2021/07/20, 2021 보안 기초 세미나

Network Security Essentials

- Chapter_3 공개키 암호와 메시지 인증(2) -

발표자 : 이 태 양(taeyang@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

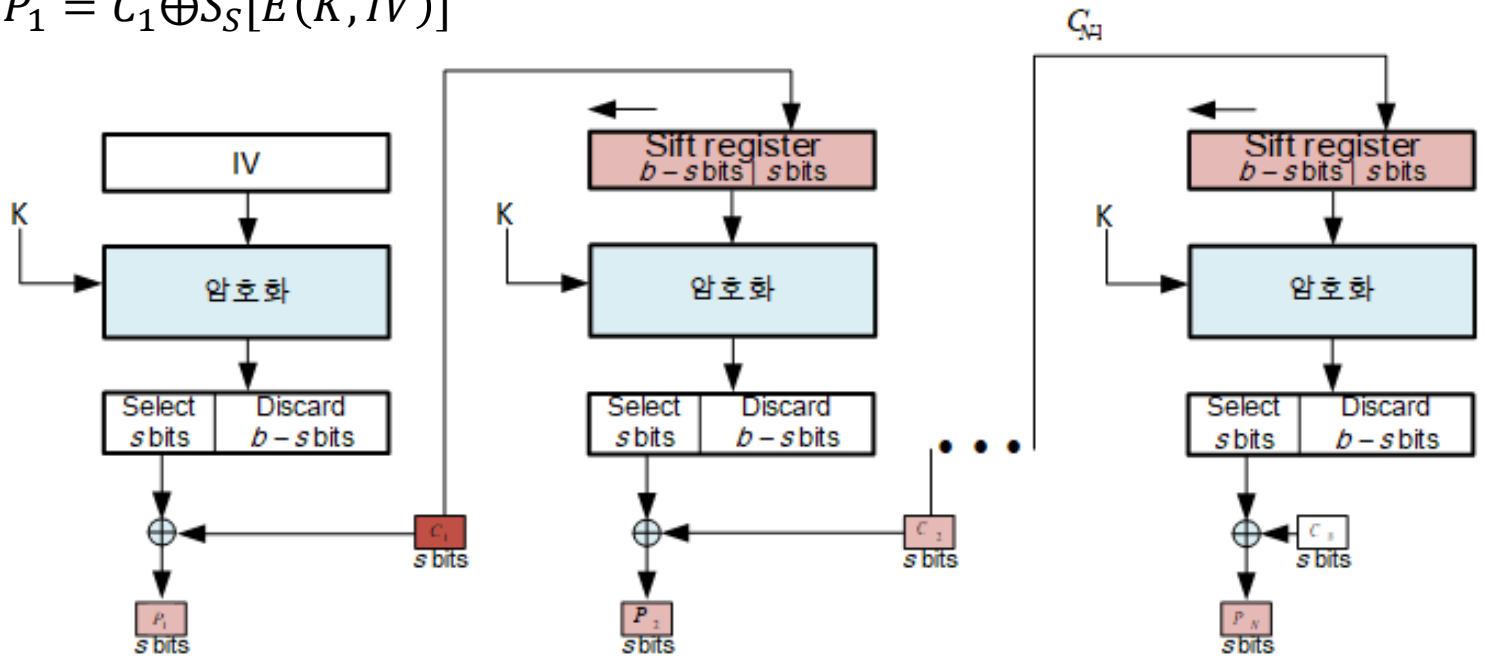
목 차

- 암호 블록 운용 모드
- 공개키 암호 원리
- 공개키 암호 알고리즘

암호 블록 운용 모드

- 종류
  - 암호 피드백 (CFB, Cipher FeedBack) 모드
    - 오류 확산
      - 복호화 과정
        - 암호문 블록 하나에 오류가 있을 때, 현재 평문 블록 이후 Shift register에서 오류가 소멸 될 때까지 평문 블록에 영향

$$P_1 = C_1 \oplus S_S[E(K, IV)]$$

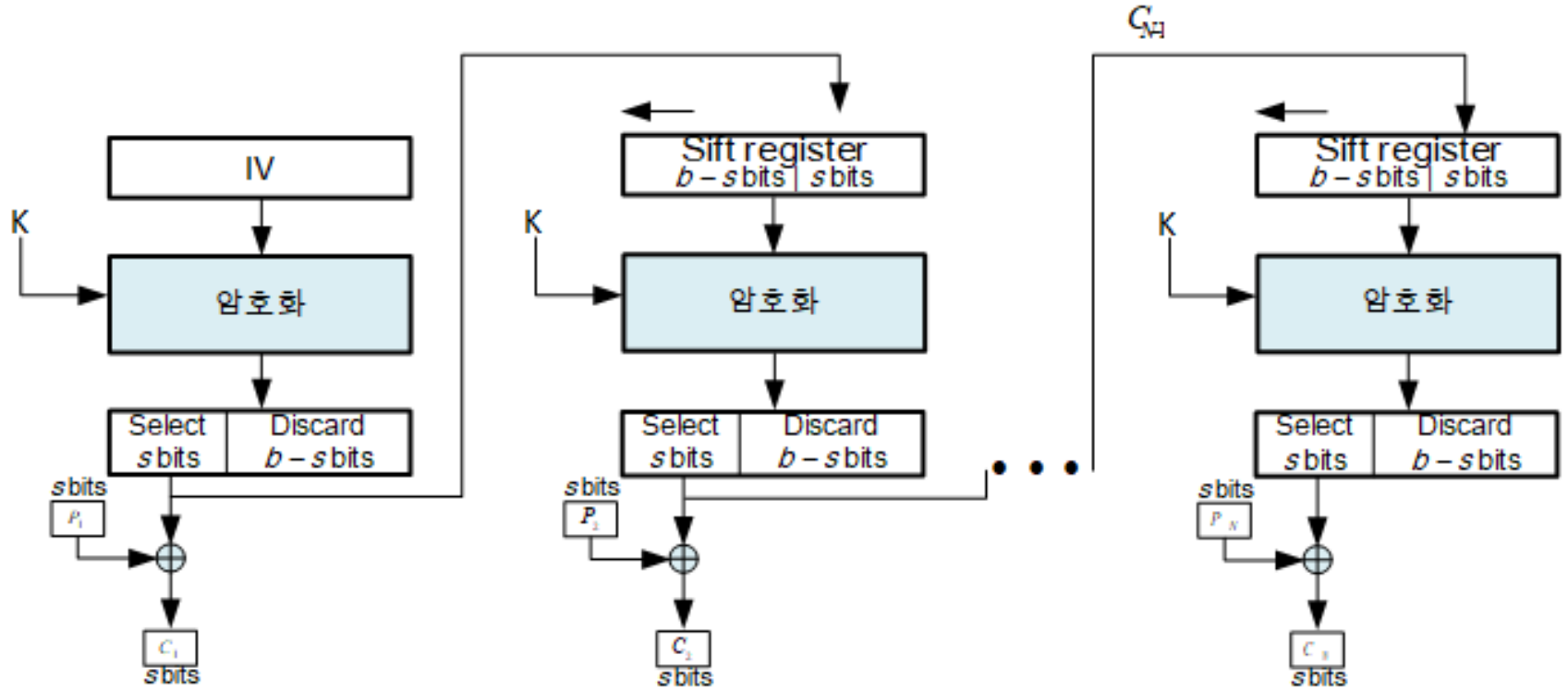


암호 블록 운용 모드

- 종류
  - 암호 피드백 (CFB, Cipher FeedBack) 모드
    - 장점
      - 패딩이 불필요
      - 실시간 사용 가능
    - 단점
      - 속도가 느림
      - 암호화 과정에서 병렬처리 불가능
      - 오류가 확산됨

암호 블록 운용 모드

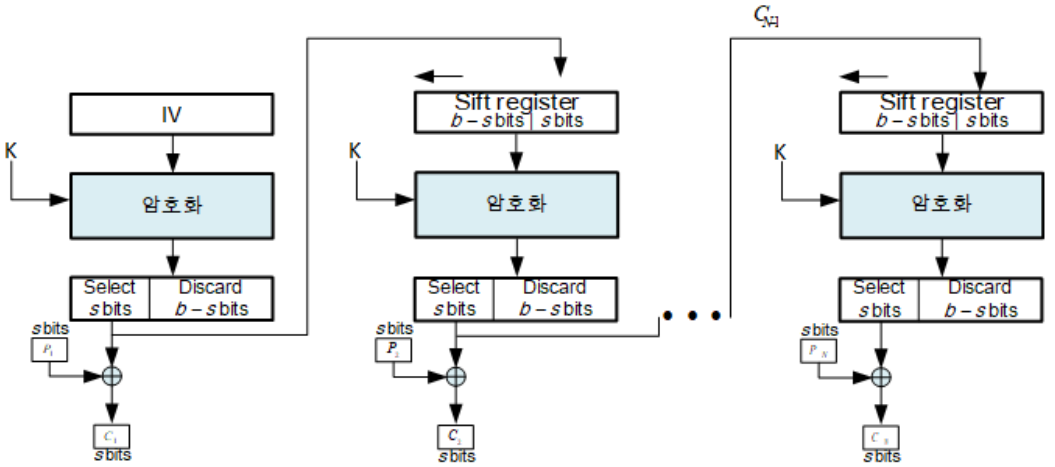
- 종류
  - 출력 피드백 (OFB, Output FeedBack) 모드
    - 평문 블록에 직접 암호화하지 않고 이전 암호 알고리즘의 출력을 입력으로 피드백하여 평문과 XOR하는 방식



암호 블록 운용 모드

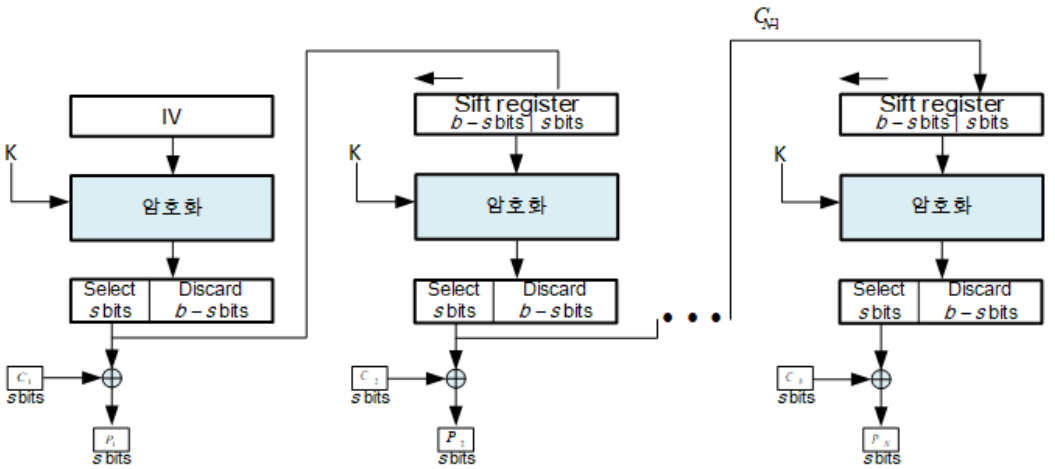
- 종류
  - 출력 피드백 (OFB, Output FeedBack) 모드
    - 암호화

$$P_i \oplus S_i = C_i$$
$$S_i = E_k(S_{i-1})$$



    - 복호화

$$C_i \oplus S_i = P_i$$
$$S_i = E_k(S_{i-1})$$

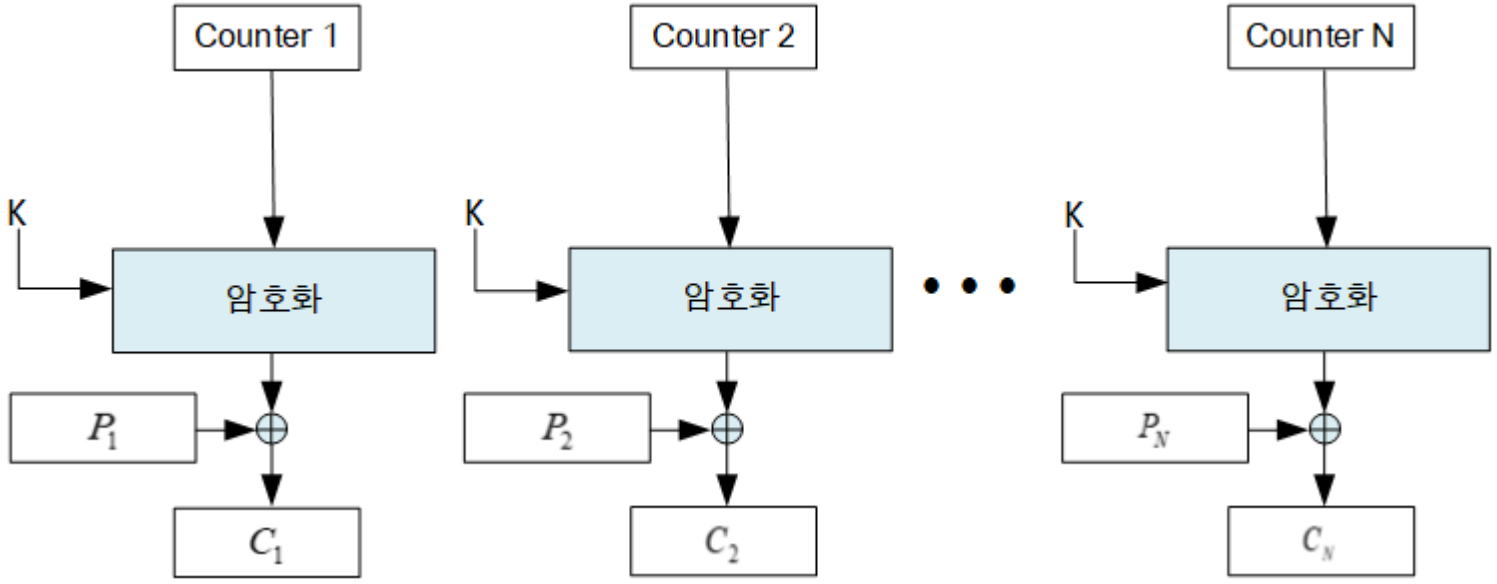


암호 블록 운용 모드

- 종류
  - 출력 피드백 (OFB, Output FeedBack) 모드
    - 장점
      - 패딩이 불필요
      - 오류가 확산되지 않음
        - 암호알고리즘의 출력과 평문을 XOR하여 암호문 생성
    - 단점
      - 병렬처리 불가능

암호 블록 운용 모드

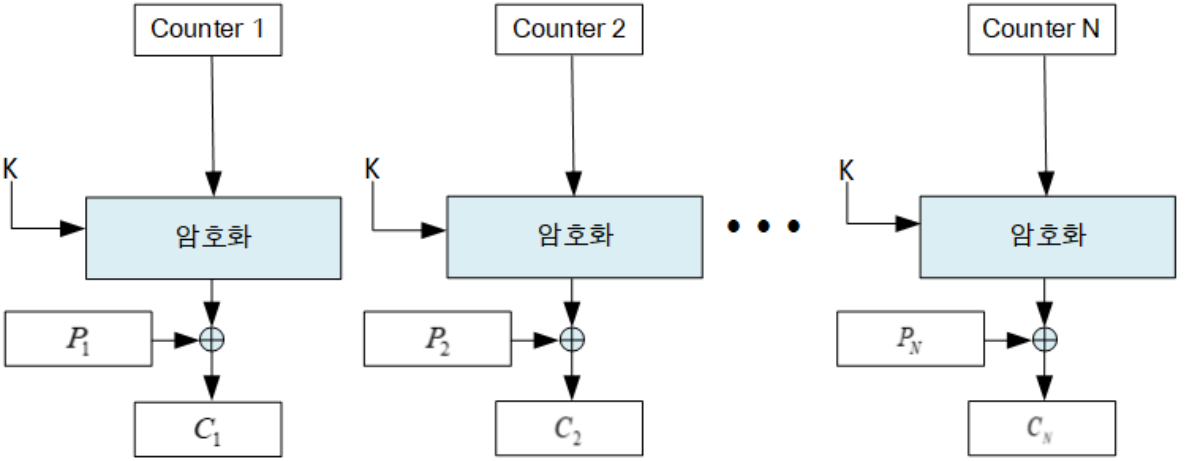
- 종류
  - 카운터 (CTR, Counter) 모드
    - 1씩 증가하는 카운터를 암호화해서 키스트림을 만들어내는 방식



암호 블록 운용 모드

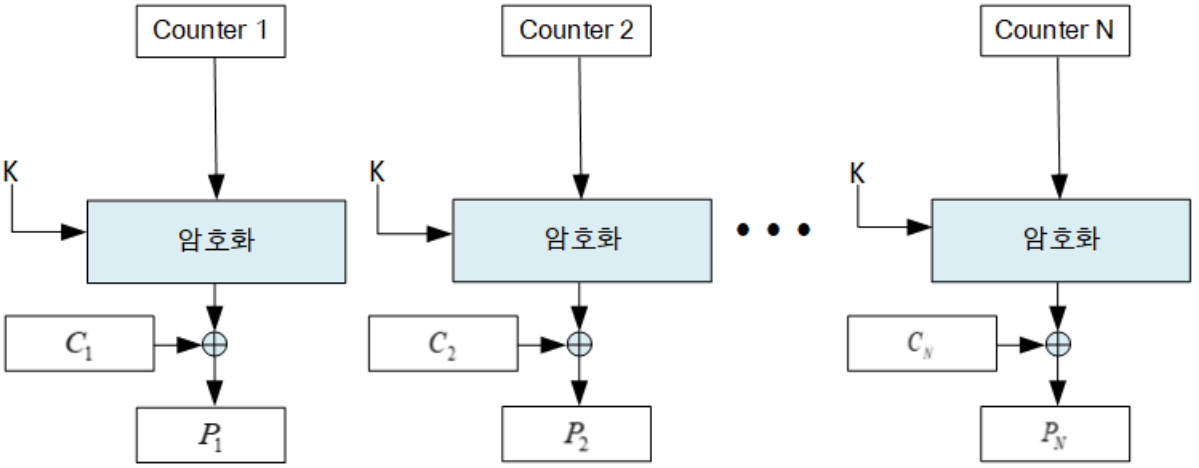
- 종류
  - 카운터 (CTR, Counter) 모드
    - 암호화

$$P_i \oplus S_i = C_i$$
$$S_i = E_k(N_i)$$



    - 복호화

$$C_i \oplus S_i = P_i$$
$$S_i = E_k(N_i)$$



암호 블록 운용 모드

- 종류
  - 카운터 (CTR, Counter) 모드
    - 카운터 구조

암호 블록 운용 모드

- 종류
  - 카운터 (CTR, Counter) 모드
    - 장점
      - 병렬처리 가능
      - 오류가 확산되지 않음
      - 사전에 카운터를 미리 암호화할 수 있음
      - 원하는 부분만 복호화할 수 있음
      - 처리속도가 빠름
    - 단점
      - 공격자가 암호문 블록의 비트를 반전시키면 대응하는 평문 블록 비트가 반전됨

암호 블록 운용 모드

- 암호 블록 운용 모드 비교

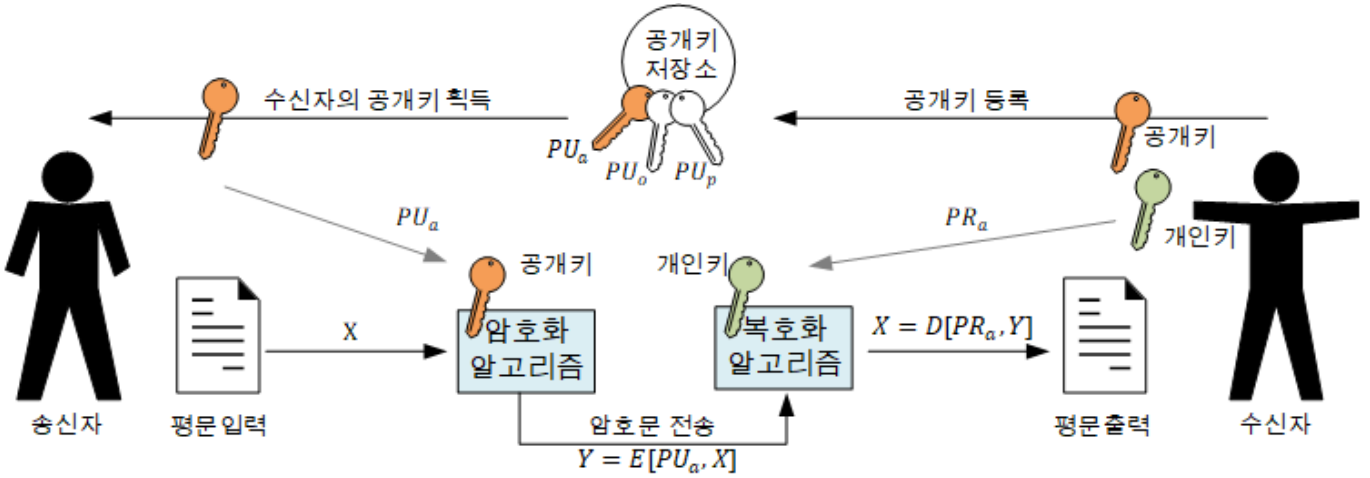
운용 모드	병렬처리	패딩	초기화 벡터	오류확산
ECB	○	○	X	X
CBC	복호화만	○	○	E : 해당 블록 이후 모든 블록 D : 해당 블록 다음 블록
CFB	복호화만	X	○	Shift register에서 오류가 완전히 소멸될 때까지
OFB	X	X	○	X
CTR	○	X	X	X

공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 정의
    - 암호화와 복호화에 서로 다른 키(개인키, 공개키)를 사용하는 암호화 방식
      - 개인키 (Private Key)는 소유자만 가지고 있는 키
      - 공개키 (Public Key)는 누구나 알 수 있도록 공개하는 키
  - 암호 방식
    - 공개키 암호
      - 공개키로 암호화 후 개인키로 복호화
      - 특정한 비밀키를 가지고 있는 사용자만 내용을 열어볼 수 있음
    - 공개키 서명
      - 개인키로 암호화 후 공개키로 복호화
      - 특정한 비밀키로 만들어졌다는 것을 누구나 확인할 수 있음

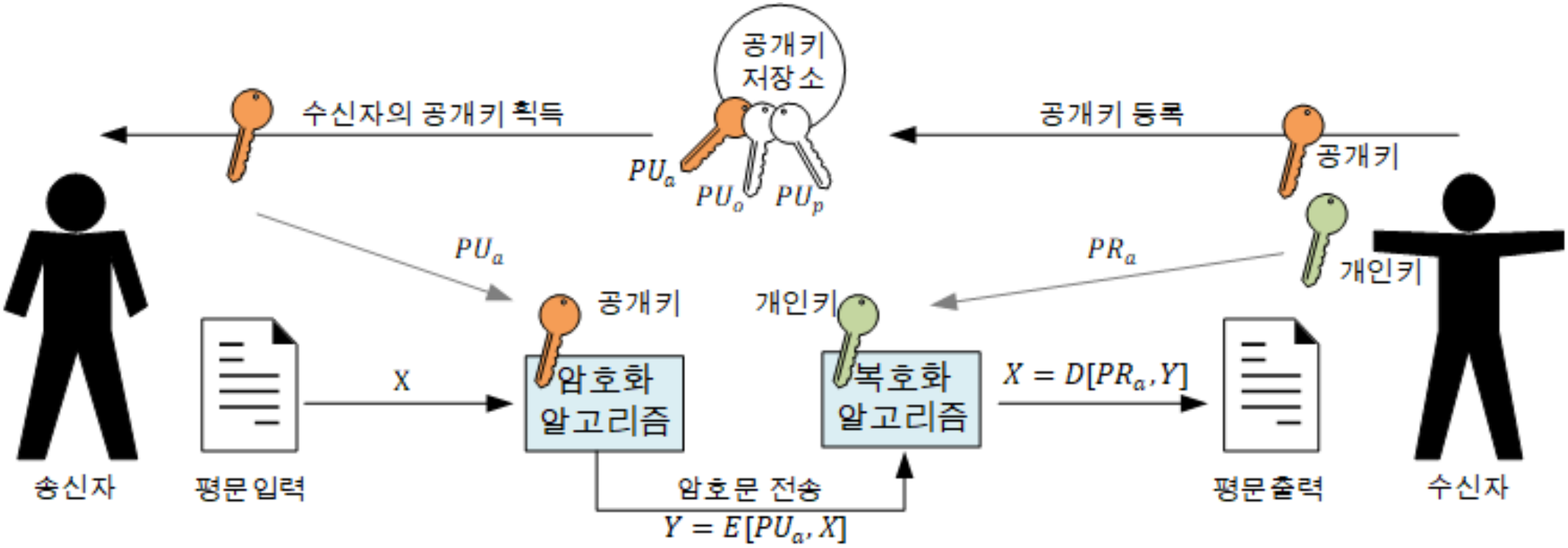
공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 구성요소
    - 평문 (Plaintext)
    - 암호화 알고리즘 (Encryption Algorithm)
    - 공개키와 개인키 (Public and Private Key)
    - 암호문 (Ciphertext)
    - 복호화 알고리즘 (Decryption Algorithm)



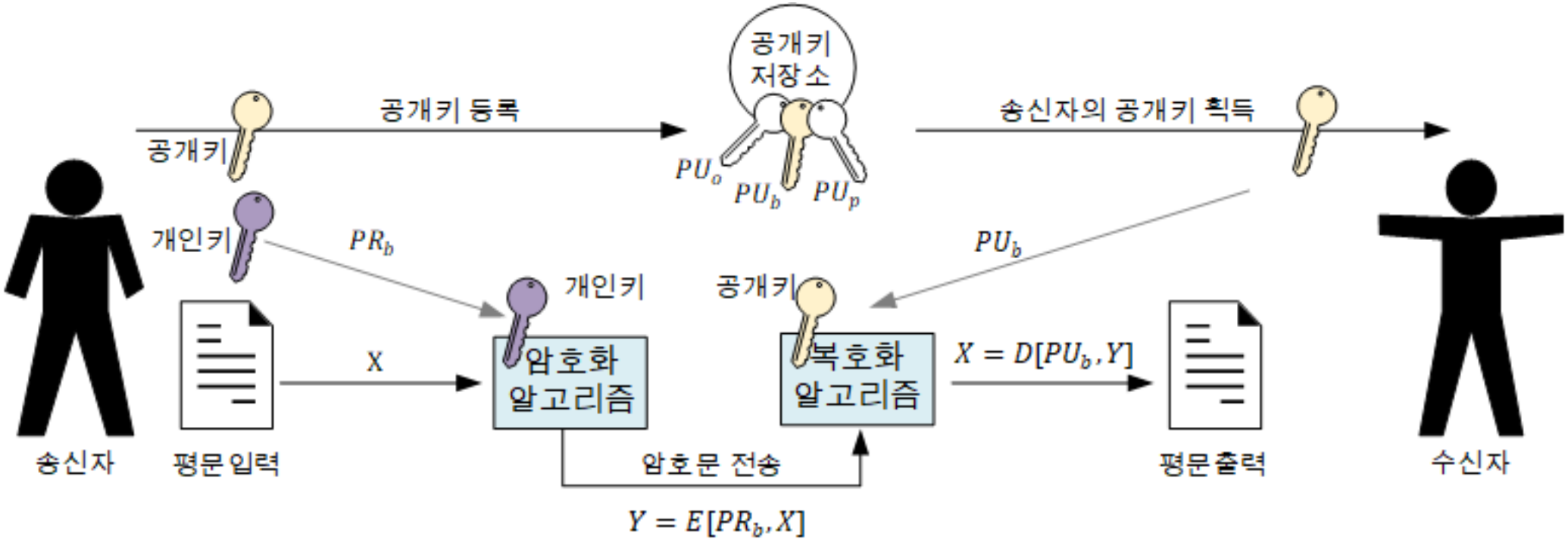
공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 구조 (1/2)
    - 공개키에 의한 암호화



공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 구조 (2/2)
    - 개인키에 의한 암호화



공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 특징
    - 암호화와 복호화에 서로 다른 두 개의 키를 사용하는 비대칭 방식
    - 사전 키 교환이 불필요
    - 연산이 복잡하여 대칭키 암호보다 속도가 약 1000배 느림

공개키 암호 원리

- 대칭키 암호 방식과 공개키 암호 방식 비교

	대칭키 암호화 방식	공개키 암호화 방식
개념	- 암호화와 복호화에 동일한 키 사용 - 대칭 구조	- 암호와와 복호화에 서로 다른 키 사용 - 비대칭 구조
장점	- 공개키 암호화에 비해 속도가 빠름	- 키 분배 및 관리가 용이
단점	- 키 관리가 어려움	- 대칭키 암호화에 비해 속도가 느림 - 개인키를 잃어버리는 경우 암호화된 데이터에 접근 할 수 없음 - 개인키가 공개되는 경우 보안에 위협을 받게 됨

공개키 암호 원리

- 공개키 암호 (Public Key Encryption)
  - 요건 (1/2)
    - 수신자가 한 쌍의 키 (공개키 : PU_b, 개인키 : PR_b) 생성 시 컴퓨터 계산 시간을 고려해야 함
    - 공개키와 암호화될 메시지를 알고 있는 송신자는 암호문을 계산적으로 쉽게 구할 수 있어야 함
      - $C = E(PU_b, M)$
    - 수신자가 암호문을 복호화하는 것이 계산적으로 쉬워야 함
      - $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$

공개키 암호 원리

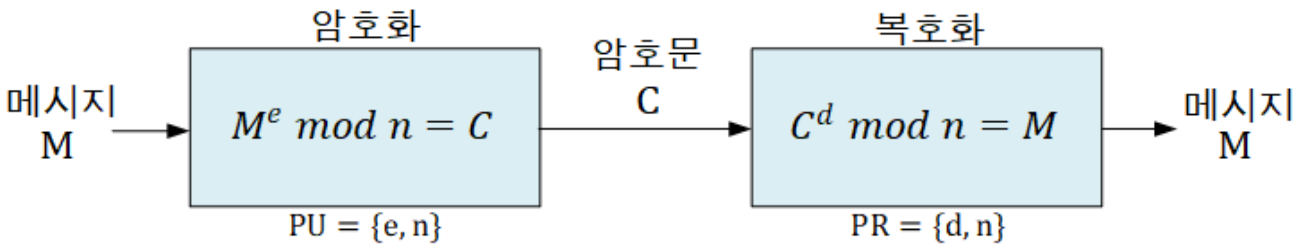
- 공개키 암호 (Public Key Encryption)
  - 요건 (2/2)
    - 공개키를 알고 있는 공격자가 개인키를 알아내는 것이 불가능해야 함
    - 공격자가 공개키와 암호문을 알고 있더라도 원문을 알아내는 것이 불가능해야 함
    - 2개의 키 중 하나를 암호화에 사용하면 다른 하는 복호화에 사용할 수 있어야 함
      - $\mathrm{M} = D[PU_b, E(PR_b, M)] = D[PR_b, E(PU_b, M)]$

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - 정의
    - 자리 수가 많은 양의 정수에 대한 소인수분해가 어렵다는 점에 착안하여 수학적으로 구현한 암호화 알고리즘
  - 특징
    - 암호화뿐만 아니라 전자서명의 용도로도 사용
    - SSL 프로토콜을 가진 웹브라우저, PGP, 공개키 암호 시스템을 사용하는 정부 시스템 등에서 사용
    - 암호화는 공개키를 복호화는 개인키를 사용

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - 암호화와 복호화의 형태
    - $C = M^e \ mod \ n$
    - $M = C^d \ mod \ n = (M^e)^d \ mod \ n = \ M^{ed} \ mod \ n$



  - 이산대수 (Discrete Logarithm) 연산
    - 지수에 대한 모듈로 연산의 역 연산
    - $y = a^x \ mod \ n$에서 y, a, n 이 주어질 때 x의 값을 구하는 연산
    - $y = a^x \ mod \ n$에서 n이 충분히 큰 수로 주어질 때 계산 불가
      - e.g., n이 1024비트 일 때

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - 동작
    - 키 $(PU = \{e, n\},\ PR = \{d, n\})$ 생성 과정

1. 서로 다른 임의의 두개의 소수 p와 q를 선택

2. n 계산
   - p와 q를 곱하여 $n = pq$ 를 계산

3. $\emptyset(n)$ 계산
   - $\emptyset(n)$: 오일러함수로서 양의 정수 중 n과 서로소인 양의 정수의 개수
   - 양의 정수 n이 두 개의 소수 p, q의 곱일 때 다음이 성립
   - $\emptyset(n) = (p-1)(q-1)$

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - 동작
    - 키 $(PU = \{e, n\},\ PR = \{d, n\})$ 생성 과정

4. 정수 e 선택
   - e 와 $\emptyset(n)$ 는 서로소 $(1 < e < \emptyset(n))$
   - $\gcd(a, b)$: a 와 b의 최대공약수
   - $\gcd(\emptyset(n), e) = 1$
   - 공개키 $PU = \{e, n\}$ 생성

5. d 계산
   - $1 < d < \emptyset(n)$
   - $de \bmod \emptyset(n) = 1$
   - 개인키 $PR = \{d, n\}$ 생성

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - 동작
    - 암호화
      - 공개키 $PU = \{e, n\}$를 사용
      - $C = M^e \ (mod\ n)$
    - 복호화
      - 개인키 $PR = \{d, n\}$를 사용
      - $M = C^d \ (mod\ n)$

공개키 암호 알고리즘

- RSA (Rivest Shamir Adleman) 암호 알고리즘
  - RSA 암호 알고리즘에 대한 공격
    - 전수공격
      - 가능한 모든 경우의 개인키로 시도해보는 공격
    - 소인수분해 공격
      - n을 소인수분해하여 p와 q값으로 키를 구하는 공격
  - 공격에 대한 대응
    - 크기가 큰 키를 사용
      - 2048-비트 키 사용을 권장

공개키 암호 알고리즘

- Diffie-Hellman 암호 알고리즘
  - 정의
    - 상호간 대칭키 공유를 위한 알고리즘
  - 특징
    - 상대방의 공개키와 자신의 개인키를 이용하여 비밀키 생성
    - 이산 대수 문제 계산의 어려움을 착안하여 만들어짐
      - a, x, n을 이용하여 $y = a^x \ mod \ n$을 구하기는 쉽지만, a, y, n 값으로 원래의 x를 구하기는 어렵다는 원리를 이용
    - 이산대수법에 의한 수학적 공식에 의해 송신자와 수신자의 비밀키가 동일하게 생성됨
    - 암호화나 전자서명에 사용되지 않음

공개키 암호 알고리즘

- Diffie-Hellman 암호 알고리즘
  - 이산대수 문제 (Discrete Logarithms Problem)
    - 원시근 (Primitive Root)
      - 소수 p의 원시근
        - 자신의 거듭제곱을 이용하여 1~ $p-1$까X 지의 정수를 생성해 낼 수 있는 수
        - $a \bmod p, a^2 \bmod p, \cdots, a^{p-1} \bmod p$
      - 예시
        - $a^i \bmod p$ $(\mathrm{p} = 13)$인 경우

a \ i	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	8	3	6	12	11	9	5	10	7	1
6	6	10	8	9	2	12	7	3	5	4	11	1
7	7	10	5	9	11	12	6	3	8	4	2	1
11	11	4	5	3	7	12	2	9	8	10	6	1

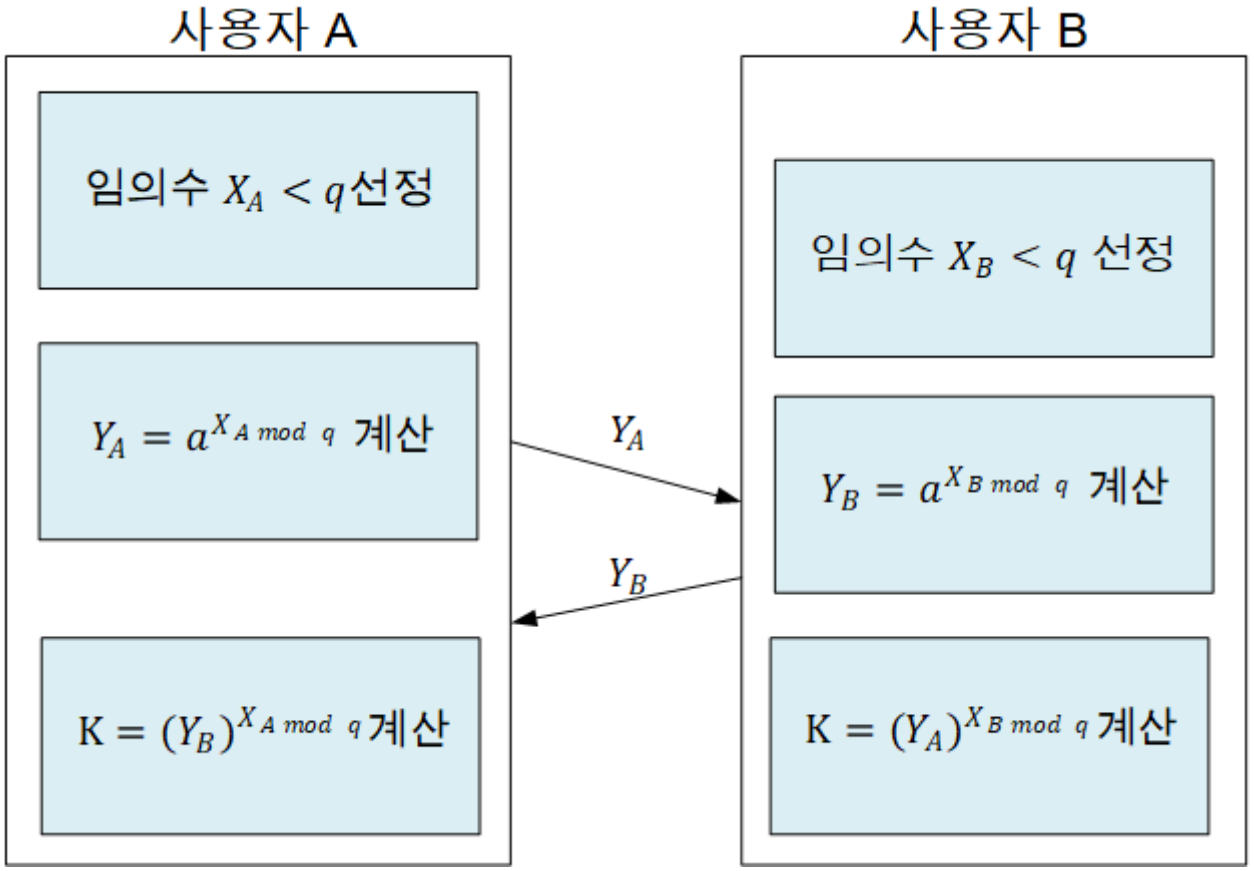
        - 2, 3, 7, 11은 p의 원시근

공개키 암호 알고리즘

- Diffie-Hellman 암호 알고리즘
  - 이산대수 문제 (Discrete Logarithms Problem)
    - 이산대수 (Discrete Logarithms)
      - $b = a^i \ mod \ p \ (0 \leq i \leq p-1)$, (b는 정수)일 때 i를 가리킴

공개키 암호 알고리즘

- Diffie-Hellman 암호 알고리즘
  - 키 교환 프로토콜



공개키 암호 알고리즘

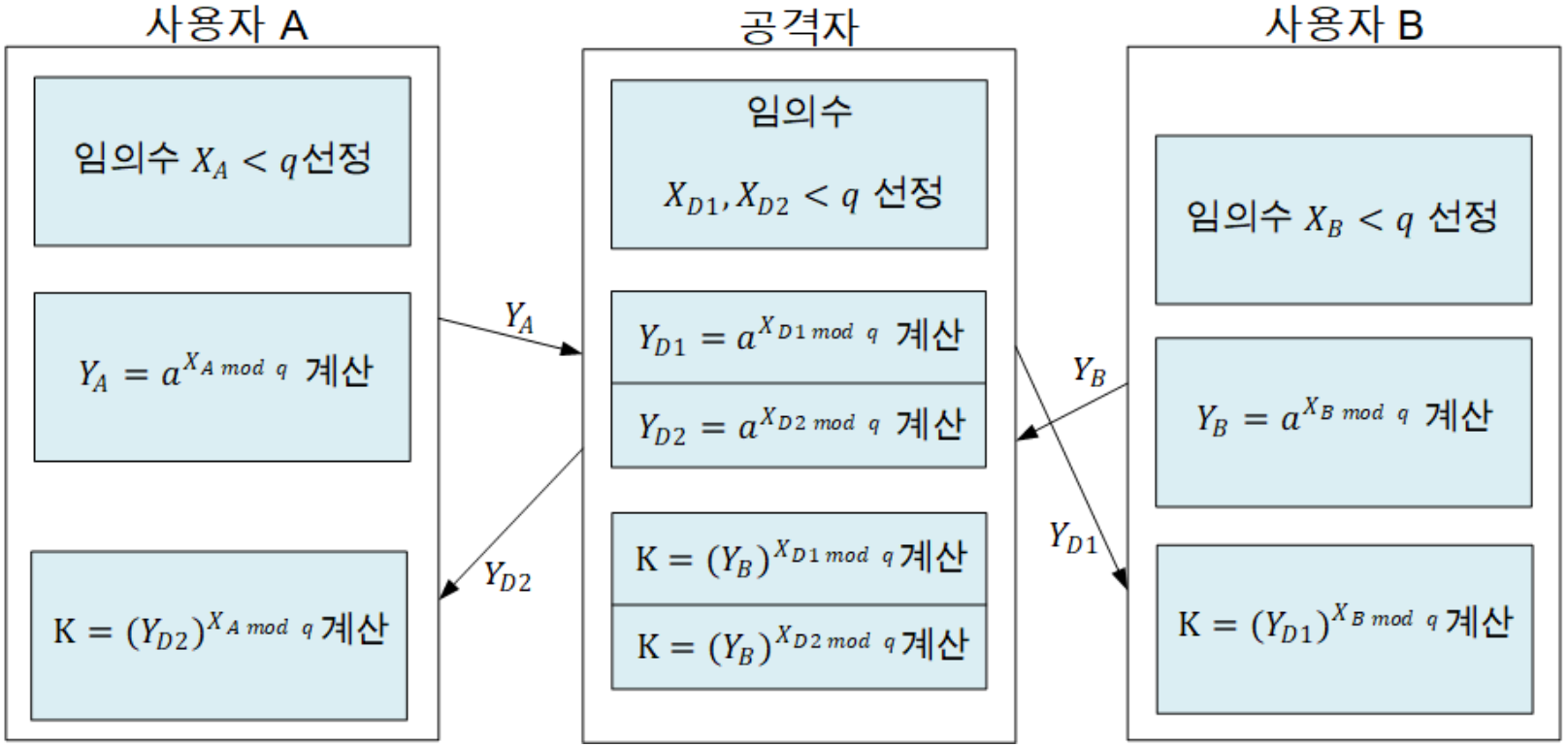
- Diffie-Hellman 암호 알고리즘
  - 키교환 알고리즘
    1. 사용자 A는 랜덤넘버 $X_A < q$ 선택
    2. 사용자 A는 $Y_A = \alpha^{X_A} \, mod \, q$ 계산
    3. 사용자 B는 랜덤넘버 $X_B < q$ 선택
    4. 사용자 B는 $Y_B = \alpha^{X_B} \, mod \, q$ 계산
    5. 사용자 A, B는 각각 X값을 개인값으로 보관, Y값은 공개
    6. 사용자 A는 $K = (Y_B \,)^{X_A} \, mod \, q$을 계산하여 비밀키 생성
    7. 사용자 B는 $K = (Y_B \,)^{X_A} \, mod \, q$을 계산하여 비밀키 생성

공개키 암호 알고리즘

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

공개키 암호 알고리즘

- Diffie-Hellman 암호 알고리즘
  - 중간자 공격 (Man-in-the-Middle Attack)
    - 권한이 없는 공격자가 두 통신 시스템 사이에서 송/수신자 행세를 하며 전달되는 메시지를 가로채는 공격



Thanks!

이 태 양 (taeyang@pel.sejong.ac.kr)