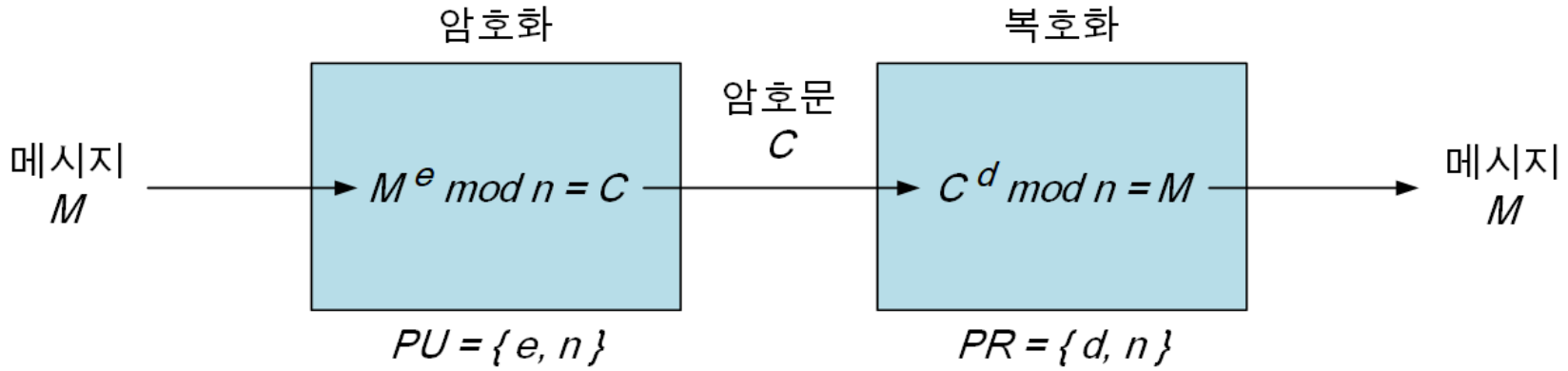
표기법	설명
M	메시지
C	암호화된 메시지 (암호문)
p, q	키를 생성하기 위해 선택하는 소수 값
$n = (p \times q)$	암·복호화에 이용되는 키의 인자 값, $modulus$로 사용
e	공개키의 인자 값 (공개 값)
d	개인키의 인자 값 (비밀 값)
$PU = \{e, n\}$	공개키
$PR = \{d, n\}$	개인키

공개키 암호 알고리즘

- RSA 알고리즘
  - 키 생성 과정
    1. 두 개의 큰 소수 p와 q를 선택 $(p \neq q)$
    2. $n = p \times q$를 계산
    3. $\varphi(n) = (p-1)(q-1)$를 계산
       - 오일러(Euler) 함수 $\varphi(n)$
         - n보다 작으면서 n과 서로소인 정수의 개수
    4. $1 < e < \varphi(n)$이면서 $\varphi(n)$과 서로소인 정수 e를 선택
    5. $de \bmod \varphi(n) = 1$를 만족하는 d를 계산
    6. 공개키 $PU = \{e, n\}$ 생성
    7. 개인키 $PR = \{d, n\}$ 생성

공개키 암호 알고리즘

- RSA 알고리즘
  - 암·복호화 과정



  - 암호화
    - $C = M^e \ mod \ n$
    - 메시지 M을 공개키 $PU = \{e, n\}$로 암호화
  - 복호화
    - $M = C^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$
    - 암호문 C를 개인키 $PR = \{d, n\}$로 복호화

공개키 암호 알고리즘

- RSA 알고리즘
  - 보안
    - 전수조사 공격
      - 가능한 모든 경우의 개인키를 시도해보는 공격
      - 대응책
        - e와 d의 비트 크기를 최소 512비트가 되도록 함
          - 10진수로 약 154자리 수
    - 소인수분해 공격
      - n을 소인수분해 한 p와 q를 구하여 키 값을 얻는 공격
      - 대응책
        - n의 비트 크기를 최소 1024비트가 되도록 함
          - 10진수로 약 300자리 수

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 정의
    - 1976년 Whitfield Diffie와 Martin Hellman에 의해 개발된 키 교환 알고리즘
  - 특징
    - 이산 대수 문제의 어려움을 기반으로 한 알고리즘
    - 공개키를 교환하여 통신 양측이 사용할 비밀키 생성

공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 이산대수 문제(Discrete logarithms problem)
    - 원시근(Primitive root) α
      - 소수 p의 원시근
        - 자신의 거듭제곱을 이용하여 1부터 $p-1$까지의 정수를 생성해 낼 수 있는 수
        - $\alpha \bmod p, \alpha^2 \bmod p, \cdots, \alpha^{p-1} \bmod p$
    - 이산대수(Discrete logarithm)
      - p보다 작은 임의의 정수 b와 p의 원시근 α에 대해서

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

        지수(Exponent) i를 밑수 α를 갖는 b의 이산대수라고 함
        - 표기 : $dloga_{\alpha,p}(b)$

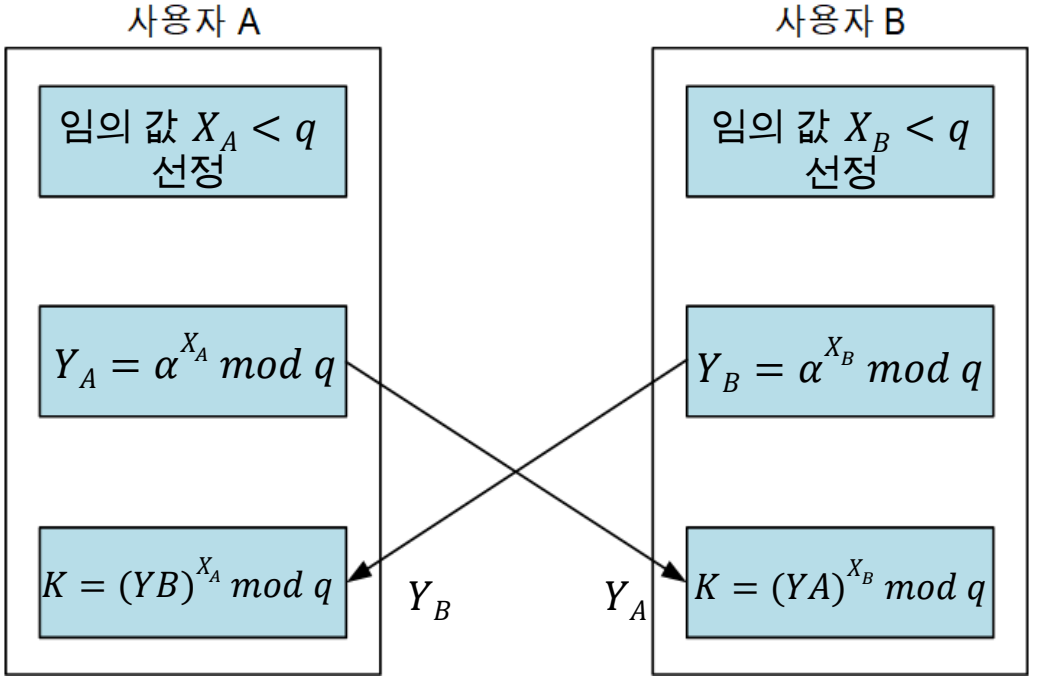
공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 이산대수 문제(Discrete logarithms problem)
    - 예제
      - $7^X \ mod \ 13 \ = \ 8$ 이 되는 X 값은 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
  - 비밀키 생성 과정
    - 공개되는 값
      - 소수 q, 원시근 α, 공개 값 Y_A, Y_B

1. 통신 양측은 임의의 개인 값 X_A, X_B 선택
2. 공개 값 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
   $K = (Y_B)^{X_A} \bmod q$
   $= (Y_A)^{X_B} \bmod q$



공개키 암호 알고리즘

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

$$= (\alpha^{X_B} \ mod \ q)^{X_A} \ mod \ q$$

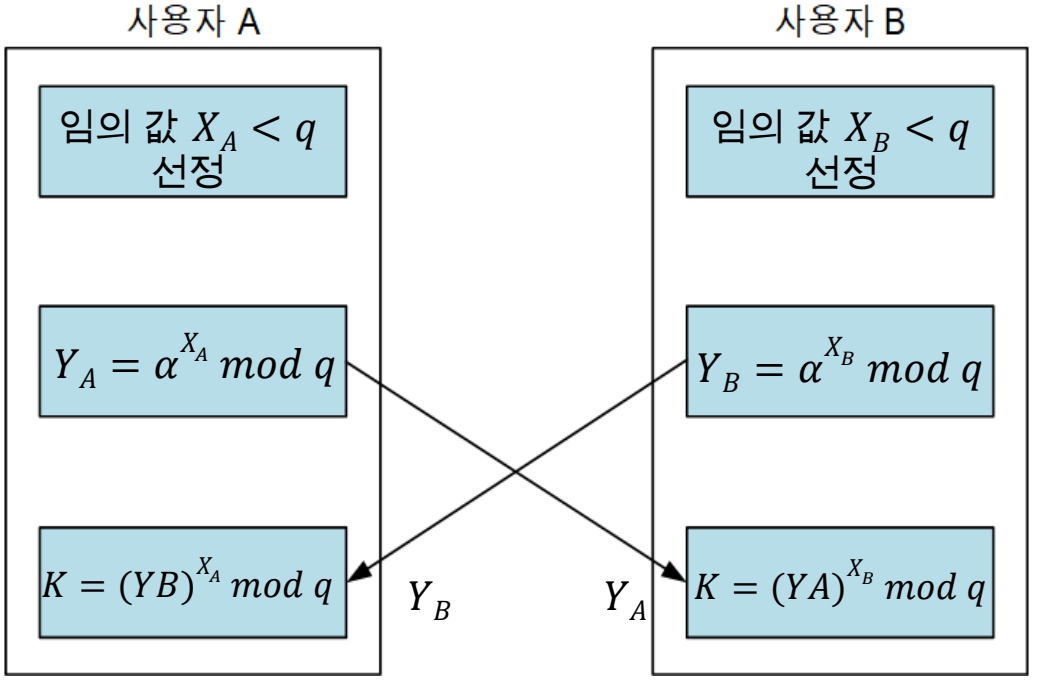
$$= (\alpha^{X_B})^{X_A} \ mod \ q$$

$$= \alpha^{X_B X_A} \ mod \ q$$

$$= (\alpha^{X_A})^{X_B} \ mod \ q$$

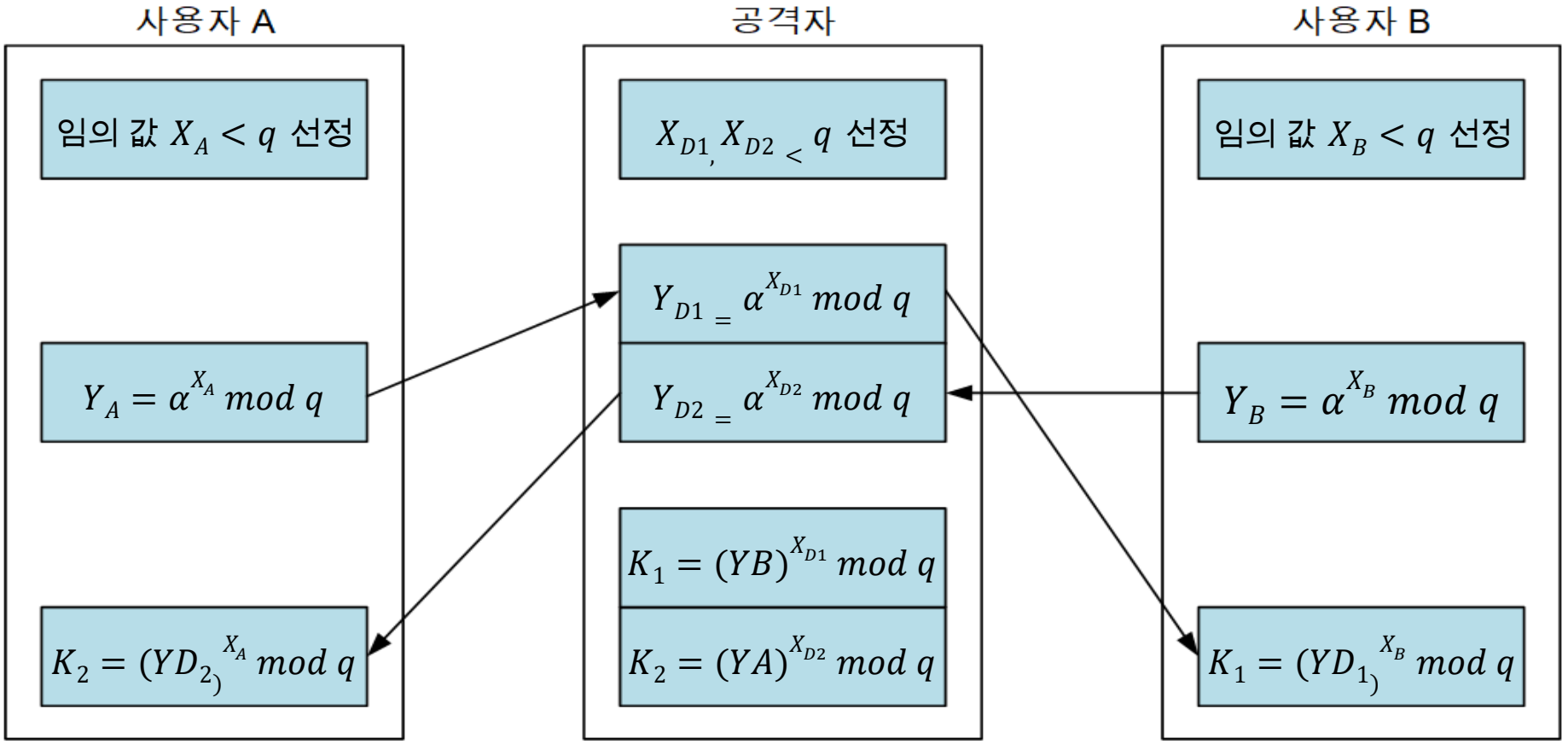
$$= (\alpha^{X_A} \ mod \ q)^{X_B} \ mod \ q$$

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



공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - 중간자 공격(Man-in-the-middle attack)



공개키 암호 알고리즘

- Diffie-Hellman 알고리즘
  - RSA와 Diffie-Hellman 알고리즘 비교 표

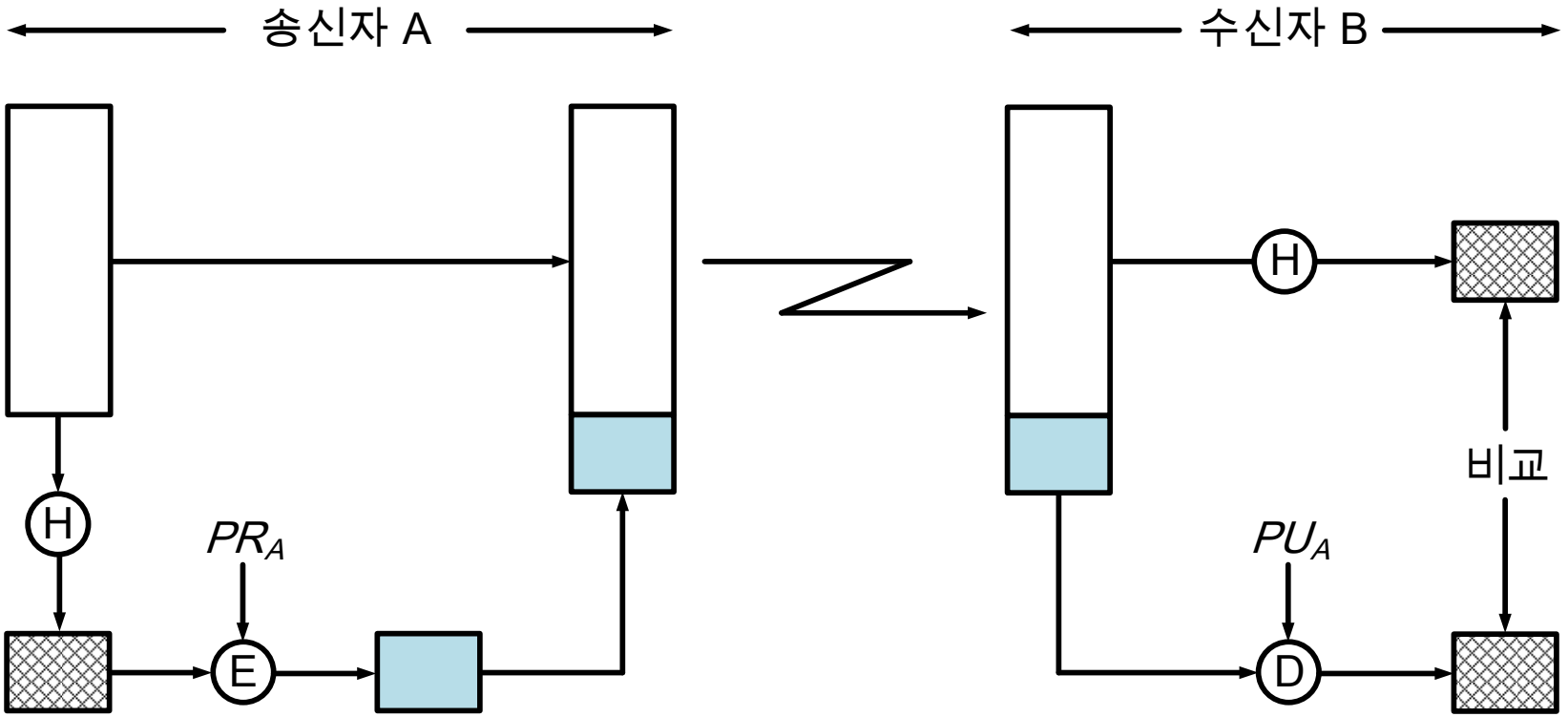
구분	RSA	Diffie-Hellman
수학적 배경	소인수분해 문제	이산대수 문제
키 분배 방법	키 전송	키 합의
응용 분야	암·복호화, 디지털 서명, 키 교환	키 교환
장점	여러 라이브러리 존재	키 분배에 최적화 키는 필요 시에만 생성, 저장 불필요
단점	느린 계산 속도	위조에 취약

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(Authenticator) 블록 만듦
        - 인증자
          - 메시지의 기능을 대신하는 작은 블록
        - 인증자를 개인키로 암호화하여 메시지의 출처, 무결성, 순서를 확인해주는 서명 생성
      - 공개키로 복호화하기 때문에 기밀성을 보장하지 않음

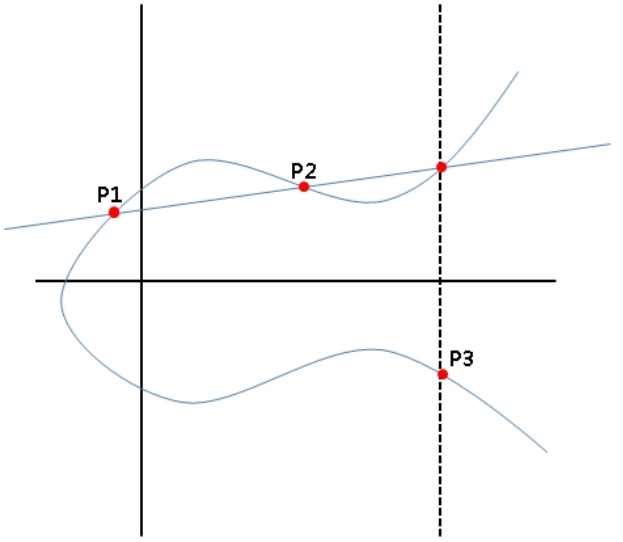
공개키 암호 알고리즘

- 기타 알고리즘
  - 디지털 서명
    - 구조



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
      - 방정식 $y^2 = x^3 + ax + b$ 를 만족하는 점들의 집합
      - RSA보다 짧은 키 길이를 사용하면서 비슷한 수준의 안전성을 제공



감사합니다!