4강.Public Key Cryptosystem

공개키 암호 시스템

중점 내용

- 공개키 암호 알고리즘
- 공개키 암호시스템
- 공개키 암호를 위한 요구사항
- 공개키 암호시스템의 응용
- 공개키 암호 분석
- 결론

공개키 암호 시스템

- 1976년 Merkle에 의해 고안
- Diffie-Hellman(DH)이 개념 정립
- 두 개의 키 사용
  - 암호화 키 : 공개키(public key)
    - 공개키 디렉터리에 공개
  - 복호화 키 : 개인키(private key)
    - 비밀리에 간직

공개키 암호 시스템

- 대표적인 알고리즘
  - RSA : Rivest, Shamir, Adleman
  - ELGamal
  - DSS(Digital Signature Standard)
  - ECC(Elliptic Curve Cryptosystem) 등

- 알고리즘 종류

알고리즘 명	개발자	개발년도	기반 문제
RSA	Rivest, Shamir, Adleman	'78	소인수분해문제
Knapsack	Merkle, Hellman	'78	부분합문제
	Chor, Rivest	'84	
McELiece	McEliece	'78	대수적 코딩론
ELGamal	T. ElGamal	'85	이산대수문제
Chor-Rivest	Chor, Rivest	'84	부분합 문제
Elliptic Curve	Koblitz, Miller	'85	타원곡선 이산대수문제
NTRU	Silverman, Hoffstein, Pipher	'96	다항식 혼합시스템
RPK	Raike	'96	이산대수문제
Lucas	Smith	'94	Lucas 수열 사용
Lattice	Goldwasser, Goldreich, Halevi	'97	근접벡터 찾는 문제

암호방식 비교

구분	대칭키암호방식	비대칭(공개키)
암호화키	비밀	공개
복호화키	비밀	비밀
키의 전송	필요	불필요
관리대상 키	많음	적음
인증, 서명	곤란	용이
암호화속도	빠름	늦음
예	DES	RSA

공개키 암호 시스템

- 다른 명칭
  - 비대칭 암호방식(Asymmetric Key Cryptography)
  - 양방향 암호방식(Two-way Cryptography)
- 안전도(Security level)
  - 소인수분해(Factorization), 이산대수 문제(Discrete logarithm)
- 장점
  - 키 분배(교환)
  - 암호화의 용이
  - 디지털 서명
  - 인증 제공

공개키 암호 시스템

- 용어
  - Pk_b : 사용자 b의 public Key
  - Sk_b : 사용자 b의 secret key
  - Ek[ ] : [ ]을 키로 암호화(Encryption)
  - Dk[ ] : [ ]을 키로 복호화(Decryption)
  - Epk_b[M] : b의 공개키로 메시지(M)을 암호화
  - Dsk_b[C] : b의 비밀키로 암호문(C)를 복호화
  - K : 키(Key)
  - C : 암호문(Ciphertext)
  - M : 메시지(Message) 또는 평문(Plaintext)

공개키 암호 시스템

- 가정
  - 키 Ek와 Dk를 직접 만들어서 Dk는 숨기고, Ek는 공개
  - 암호 알고리즘과 암호문이 공개
- 목표
  - Ek, 암호 알고리즘과 암호문만으로 평문을 알 수가 없다
- 문제점
  - 안전을 위해서는 키 길이가 길고 암호문 작성 시간도 비밀 키 암호방식에 비해 100~1000배 정도 느림

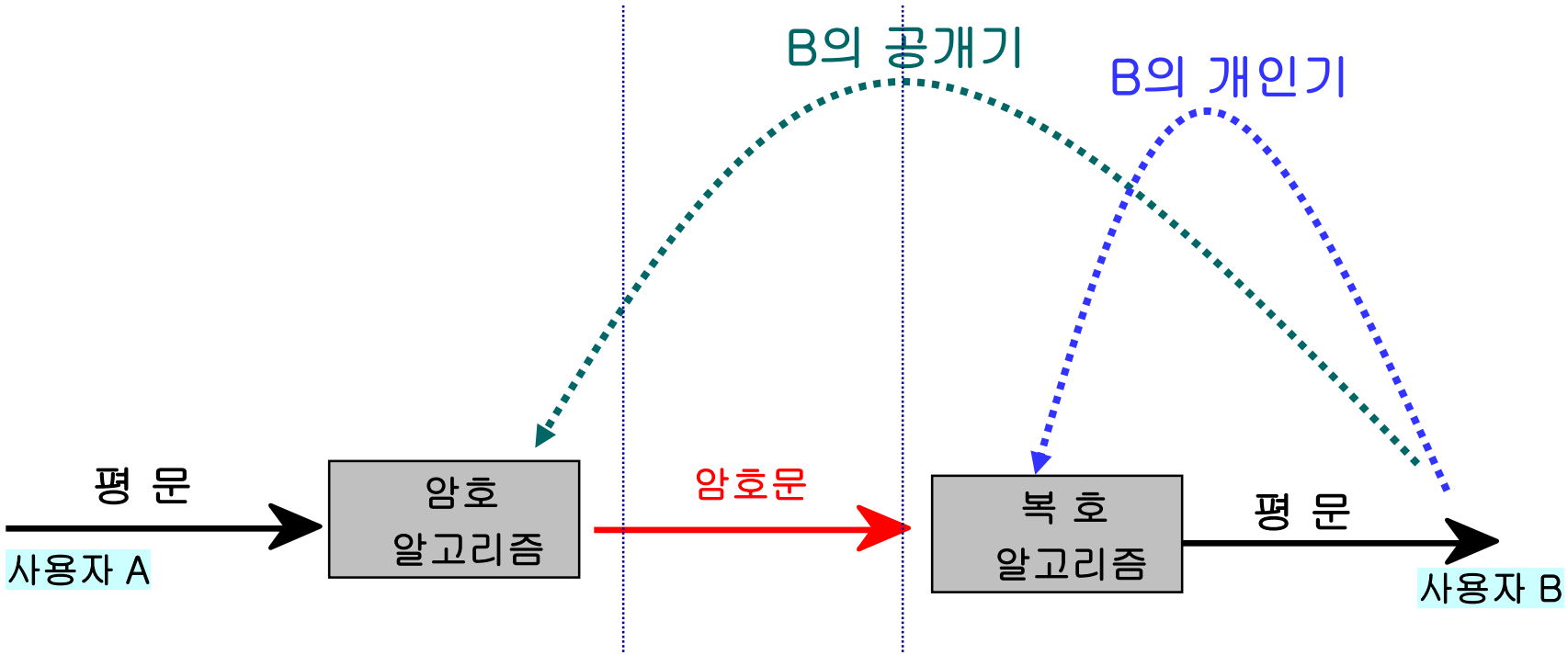
공개키 암호 시스템

- 알고리즘
  - 네트워크상의 각 사용자들은 암/복호화를 위한 키 쌍을 생성
  - 공개키는 공개된 저장소나 파일에 공개, 비밀키는 간직
  - Encryption(암호화)

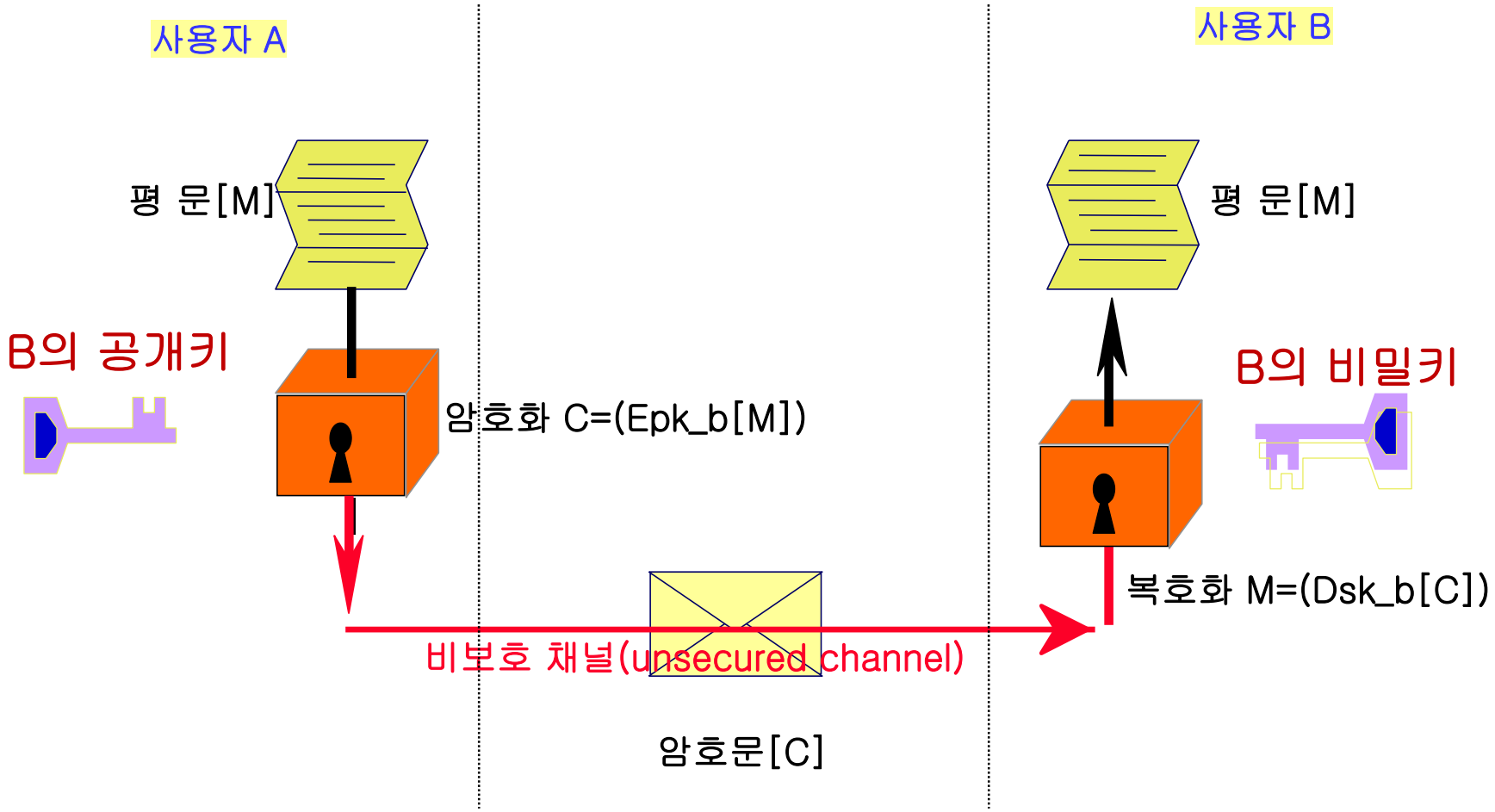
    : 메시지를 전송할 사용자는 상대방의 공개키로 암호화
  - Decryption(복호화)

    : 메시지를 받은 당사자는 자신의 비밀키로 복호화

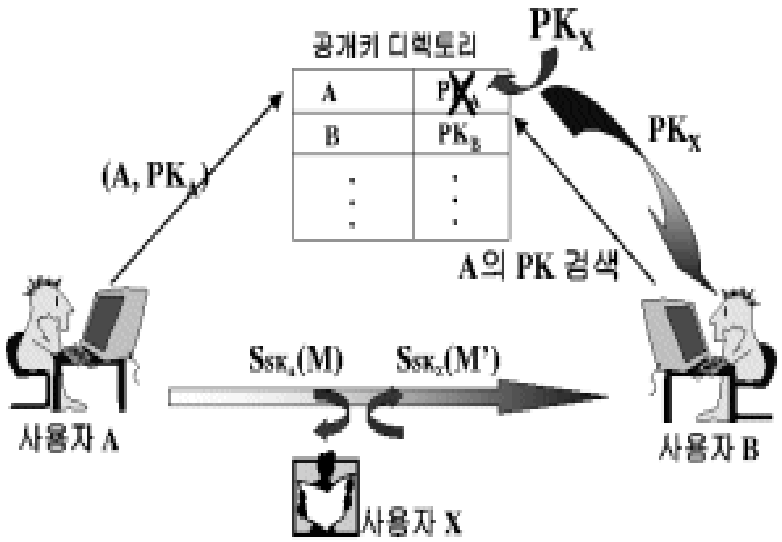
- 공개키 암호시스템의 단순 모델
  - 메시지를 받을 상대방의 공개키로 암호화 실행



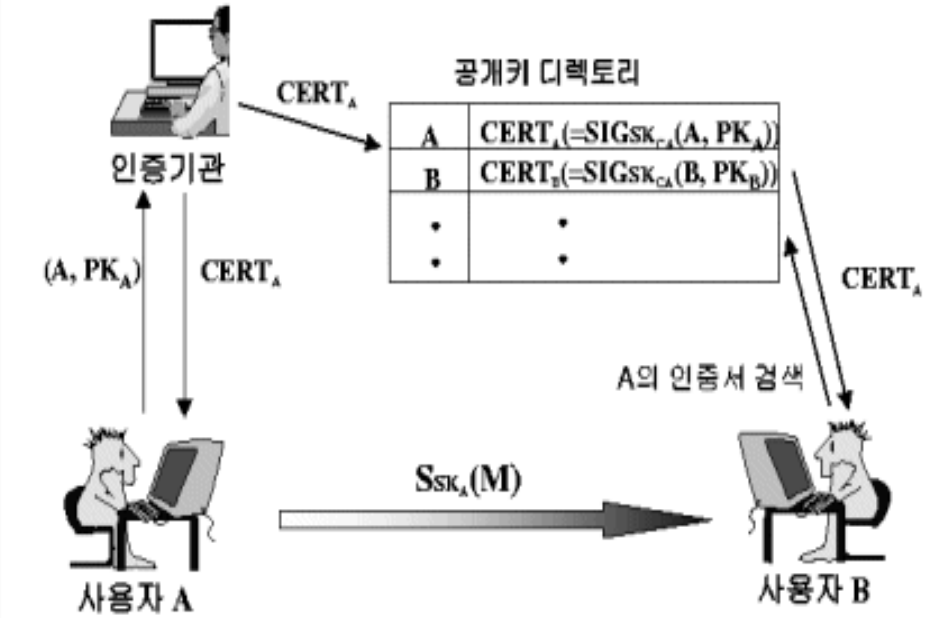
- 공개키 암호시스템의 원리



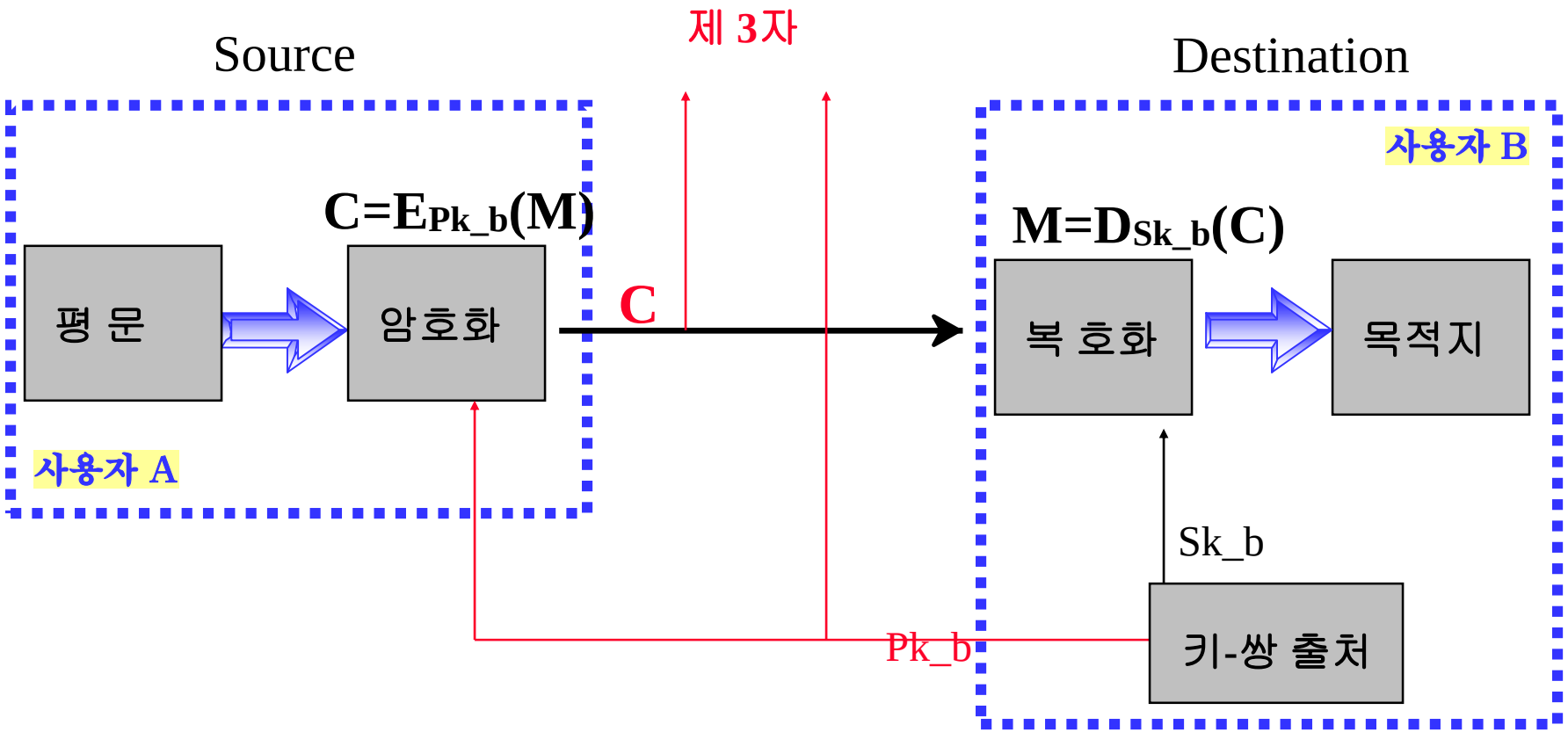
- 공개키 분배


공개키 디렉토리
A
B
PK_X
PK_X
PK_X
PK_B
(A, PK_A)
A의 PK 검색
$S_{SK_A}(M)$
$S_{SK_X}(M')$
사용자 A
사용자 B
사용자 X

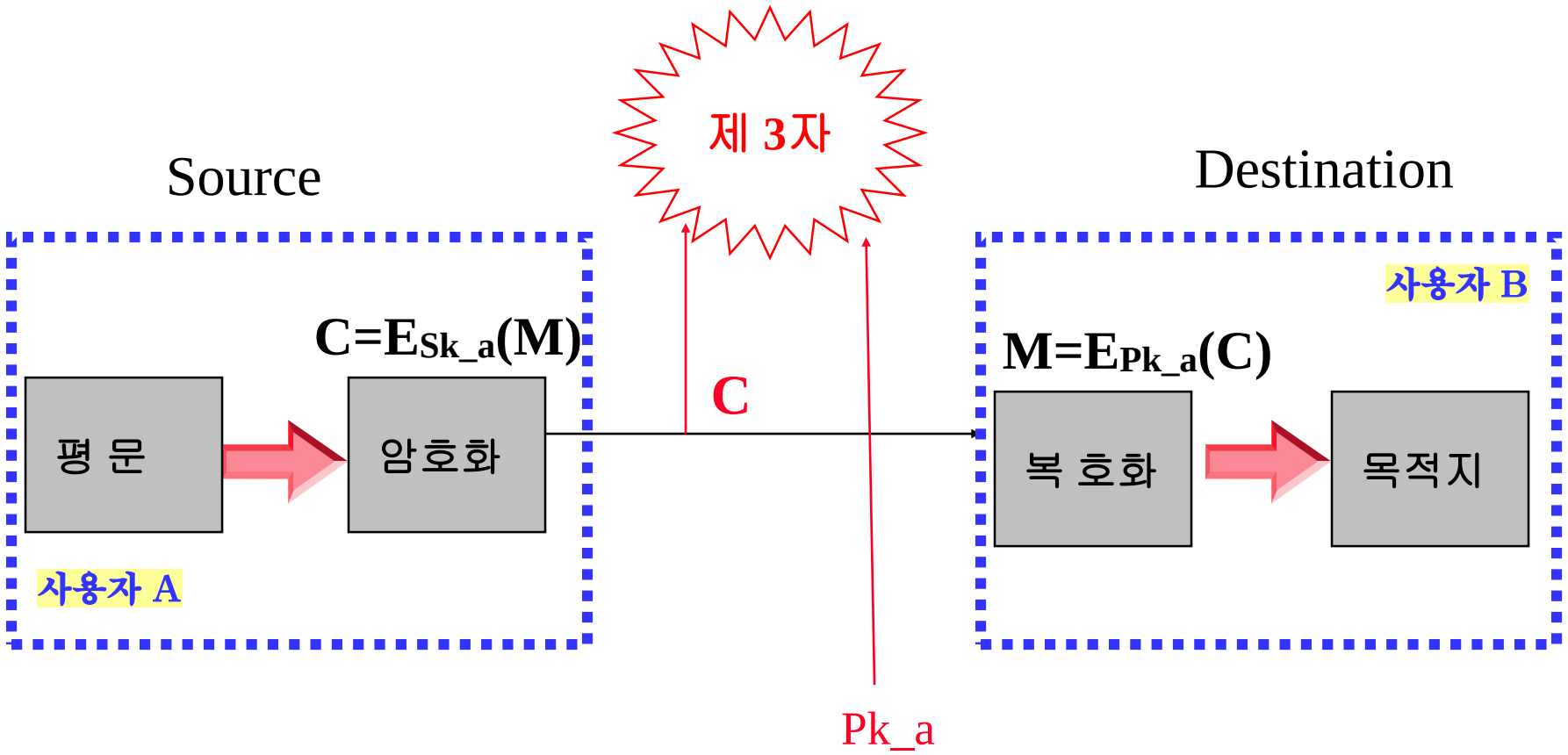
인증서 역할


인증기관
$CERT_A$
공개키 디렉토리
A
B
$CERT_A(=SIG_{SK_{CA}}(A, PK_A))$
$CERT_B(=SIG_{SK_{CA}}(B, PK_B))$
(A, PK_A)
$CERT_A$
$CERT_A$
A의 인증서 검색
$S_{SK_A}(M)$
사용자 A
사용자 B

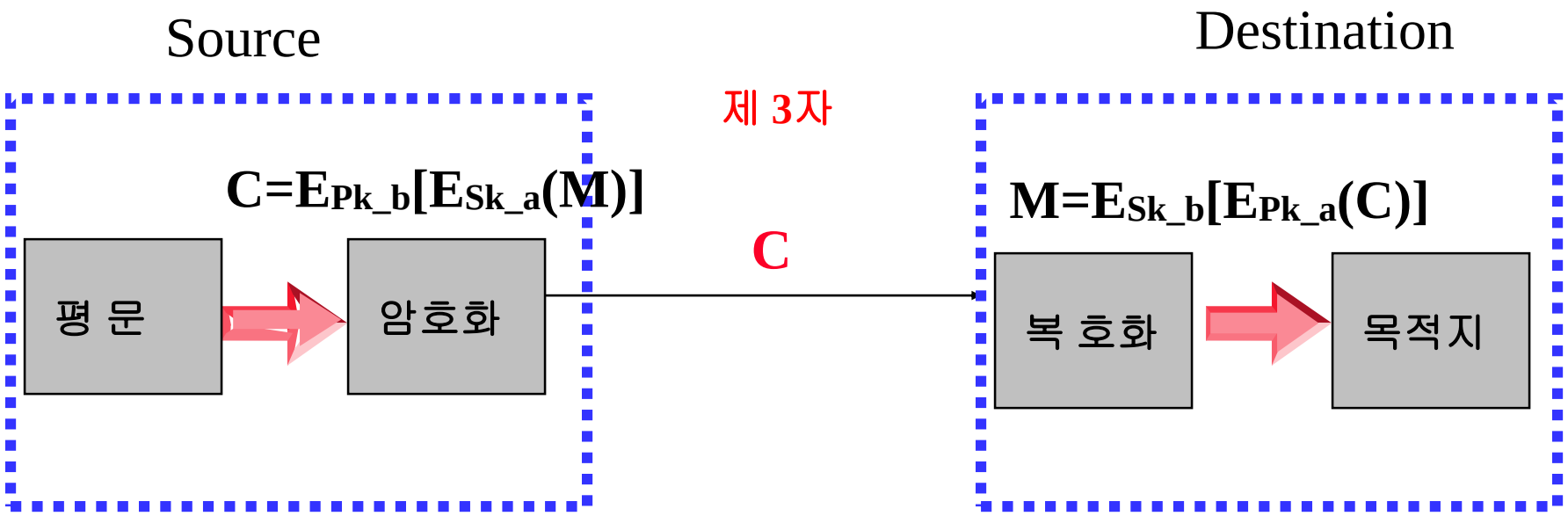
- 비밀(기밀)성
  - 메시지를 받을 상대방의 공개키로 암호화
  - 공개키 암호시스템의 단순 모델과 동일



- 인증(authentication)
  - 메시지를 보낼 당사자의 개인키로 암호화



- 비밀성과 인증


Source
Destination
제 3자
$C=E_{Pk\_b}[E_{Sk\_a}(M)]$
$M=E_{Sk\_b}[E_{Pk\_a}(C)]$
C
평 문
암호화
복 호화
목적지

공개키 암호를 위한 요구사항

- 키 쌍(공개키 Pk_b, 개인키 Sk_b)을 생성하는 것은
  - B에게는 계산량의 관점에서 쉽다.
- 다음과 같은 암호문을 생성하는 것은
  - 공개키와 메시지 M을 아는 송신자 A에게는 계산량의 관점에서 쉽다.
  - $C = E_{Pk\_b}(M)$
- 원래의 메시지를 복구하기 위해 개인키를 사용하여 암호문을 복호화하는 데에 있어서
  - 수신자 B는 계산량의 관점에서 쉽다.
  - $M = D_{Sk\_b}(C)$

공개키 암호를 위한 요구사항

- 개인키 Sk_b를 결정하는 것은

  공개키 Pk_b를 아는 제3자는 계산량의 관점에서 어렵다.

- 원 메시지 M을 복구하는 것은

  - Pk_b와 암호문 C를 아는 제3자는 계산량의 관점에서 어렵다.

즉 소인수분해가 어려운 일방향 함수(one-way function, 단방향 함수)와 비밀통로 함수(trapdoor function)이다.

$Y = f(X)$ 쉽다

$Y = f^{-1}(X)$ 어렵다

- **일방향 함수** (one-way function)

주어진 평문 x에 대해, 암호문 f(x)를 계산하는 것은 용이하나 f(x)로부터 x를 계산하는 것은 계산상 불가능한 함수 f

공개키 암호를 위한 요구사항

- 소인수분해(prime factorization)의 어려움
  - N=pXq, p와 q가 소수

두 수를 소인수분해 해서 중복되는 부분을 찾으면 최대공약수가 됩니다.

예를 들어 $a = 192, b = 72$라고 하면

$$192 = 2^6 \times 3$$
$$72 = 2^3 \times 3^2$$
$$\therefore GCD(192, 72) = 2^3 \times 3 = 24$$

이런 식으로 구할 수 있습니다.

문제는 소인수분해를 하는게 번거롭고 느리다는 겁니다.

  - 큰 수의 소인수 분해를 고속으로 행하는 방법은 아직 발견되지 않았다.

공개키 암호를 위한 요구사항

- 안전성을 위한 요구
  - 두 개의 키 중 하나는 비밀을 유지해야 함
  - 암호 알고리즘, 암호문의 표본, 키 중 한 개의 키를 아는 것으로는 키를 결정하는데 불충분해야 함

공개키 암호 분석

- 키 관리의 용이성
  - 비밀키만 보관
  - 사용자가 증가해도 관리해야 하는 키는 비밀키 뿐이며 기능적으로 믿을 만한 제3의 신뢰 기관(인증기관)만 있으면 됨
- 디지털 서명으로의 쉬운 변형
  - 서명 생성 : $S = h(M)^d \mod n$
  - 서명 검증 : $h(M) = S^e \mod n$
- 여러 분야에서 쉽게 응용 가능

공개키 암호 분석

- 암호화/복호화 속도가 비밀키 암호 시스템에 비해 매우 느림
- 키의 길이가 비밀키 암호시스템에 비해 상대적으로 큼
- 전사적 공격에 취약 ⇒ 큰 키의 사용 ⇒ Trade off

공개키 암호 시스템의 응용

- 암호 및 복호
  - 송신자는 수신자의 공개키로 암호화
- 디지털 서명
  - 송신자는 개인키로 메시지를 서명
    - 서명은 메시지에 암호알고리즘을 적용하거나 메시지의 단위를 이루는 작은 데이터 블록에 암호 알고리즘을 적용하여 얻는다.
- 키 교환(key exchange)
  - 양쪽은 세션 키를 교환하기 위하여 상호 협력

공개키 암호 시스템

- 공개키 암호화 강도

키 길이(bit)	해독 시간(추측 가능성)
256	누구나 쉽게 인수분해 가능
384	대학이나 연구기관에서 해독 가능
512	정부기관에서 가능
1024	현재까지는 안전한 수준
2048	수십 년간 안전한 키의 길이

응용 분야

알고리즘	암호/복호화	디지털 서명	키 교환
RSA	가능	가능	가능
LUC	가능	가능	불가능
DSS	불가능	가능	불가능
Diffie-Hellman	불가능	불가능	가능

RSA

- 1978년에 미국 MIT에서 개발
- RSA : Rivet, Shamir, Adelman 세 사람의 첫 이름
- 소인수분해의 어려움

Gen(): Set $n = pq$ where p and q are large primes
Select e such that $\gcd(e, \phi(n)) = 1$ where $\phi(n) = (p-1)(q-1)$
Find d of e such that $ed = 1 \bmod \phi(n)$

$PK = (n,e)$ and $SK = (p,q,d)$

Enc(M, PK): $C \equiv M^e \bmod n$

Dec(C, SK): $M \equiv C^d \bmod n$

RSA

[1단계] 두 개의 큰 소수 p, q를 찾는다

- p와 q는 비밀정보(1024 bit나 2048 bit 정도)

[2단계] 두 소수를 곱하여 n=p*q을 생성

- n은 공개 정보

[3단계] 두 소수를 다른 방법으로 결합

- $\phi(n) = (p-1)(q-1)$

[4단계] 파이(ϕ)를 사용하여 e, d 키 쌍을 생성

- $ed = 1 \pmod{\phi(n)}$ e=(p-1, n)

  $\equiv 1 \pmod{(p-1)(q-1)}$

[5단계] e, d, n을 이용하여 암·복호화

- encryption
  - $C \equiv P^e \pmod{n}, \qquad 0 \le C < n$

- decryption
  - $P \equiv C^d \pmod{n}$
    $\equiv (P^e)^d \pmod{n}$

✓ 이산대수 문제 : 암호문으로부터 평문 구하기

✓ 암호문 = $(\text{평문})^e \bmod N$

✓ 현재까지 아직 이산대수를 구하는 빠른 방법을 알지 못함

RSA

- 소수 선택의 조건
  - 두 소수 p, q의 크기가 거의 같아야 함
  - p-q가 너무 작으면 안됨
  - p-1이 큰 수를 인수로 가져야 함
  - p+1이 큰 수를 인수로 가져야 함

RSA

- 예) 공개키와 개인키 생성

  [1] 두 소수 $p = 7$, $q = 17$ 을 선택

  [2] $n = p \cdot q = 7 \times 17 = 119$ 계산

  [3] $\phi(n) = (p-1)(q-1) = 96$ 계산

  [4] $\phi(n) = 96$과 서로 소이고, $\phi(n)$보다 작은 e 선택 $(e=5)$

  [5] $de = 1 \mod 96$이고, $d < 96$인 d를 결정 $(d=77)$

  $\Rightarrow$ 공개키 $KU = \{5, 119\}$, 개인키 $KR = \{77, 119\}$

- 예제1

[1] 두 소수 $p = 3$, $q = 11$ 선택

[2] $n = pq = 3 \times 11 = 33$

[3] $\phi(n) = (p-1)(q-1) = 2 \cdot 10 = 20$

[4] e 결정하기

$\phi(n)=20$과 서로 소($\gcd(e, \phi(n))=1$)의 관계인 임의의 정수 선택

e= 3로 선정

[5] d(개인키) 결정하기

유클리드 알고리즘($de = 1 \mod 20$)과 $d < 20$은 d를 결정

d = 7로 선정

[6] 암호화 : $M=5$, $5^3 \mod 33 = 26$

복호화 :

- 예제2

[1] 두 솟수 $p = 47$, $q = 71$ 을 선택

[2] $n = pq = 47 \times 71 = 3337$

[3] $\phi(n) = (p-1)(q-1) = 46 \cdot 70 = 3220$

[4] e 결정하기

$\phi(n) = 3220$과 서로 소의 관계인 임의의 정수 선택

$e = 79$로 선정

[5] d(개인키) 결정하기

유클리드 알고리즘($de = 1 \mod 3220$)과 $d < 3220$은 d를 결정

$d = 1019$로 선정

$\Rightarrow$ Public key = {79, 3337}, Private key = {1019, 3337}

- 예제2) cont'd
  - encryption
    - m = 688로 가정, e= 79
    - $M^e \mod 3337 = 1570$
  - decryption
    - $c^d \mod 3337 = (m^e)^d \mod 3337 = 1570$
    - $1570^{1019} \mod 3337 = 688$

RSA

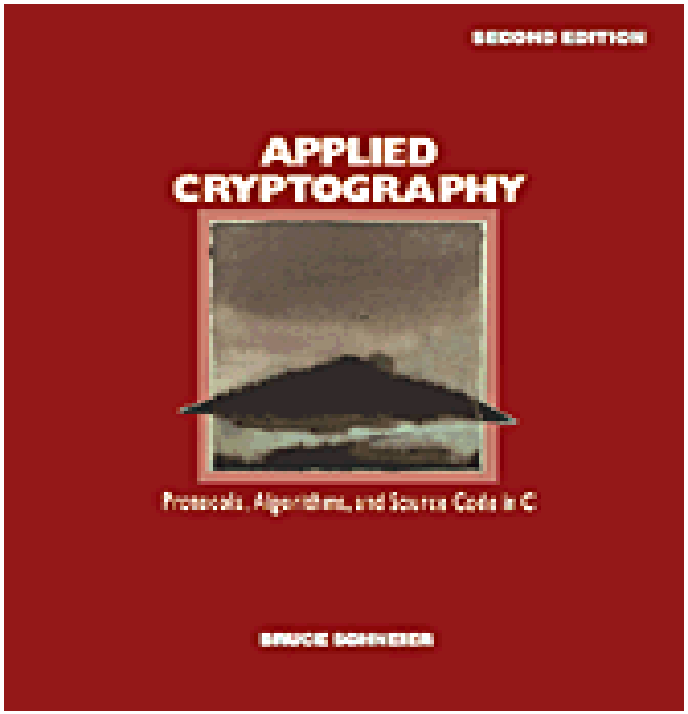
- Fermat 정리
  - p가 소수이고, a와 p가 서로 소이면, a^{p-1} mod p = 1.
- 오일러 정리(Euler theorem)
  - a와 m이 정수이고, gcd(a, m)=1 일 때, .
  - n과 서로 소의 관계에 있는 모든 a에 대해 $a^{\varphi(m)} \equiv (mod\, m)$ = 1.
  - Φ(n) : n보다 작은 자연수 중에서 n과 서로 소인 자연수의 수

결 론

- 공개키 암호시스템은
  - 소인수분해와 이산대수를 이용하여 공개키와 비밀키 쌍을 생성
  - 키의 분배(교환)와 비밀성, 서명, 인증, 부인 봉쇄 등의 기능을 가지며 다양한 분야에서 응용
  - 보안 공격의 불법수정, 위조로부터 안전함

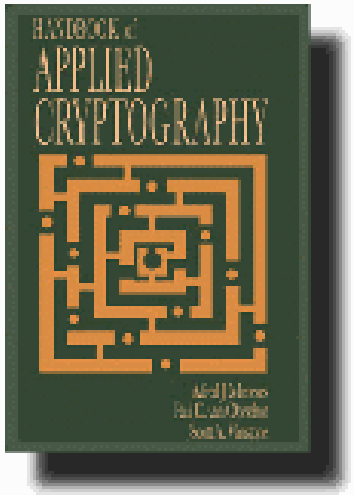
한 권의 책

- Applied CRYPTOGRAPHY



한 권의 책

- Handbook of Applied CRYPTOGRAPHY



Handbook of
APPLIED
CRYPTOGRAPHY