

# 제 12 장 키



**박 종 혁 교수**

**Tel: 970-6702**

**Email: [jhpark1@seoultech.ac.kr](mailto:jhpark1@seoultech.ac.kr)**

**1절 키란 무엇인가?**

**2절 다양한 키**

**3절 콘텐츠를 암호화하는 키와 키를 암호화 하는 키**

**4절 키 관리**

**5절 Diffie-Hellman 키 교환**

**6절 패스워드를 기초로 한 암호(PBE)**

**7절 안전한 패스워드를 만들려면**

# 제1절 키란 무엇인가?

**1.1 키는 대단히 큰 수**

**1.2 키는 평문과 동일한 가치를 갖는다**

**1.3 암호 알고리즘과 키**

## 1.1 키는 대단히 큰 수

- 암호 기술을 사용하려면 반드시 키(key)라 불리는 대단히 큰 수가 필요
- 중요한 것은 수 그 자체의 크기라기보다도 **키 공간(Key Space)의 크기**
- 키 공간의 크기는 키의 비트 길이로 결정

# 암호별 키 길이

- DES: 56비트
- 3DES
  - DES-EDE2: 112비트
  - DES-EDE3: 168비트
- AES: 128, 192, 256
- RSA: 1024, 2048

## • 보안강도

- 암호알고리즘/시스템의 암호화 키 또는 해시함수의 취약성을 찾아내는데 소요되는 작업량을 수치화한 것 (112, 128, 192, 256 비트)  
 . 112비트 보안강도:  $2^{112}$ 번의 계산을 해야 암호키/ 알고리즘의 취약성을 알아낼 수 있음

## • 보안강도별 암호알고리즘 비교

(\*NIST SP 800-57 권 고)

| 보안강도   | 대칭키<br>암호<br>알고리즘<br>(보안강도) | 해시함수<br>(보안강도) | 공개키 암호 알고리즘  |         |         |                | 암호 알고리즘<br>안전성<br>유지기간<br>(년도) |
|--------|-----------------------------|----------------|--------------|---------|---------|----------------|--------------------------------|
|        |                             |                | 인수분해<br>(비트) | 이산대수    |         | 타원곡선<br>암호(비트) |                                |
|        |                             |                |              | 공개키(비트) | 개인키(비트) |                |                                |
| 112 비트 | 112                         | 112            | 2048         | 2048    | 224     | 224            | 2011년에서<br>2030년까지             |
| 128 비트 | 128                         | 128            | 3072         | 3072    | 256     | 256            | 2030년 이후                       |
| 192 비트 | 192                         | 192            | 7680         | 7680    | 384     | 384            |                                |
| 256 비트 | 256                         | 256            | 15360        | 15360   | 512     | 512            |                                |

# DES 키 예

- 2진수 표현(56비트):

01010001 11101100 01001011 00010010  
00111101 01000010 00000011

- 16진수 표현:

51 EC 4B 12 3D 42 03

- 10진수 표현:

23059280286269955

# RSA 키 예

- 16진수 표현(1024비트):

00:bc:04:e4:fa:13:39:f0:34:96:20:6b:6c:68:bb:fa:db:77:ff:27:f7:ac:e  
c:2f:e7:fd:f0:7f:6d:6f:8c:2a:cd:25:09:5b:24:f4:a1:68:fc:28:ec:c9:25:  
e2:ac:ed:de:c8:33:84:f5:b0:a5:09:3a:a7:b1:47:48:c5:cc:4f:8c:79:9c:  
f9:06:57:7d:dd:ee:38:f6:cf:14:b2:9c:ea:d3:c0:5d:77:62:f0:47:0d:b9:  
1a:40:53:5c:64:70:af:08:5a:c0:f7:cf:75:f9:6c:8d:64:28:1e:20:fe:b7:1  
b:19:d3:5a:66:83:72:e2:b0:9b:bd:d3:25:15:0d:32:6f:64:37:94:85:46:  
c8:72:be:77:d5:6e:1f:28:2f:c7:69:ed:e7:83:89:33:58:d3:de:a0:bf:40:  
e8:43:50:ee:dc:4d:6b:bc:a5:ea:a6:c8:61:8e:f5:c3:64:af:06:15:dc:29:  
8b:3f:75:8c:bc:71:44:db:fc:ad:b5:17:1d:6d:89:83:cf:c6:33:bd:bf:45:  
a2:fe:0a:9f:a3:11:5f:0f:b9:1f:9c:1a:c2:46:cc:9c:28:66:9f:70:26:3c:2  
e:df:aa:80:fe:8c:c5:04:09:25:4f:cd:93:47:3c:37:ea:02:67:92:fe:fc:22  
:24:5c:ac:d2:2c:e0:5c:01:33:8a:c1:19:db



## 1.2 키는 평문과 동일한 가치를 갖는다

- 키는 평문과 같은 가치
  - 도청자 이브에게 「키가 넘어가는 것」은 「평문이 넘어가는 것」과 같은 것

## 1.3 암호 알고리즘과 키

- 암호의 기본 상식:
  - 검증된 암호 알고리즘을 사용
  - 정보의 기밀성은 암호 알고리즘을 비밀로 하는 것이 아님
    - 감추는 것에 의한 보안(security by obscurity)은 전형적인 잘못된 생각
  - 키를 비밀로 하는 것에 의해 기밀성이 지켜져야 함

## **제2절 다양한 키**

**2.1 대칭 암호 키와 공개 키 암호 키**

**2.2 메시지 인증 코드 키와 디지털 서명 키**

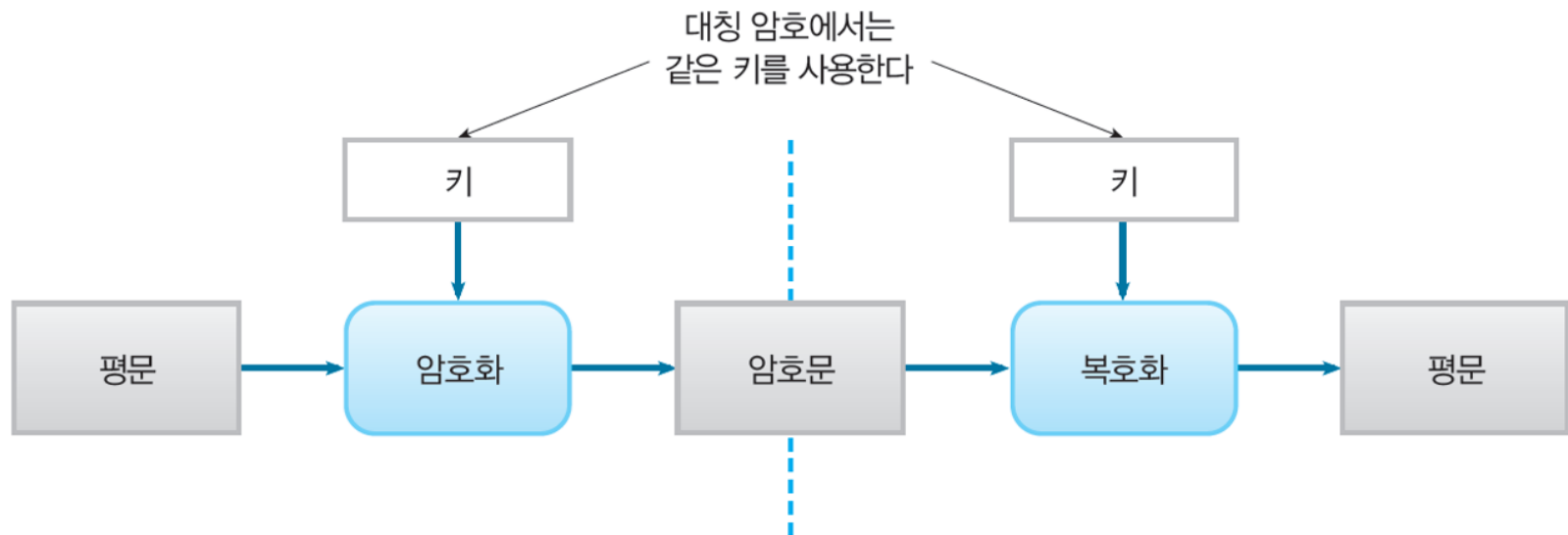
**2.3 기밀성을 위한 키와 인증을 위한 키**

**2.4 세션 키와 마스터 키**

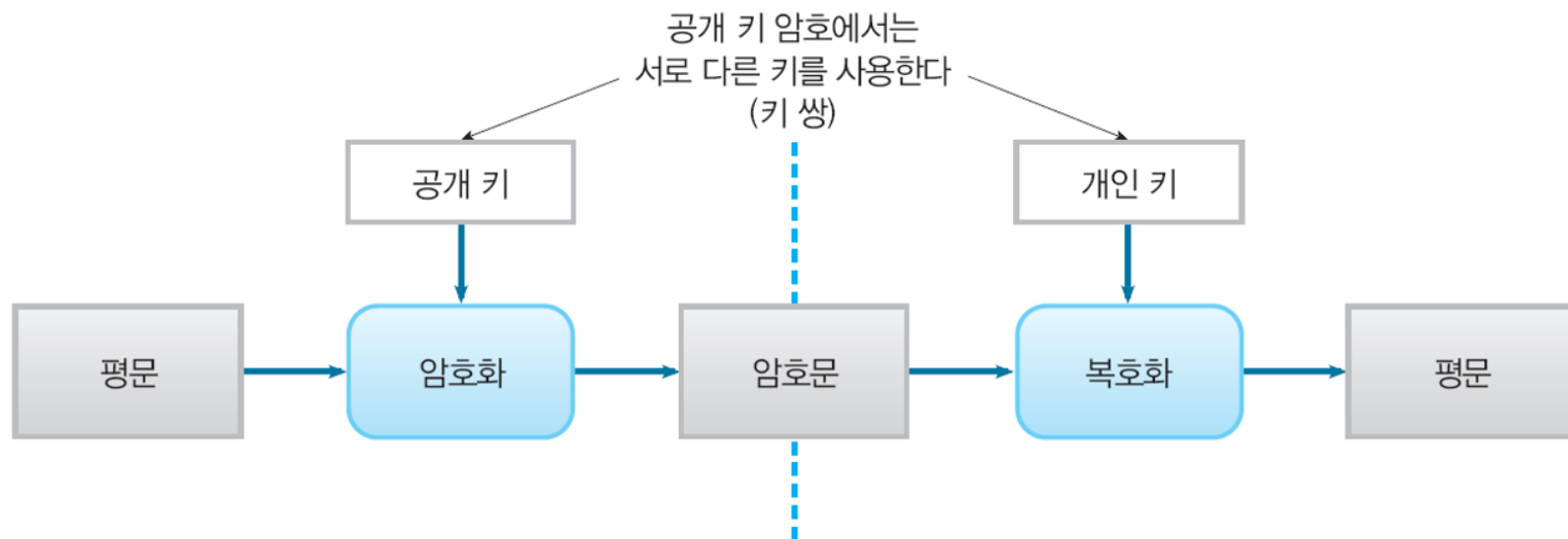
## 2.1 대칭 암호 키와 공개 키 암호 키

- 대칭 암호
  - 키는 송신자와 수신자만 공유
  - 양측이 공유 키를 비밀로 유지
- 공개 키 암호
  - 암호화와 복호화에서 다른 키 사용
  - 개인 키를 비밀로 유지

# 대칭 암호는 암호화와 복호화에서 공통 키를 사용



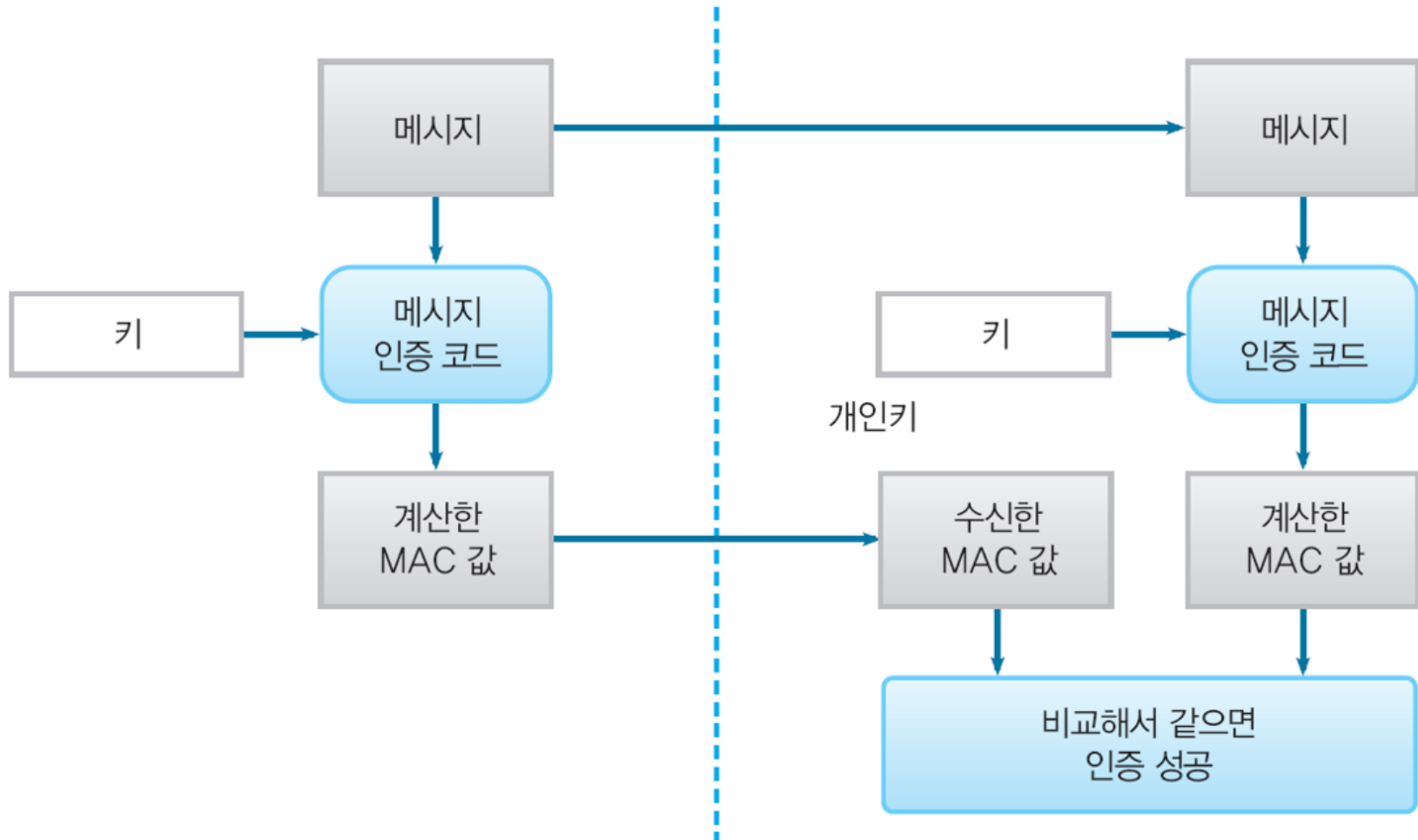
# 공개 키 암호는 공개 키로 암호화하고, 개인 키로 복호화



## 2.2 메시지 인증 코드 키와 디지털 서명 키

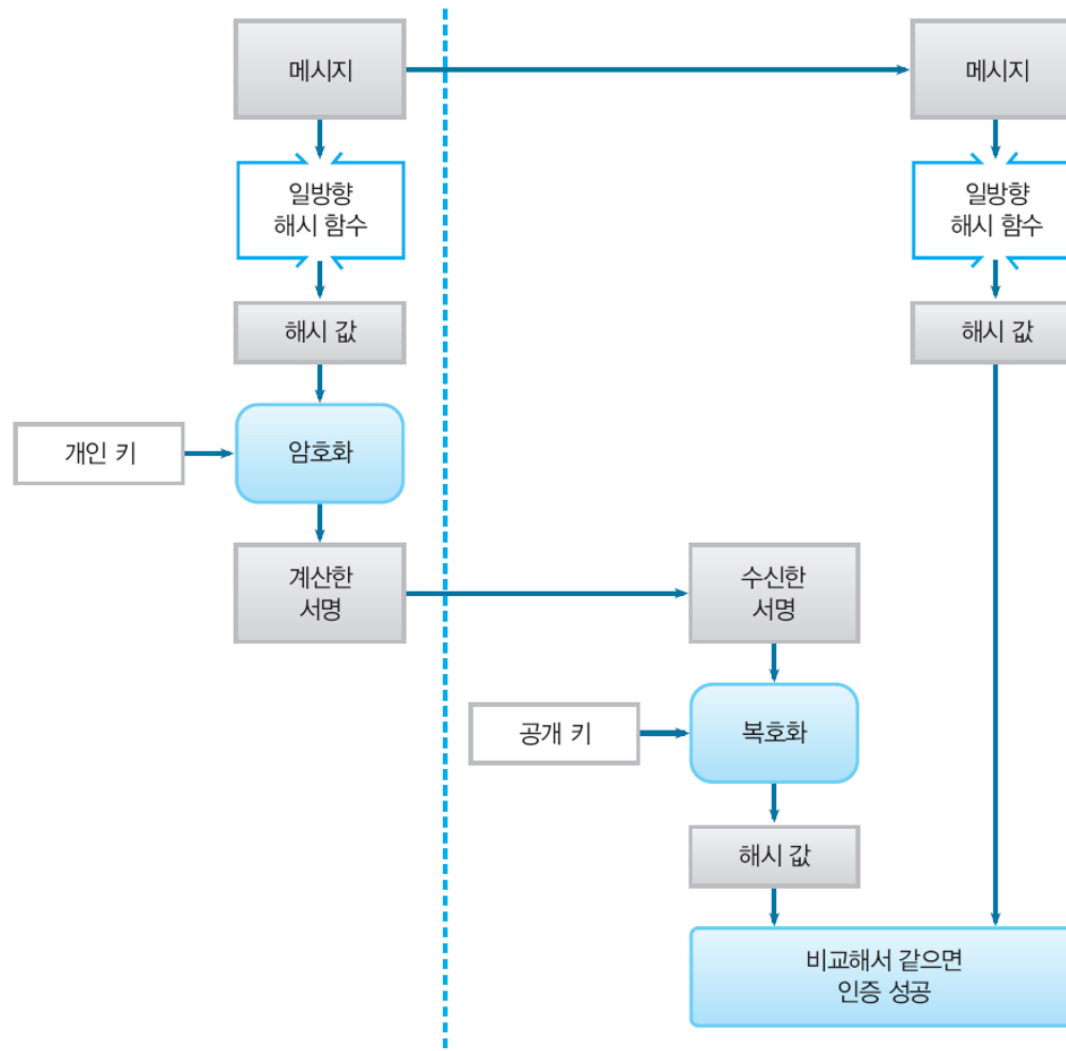
- 메시지 인증 코드
  - 송신자와 수신자가 공통의 키를 사용해서 인증을 수행
- 디지털 서명
  - 서명 작성과 서명 검증에 서로 다른 키를 사용

# 메시지 인증 코드 키





# 디지털 서명 키



## 2.3 기밀성을 위한 키와 인증을 위한 키

- 보안 속성에 따른 분류

- 기밀성을 유지하기 위한 키:

- 대칭 암호나 공개 키 암호에서 사용하는 키
    - 복호화 키를 모르면 복호 불가

- 인증을 수행하기 위한 키:

- 메시지 인증 코드나 디지털 서명에서 사용하는 키
    - 키를 모르면 데이터 변경이나 위장 불가

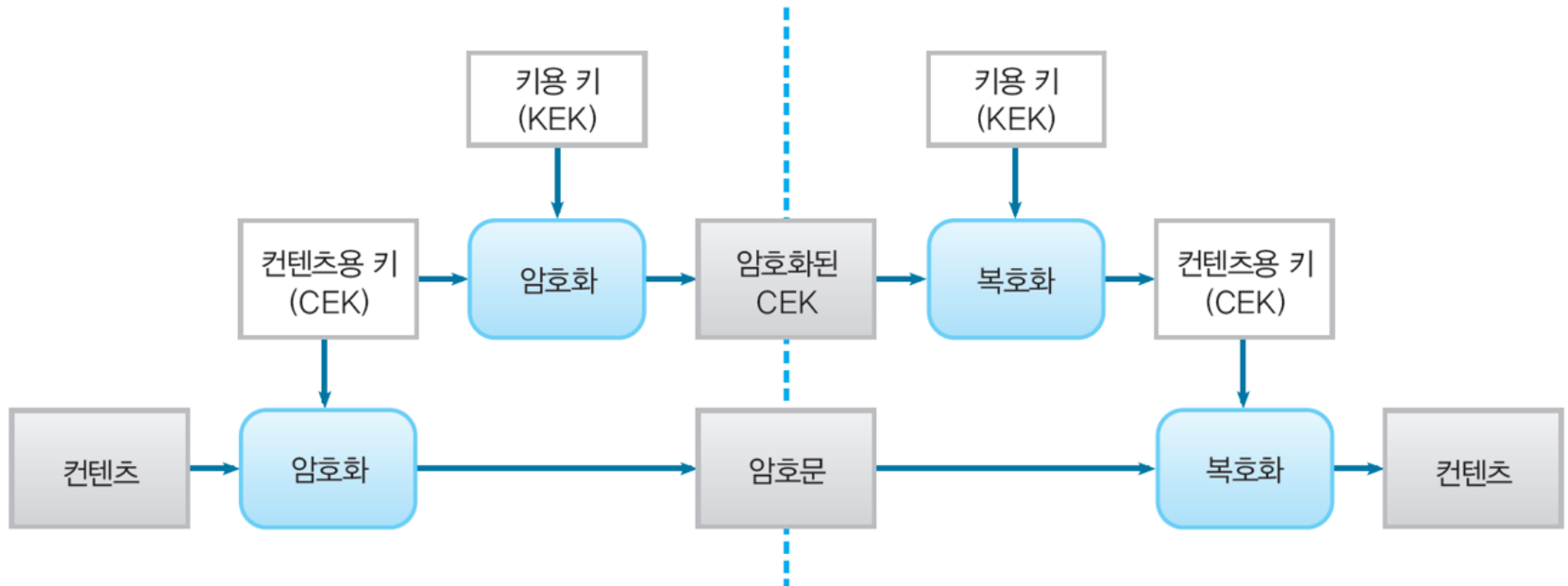
## 2.4 세션 키와 마스터 키

- 키 사용 횟수에 따른 분류
  - 세션 키(session key):
    - 통신 때마다 한 번만 사용되는 키
  - 마스터 키(master key)
    - 반복적으로 사용되는 키

## 제3절 콘텐츠를 암호화하는 키와 키를 암호화 하는 키

- 키를 사용할 때 암호화 대상에 따른 분류
  - CEK(contents encrypting key):
    - . 정보(콘텐츠)가 암호화의 대상
  - KEK (key encrypting key):
    - . 키가 암호화의 대상

# 콘텐츠를 암호화하는 키(CEK)와 키를 암호화하는 키(KEK)



# 제4절 키 관리

**4.1 키 생성**

**4.2 키 배송**

**4.3 키 갱신**

**4.4 키 보존**

**4.5 키 폐기**

## 4.1 키 생성

- 난수를 이용한 키 생성
- 패스워드를 이용한 키 생성

# 난수를 이용한 키 생성

- 난수 사용

- 이유: 키 성질로 「다른 사람이 추측하기 어려워야 한다」를 가져야 하기 때문
- 난수는 추측하기 어렵기 때문에 키로 적합

- 난수 생성

- 하드웨어를 사용하는 것이 좋지만
- 통상적으로 암호용으로 설계된 의사난수 생성기 소프트웨어를 사용



# 의사난수 만들기

- 자신이 적당한 바이트 열을 만들면 안 됨
  - 이유: 스스로는 랜덤한 값이라고 생각하고 생성해도, 거기에는 아무래도 인위적인 편중이 있기 때문에 랜덤한 값이 되지 못함
- 암호용으로 이용하는 의사난수 생성기
  - 반드시 암호용으로 설계되어 있는 것을 선택
    - 이유: 암호용으로 설계되어 있지 않은 의사난수 생성기는 「예측 불가능」 성질을 갖지 않기 때문

# 패스워드

- 패스워드(password) 혹은 패스프레이즈(passphrase)로부터 키를 만드는 경우도 있음
- 패스프레이즈: 복수의 단어로 이루어지는 긴 문장의 패스워드
  - 비밀번호보다 길고 기억하기 쉬운 문장을 사용
    - . 예) Iliketoeat2banners
- 패스워드를 키로 직접 이용하지 않고, 패스워드를 일방향 해시 함수에 입력해서 얻어진 해시 값을 키로 이용

# PBE와 솔트(salt)

- 「패스워드를 기초로 한 암호」 (password based encryption; PBE)
  - 일방향 해시 함수의 입력 : 패스워드 + 솔트(salt) : 난수  
출력을 키로 사용
  - 사전 공격(dictionary attack)을 막기 위한 조치

## 4.2 키 배송

- 키 배송 문제
  - 키를 사전에 공유하는 방법
  - 키 배포 센터를 이용하는 방법
  - 공개 키 암호를 사용하는 방법
  - Diffie-Hellman 키 교환

## 4.3 키 갱신

- 키 갱신(key updating)
  - 공통 키를 사용하여 통신을 하고 있는 중에 정기적으로(예를 들면 1000문자 통신할 때마다) 키를 교환해 가는 방법
  - 송신자와 수신자가 동시에 같은 방법으로 키를 교환해야만 함
  - 현재 키의 해시 값을 다음 키로 사용

# 키 갱신의 장점

## [Backward Security와 Forward Security]

- Backward Security / Secrecy (후방향 보안/성)
  - 새로운 세션 키가 생성되더라도 이전에 사용된 세션 키의 기밀성이 유지되는 특성
  - 특정 시점에 암호화 키가 유출되었더라도, 그 이후에 생성된 암호화된 데이터는 여전히 안전하게 보호  
(과거의 키가 탈취되더라도 현재의 데이터가 보호되도록 보호)
  - 새로운 키를 통해 기존 메시지에 대한 암호 해독을 방지
  - **키관리**: 각 세션에서 생성된 새로운 키가 이전 키를 기반으로 생성
  - PGP (Pretty Good Privacy), S/MIME 등 이메일 암호화 시스템, 블록체인 기술에 활용

## 4.4 키 보존

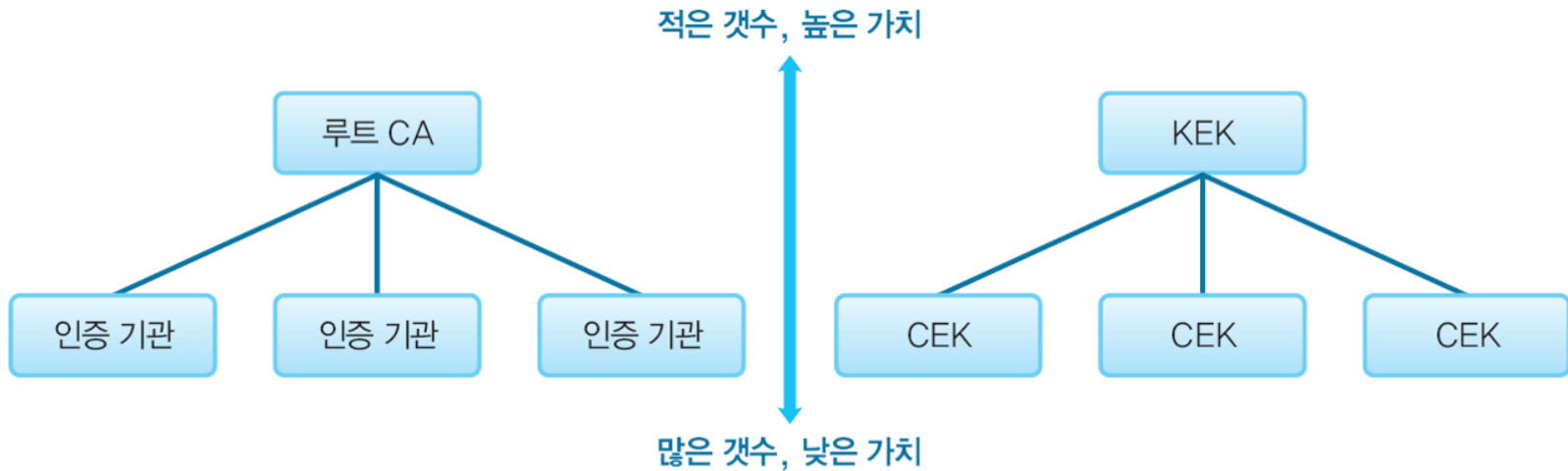
- 키를 반복해서 사용할 경우 키 보존 문제를 고려
  - 키 기억
    - .보통 실용적 키의 크기나 비트화된 표현 등으로 기억할 수 없음
  - 키 암호화
    - .키를 암호문과 동일한 컴퓨터 내에 두는 것은 어리석은 짓
    - .파일 형태로 보존된 키를 금고 등의 안전한 장소에 보관 (공간적 제약)
    - .키를 암호화해서 보존하는 기술을 사용

# 키를 암호화하는 키

- KEK(Key Encryption Key)
  - 키를 암호화하는 키
  - 다수의 키를 한 개의 키(KEK)로 암호화 하여 보관
  - 구현방법
    - 하드웨어 보안 모듈(HSM)을 사용하여 KEK를 안전하게 저장하고, 암호화 및 복호화 작업을 수행
    - 클라우드나 서버에서 제공하는 AWS KMS (소프트웨어 기반 키 관리 시스템), Google Cloud KMS 등과 같은 키 관리 시스템을 이용



# 인증 기관의 계층화와 키 계층의 비교



- 가치의 높이: 뚫렸을때 피해의 크기

## 4.5 키 폐기

- 왜 키를 버리지 않으면 안 될까?
  - 불필요한 키를 도청자가 사용한다면 암호문을 복호화 가능
  - 불필요해진 키는 확실히 삭제
- 어떻게 버리는 것인가?
  - 암호 소프트웨어뿐만 아니라 컴퓨터 전체가 보안을 염두에 두고 설계 (삭제 복구 가능성도 고려)
- 키를 잃어버리면 어떻게 될까?
  - 대칭 암호의 공유 키 분실 (복호화?)
  - 메시지 인증 코드 키 (인증?)
  - 공개키 암호의 개인키 분실 (복호화? 서명작성?)

# 제5절 Diffie-Hellman 키 교환

## 5.1 Diffie-Hellman 키 교환

## 5.2 Diffie-Hellman 키 교환의 수순

## 5.3 이브는 키를 계산 할 수 없는 것일까?

## 5.4 원시근의 의미

## 5.5 구체적 키 교환의 예

## 5.6 타원 곡선 Diffie-Hellman 키 교환

## 5.1 Diffie-Hellman 키 교환

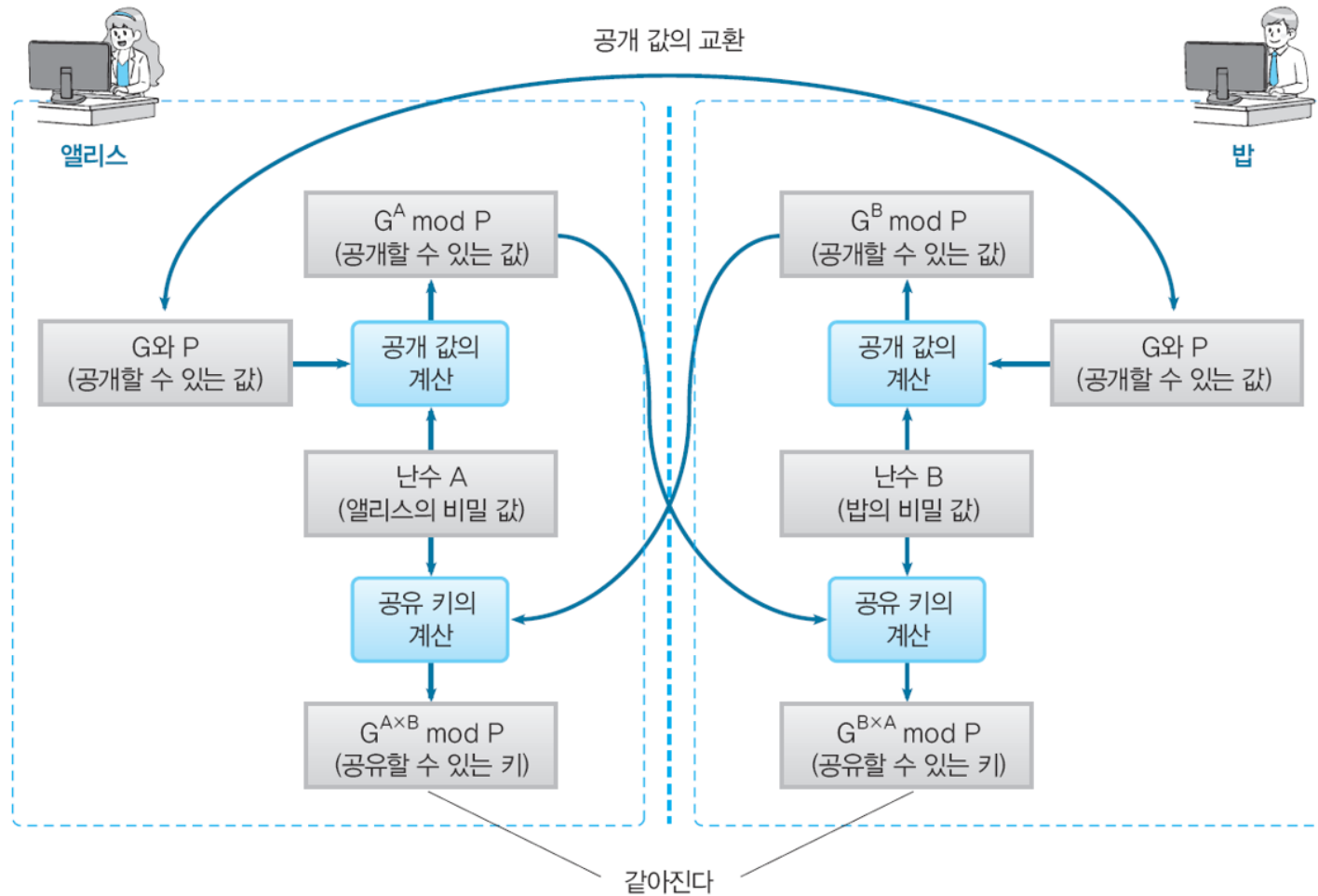
- Diffie-Hellman 키 교환(Diffie-Hellman key exchange)
  - 1976년에 휘트필드 디피(Whitfield Diffie)와 마틴 헬먼(Martin Hellman)이 발명한 알고리즘
  - 타인에게 알려져도 상관없는 정보를 두 사람이 교환하는 것만으로 공통의 비밀 값을 만들어내는 방법
  - IPsec에서는 Diffie-Hellman 키 교환을 개량한 방법을 사용

## 5.2 Diffie-Hellman 키 교환의 수순

- 1) 앨리스는 밥에게 2개의 소수  $P$ 와 한 원시근  $G$ 를 송신
- 2) 앨리스는 난수  $A$ 를 준비
- 3) 밥은 난수  $B$ 를 준비
- 4) 앨리스는 밥에게  $G^A \bmod P$ 라는 수를 송신
- 5) 밥은 앨리스에게  $G^B \bmod P$ 라는 수를 송신
- 6) 앨리스는 밥이 보낸 수를  $A$ 제곱해서  $\bmod P$ 를 계산
  - 앨리스가 계산한 키 =  $(G^B \bmod P)^A \bmod P = G^{B \times A} \bmod P$
- 7) 밥은 앨리스가 보낸 수를  $B$ 제곱해서  $\bmod P$ 를 계산
  - 밥이 계산한 키 =  $(G^A \bmod P)^B \bmod P = G^{A \times B} \bmod P$

앨리스가 계산한 키 = 밥이 계산한 키

# Diffie-Hellman 키 교환



## 5.3 이브는 키를 계산 할 수 없는 것일까?

- 공격자 이브가 알 수 있는 것
  - $P, G, G^A \bmod P, G^B \bmod P$ 라는 4개의 수
- 이 4개의 수로부터 앨리스와 밥이 공유한 키 ( $G^{A \times B} \bmod P$ )를 계산하는 것은 수학적으로 난해
- 유한체상의 이산대수문제:
  - $G^A \bmod P$ 로부터 수  $A$ 를 효율적으로 계산하는 알고리즘은 아직 없음

# Diffie-Hellman 키교환의 안전성

- 유한체상의 이산대수문제를 풀기 어렵기 때문에 Diffie-Hellman 키 교환의 안전성이 보장
- 단, 소수  $p$ 가 적당히 커야 하고, 양측이 선택하는 수도 랜덤해야 한다



## 5.4 원시근의 의미

- 위수(Order)와 원시근 (Primitive Root)

- 위수(Order)

자연수  $n$ 과  $\gcd(a, n) = 1$  을 만족하는 정수  $a$ 가 임의로 주어 졌을때,  
 $a^k \equiv 1 \pmod{n}$  을 만족하는 가장 작은 자연수  $k$ 를 법  $n$ 에 대한  $a$ 의 **위수 (Order)**라고 정의 → 기호:  $\text{ord}_n(a)$

예) 적당한 정수의 위수를 하나 구해보면  $n = 8$  일 때,  
 $\gcd(3, 8) = 1$  이고  $3^2 \equiv 1 \pmod{8}$ 이므로  $\text{ord}_8(3) = 2$

- 원시근 (Primitive Root)

$n$ 은 자연수이고  $a$ 는  $\gcd(a, n) = 1$ 을 만족하는 정수일 때,  
정수  $a$ 가  $\text{ord}_n(a) = \phi(n)$ 을 만족하면  $n$ 를 법  $n$ 의 **원시근 (Primitive Root)**이라고 정의

## $G^A \bmod P$ 의 표( $P = 13$ 인 경우)

| $G^A$ | 1  | 2  | 3  | 4 | 5  | 6  | 7  | 8 | 9  | 10 | 11 | 12 | 원시근 여부 |
|-------|----|----|----|---|----|----|----|---|----|----|----|----|--------|
| 0     | 0  | 0  | 0  | 0 | 0  | 0  | 0  | 0 | 0  | 0  | 0  | 0  |        |
| 1     | 1  | 1  | 1  | 1 | 1  | 1  | 1  | 1 | 1  | 1  | 1  | 1  |        |
| 2     | 2  | 4  | 8  | 3 | 6  | 12 | 11 | 9 | 5  | 10 | 7  | 1  | 원시근    |
| 3     | 3  | 9  | 1  | 3 | 9  | 1  | 3  | 9 | 1  | 3  | 9  | 1  |        |
| 4     | 4  | 3  | 12 | 9 | 10 | 1  | 4  | 3 | 12 | 9  | 10 | 1  |        |
| 5     | 5  | 12 | 8  | 1 | 5  | 12 | 8  | 1 | 5  | 12 | 8  | 1  |        |
| 6     | 6  | 10 | 8  | 9 | 2  | 12 | 7  | 3 | 5  | 4  | 11 | 1  | 원시근    |
| 7     | 7  | 10 | 5  | 9 | 11 | 12 | 6  | 3 | 8  | 4  | 2  | 1  | 원시근    |
| 8     | 8  | 12 | 5  | 1 | 8  | 12 | 5  | 1 | 8  | 12 | 5  | 1  |        |
| 9     | 9  | 3  | 1  | 9 | 3  | 1  | 9  | 3 | 1  | 9  | 3  | 1  |        |
| 10    | 10 | 9  | 12 | 3 | 4  | 1  | 10 | 9 | 12 | 3  | 4  | 1  |        |
| 11    | 11 | 4  | 5  | 3 | 7  | 12 | 2  | 9 | 8  | 10 | 6  | 1  | 원시근    |
| 12    | 12 | 1  | 12 | 1 | 12 | 1  | 12 | 1 | 12 | 1  | 12 | 1  |        |

# $2^i \bmod 13$ 계산

- 위표에서 G가 2인 부분

$$2^1 \bmod 13 = 2$$

$$2^2 \bmod 13 = 4$$

$$2^3 \bmod 13 = 8$$

$$2^4 \bmod 13 = 3$$

$$2^5 \bmod 13 = 6$$

⋮

$$2^{11} \bmod 13 = 7$$

$$2^{12} \bmod 13 = 1$$

- 따라서

$$\{2^i \bmod 13 \mid i = 1, 2, \dots, 12\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

- 그러므로 2는 13의 원시근
- 6, 7, 11도 원시근임

## 5.5 구체적 키 교환의 예

- 1) 앨리스는 밥에게 2개의 수  $P=13$  과  $G=2$ 를 송신
- 2) 앨리스는 랜덤한 수  $A=9$ 를 준비
- 3) 밥은 랜덤한 수  $B=7$ 을 준비
- 4) 앨리스는 밥에게  $G^A \bmod P = 2^9 \bmod 13 = 5$ 를 송신
- 5) 밥은 앨리스에게  $G^B \bmod P = 2^7 \bmod 13 = 11$ 를 송신

# 키 교환 예

6) **앨리스**는 밥이 보내 온 수 11을 A 제공해서 P로 mod를 계산  
앨리스가 계산한 키 =  $(G^B \bmod P)^A \bmod P$

$$= 11^A \bmod P$$

$$= 11^9 \bmod 13$$

$$= \mathbf{8}$$

7) **밥**은 앨리스가 보내 온 수 5를 B 제공해서 P로 mod를 계산  
밥이 계산한 키 =  $(G^A \bmod P)^B \bmod P$

$$= 5^B \bmod P$$

$$= 5^7 \bmod 13$$

$$= \mathbf{8}$$

## 5.6 타원 곡선 Diffie-Hellman 키 교환

- 타원곡선 Diffie-Hellman 키 교환
  - Diffie-Hellman 키 교환에서는 「이산대수 문제」를 풀기가 매우 어렵다는 사실을 이용해서 키 교환을 실현
  - 「이산대수 문제」를 「타원곡선상의 이산대수 문제」로 대체한 키 교환 알고리즘
  - 디피-헬만 키 교환보다 짧은 키 길이로 높은 보안성을 실현

# 제6절 패스워드를 기초로 한 암호(PBE)

**6.1 패스워드를 기초로 한 암호란 무엇인가?**

**6.2 PBE의 암호화**

**6.3 PBE의 복호화**

**6.4 솔트의 역할**

**6.5 패스워드의 역할**

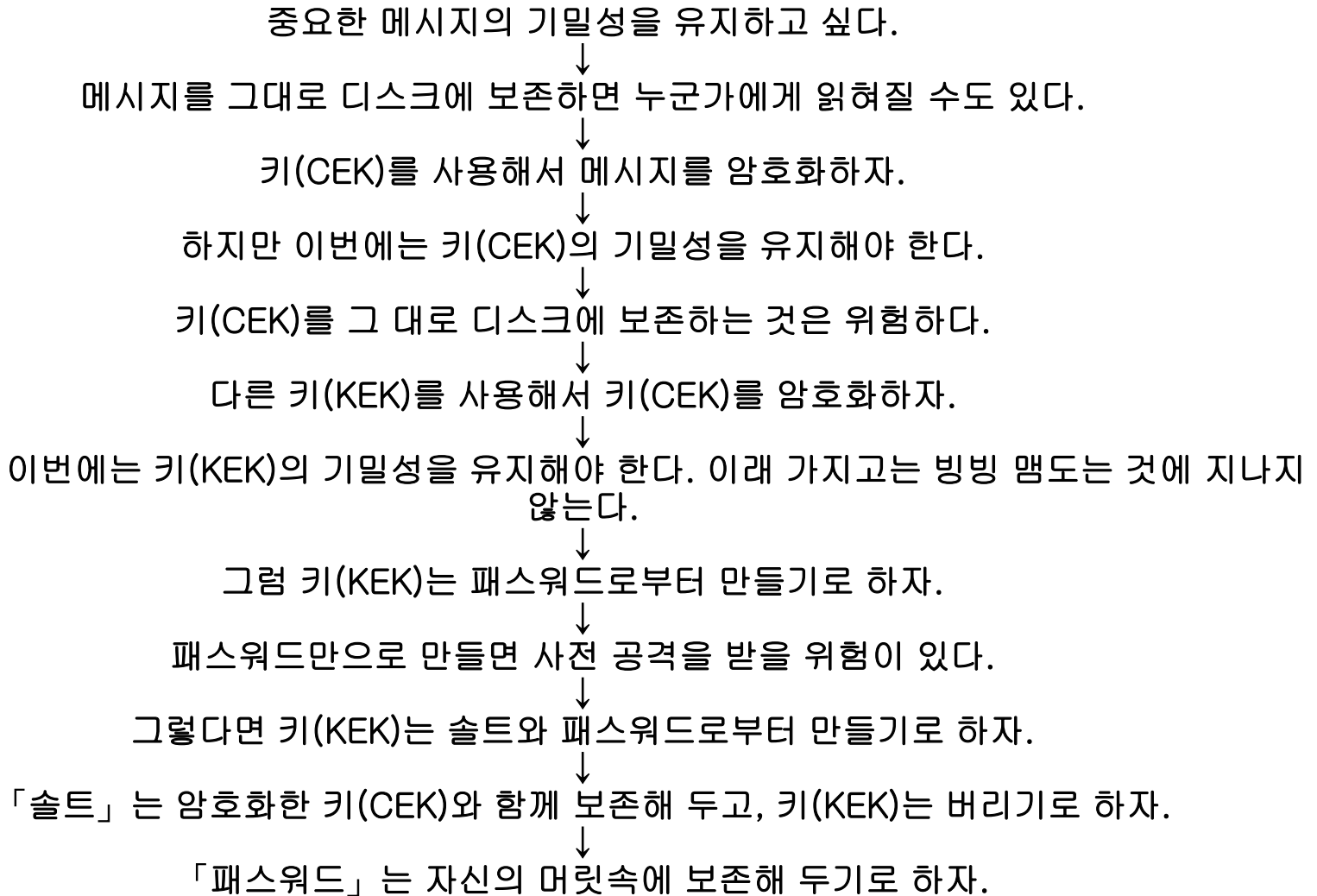
**6.6 스트레칭에 의한 PBE의 개선**

## 6.1 패스워드를 기초로 한 암호란 무엇인가?

- 패스워드를 기초로 한 암호(password based encryption; PBE)
  - 패스워드를 기초로 해서 만든 키로 암호화를 수행하는 방법
  - RSA사의 PKCS #5 규격으로 규정되어 있는 PBE는 Java의 `java.crypto` 패키지 등에 내장
  - 암호 소프트웨어 PGP에서 키를 보존



# PBE 절차



## 6.2 PBE의 암호화

- 1) KEK 생성
- 2) 세션 키 생성과 암호화
- 3) 메시지 암호화

# KEK 생성

- 의사난수 생성기로 솔트(salt)라는 난수를 생성
- 솔트와, 앨리스가 입력한 패스워드를 순서대로 일방향 해수 함수에 입력
- 얻어진 해시 값이 키의 암호화를 위한 키(KEK)

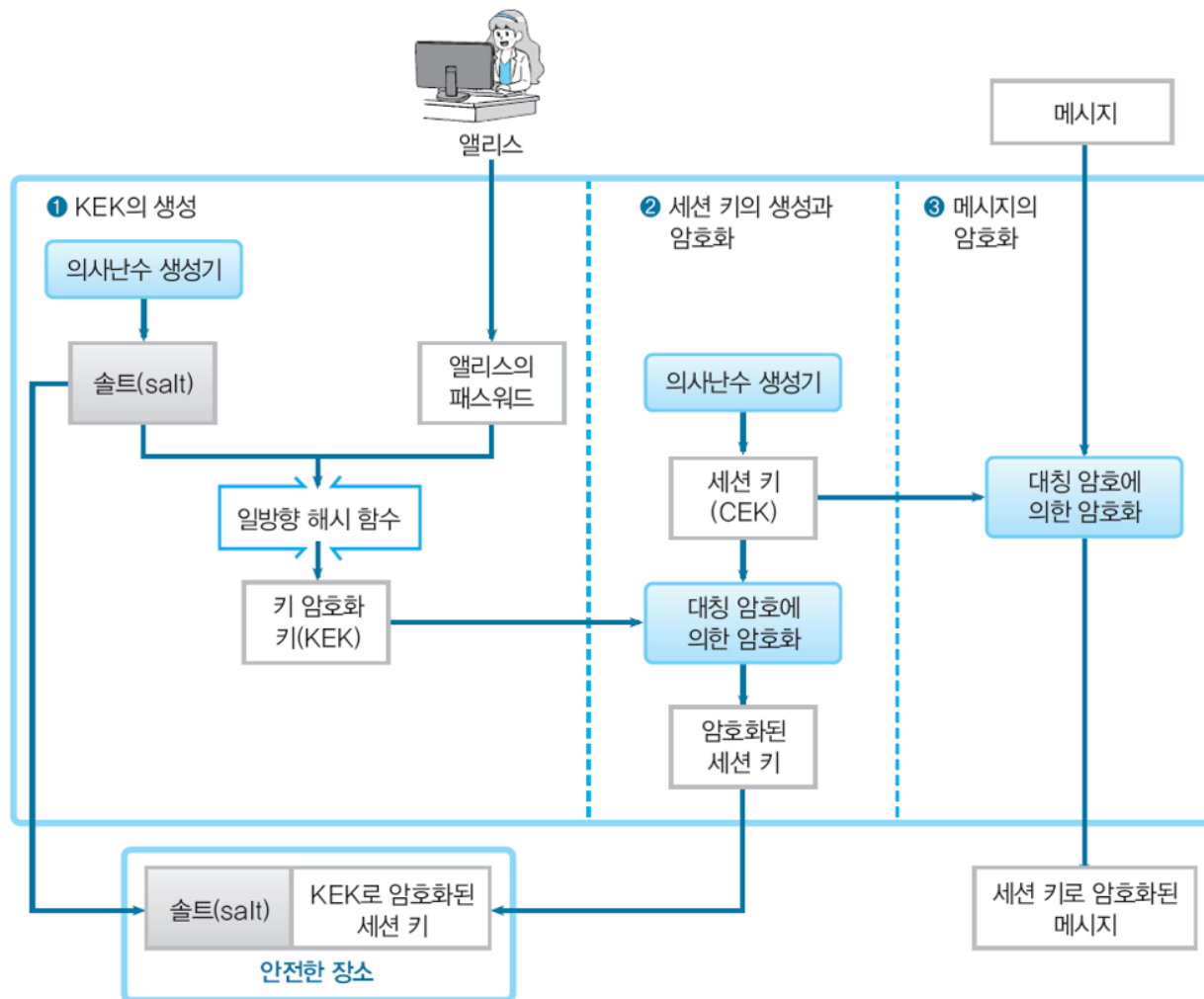
# 세션 키 생성과 암호화

- 의사난수 생성기를 사용해서 세션 키를 생성
- KEK를 사용해서 암호화하고, 솔트와 함께 안전한 장소에 보존
- 세션 키의 암호화가 끝나면 KEK는 폐기
  - 솔트와 비밀번호만 있으면 KEK는 복원 가능

# 메시지 암호화

- 세션 키를 사용해서 메시지를 암호화
- PBE의 암호화에서 하는 것
  - 솔트
  - KEK로 암호화된 세션 키
  - 세션 키로 암호화된 메시지
- 「솔트」와 「KEK로 암호화된 세션 키」는 안전한 장소에 보관

# PBE의 암호화



## 6.3 PBE의 복호화

- 1) KEK 복원
- 2) 세션 키 복호화
- 3) 메시지 복호화

# KEK 복원

- 보존해 둔 솔트와, 앨리스가 입력한 패스워드를 일방향 해시 함수에 순서대로 입력
- 이것은 KEK를 생성했을 때와 같은 계산이므로 얻어진 해시 값은 KEK



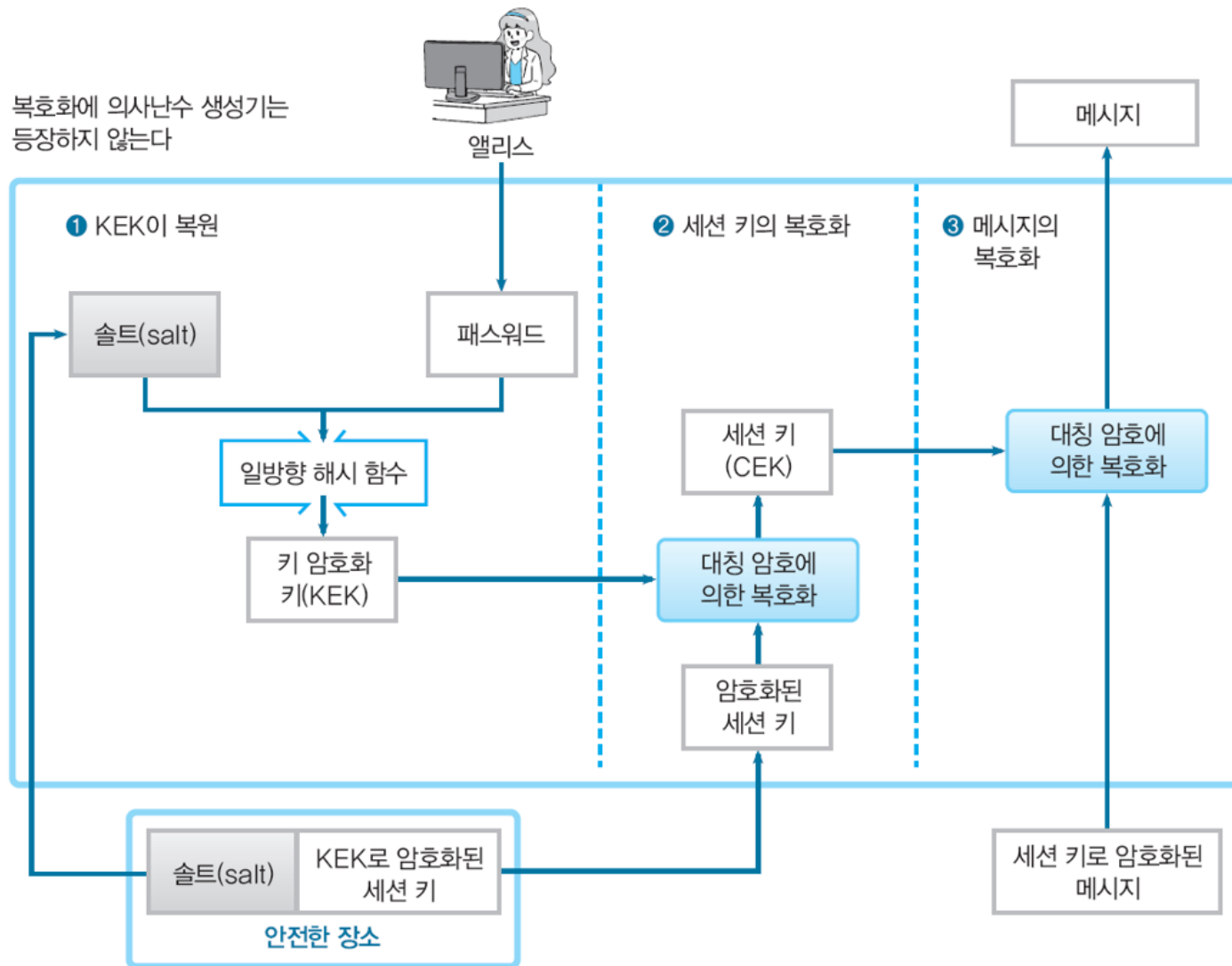
# 세션 키 복호화

- 세션키 구하기
- 보존해 둔 「KEK로 암호화된 세션 키」를 가지고 와서, (1)에서 복원시킨 KEK를 사용해서 복호화

# 메시지 복호화

- 복호화한 세션 키를 사용해서 암호화된 메시지를 복호화

# PBE의 복호화



## 6.4 솔트의 역할

- 의사난수 생성기로 만들어지는 랜덤한 수로 키(KEK)를 만들 때  
에 패스워드와 함께 일방향 해시 함수에 입력
- 사전 공격을 막기 위해 필요

# 솔트 미사용의 경우

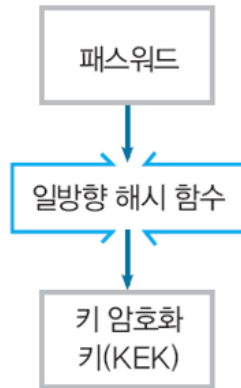
- 적극적 공격자 맬로리는 사전 데이터 등을 기초로 해서 KEK의 후보를 미리 대량으로 만들어 두는 것이 가능
- 암호화된 세션 키를 훔친 다음 복호화를 시도하는데, KEK의 후보를 미리 만들어 둬으로써 시행시간을 대폭 단축할 수가 있다

# 솔트를 사용할 경우

- KEK 후보의 종류 수가 솔트의 비트 길이만큼 늘어나기 때문에, KEK의 후보를 미리 만들어 놓는다는 것이 매우 어렵다
- 솔트가 확보되지 않으면 KEK의 후보 생성 불가
  - 솔트에 의해 KEK의 후보수가 대폭 증가되기 때문

# 사전 공격과 솔트의 역할

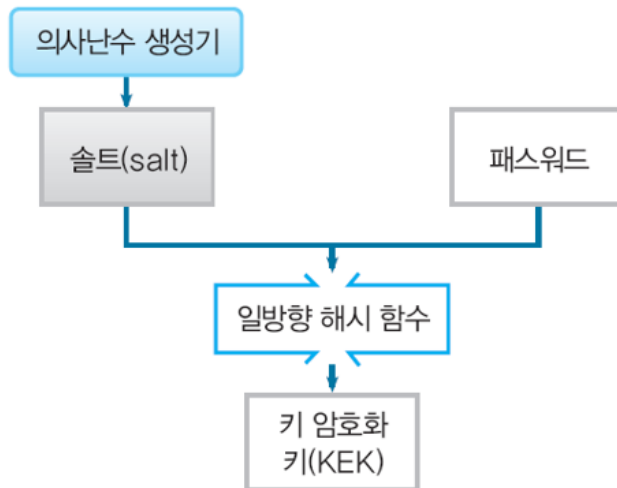
솔트를 사용하지 않은 경우



공격자는 패스워드에 대응하는 KEK 값을 미리 계산할 수 있다  
(사전 공격이 가능하다)

| 패스워드   | 대응하는 KEK의 값       |
|--------|-------------------|
| abc    | 02 E3 29 13 2A D0 |
| abcd   | F5 21 62 FE 72 77 |
| abcxyz | 81 75 8E B2 9F 66 |
| hello  | 3E F3 C7 06 DF B7 |
| pass   | 18 1C 48 22 E6 EF |
| ...    | ...               |

솔트를 사용한 경우



솔트가 다르면 비록 패스워드가 같아도  
KEK 값이 다르므로 사전 공격이 불가능하다

| 솔트     | 패스워드 | 대응하는 KEK 값        |
|--------|------|-------------------|
| 5B94E7 | abc  | 4D 85 FD 69 87 38 |
| E5AB9D | abc  | EB 4D CB A9 C3 A4 |
| F8DC3B | abc  | 09 70 F0 7D AC 20 |
| C6541B | abc  | 44 40 32 6F AB 16 |
| F6C109 | abc  | 1F C5 3C 14 DF D8 |
| ...    | ...  | ...               |

## 6.5 패스워드의 역할

- 충분한 비트 수를 갖는 패스워드를 기억할 수는 없다
- PBE에서는 패스워드로 만든 키(KEK)로 세션 키(CEK)를 암호화
- 패스워드로 만든 키(KEK)는 의사난수 생성기로 만든 세션 키(CEK)보다도 약하다
- 말하자면 튼튼한 금고의 키를 약한 금고에 보관하고 있는 것과 같은 것
- PBE를 이용하려면 솔트와 암호화한 CEK를 물리적으로 지키는 방법을 병용해야 함
  - 예: CEK를 항상 휴대하고 있는 IC 카드에 보관



## 6.6 스트레칭에 의한 PBE의 개선

- 스트레칭(stretching)
  - 복잡한 연산을 반복적으로 적용
    - ➔ 일방향 해시함수 적용 회수를 증가하는 방법
  - KEK를 만들 때 일방향 해시 함수를 여러 번 통과하도록 하면 안전
  - 사용자 입장에서 해시 함수를 1000회 반복하는 것은 용이
  - 공격자 맬로리에게는 작은 차이가 큰 부담
    - . 바른 KEK를 찾을 때까지 대량의 패스워드를 시도해야만 함

## **제7절 안전한 패스워드를 만들려면**

**7.1 자신만이 알 수 있는 정보를 사용할 것**

**7.2 복수 패스워드를 사용할 것**

**7.3 메모를 유효하게 사용할 것**

**7.4 패스워드의 한계**

**7.4 패스워드 생성/관리 툴을 사용할 것**

# 안전한 비밀번호 만들기

- 자신만이 알 수 있는 정보를 사용할 것
- 복수의 비밀번호를 나누어 쓸 것
- 메모를 유효하게 사용할 것
- 비밀번호의 한계를 알 것

## 7.1 자신만이 알 수 있는 정보를 사용할 것

- 중요한 것의 이름을 사용해서는 안 된다
  - 배우자 이름, 애인 이름, 아이 이름, 애완동물 이름, 유명인 이름, 자동차 이름, 브랜드명 등
- 자신에 관한 정보를 사용해서는 안 된다
  - 자신의 이름, 자신의 로그인 명, 주소, 사원 번호 등
- 타인이 보기 쉬운 정보를 사용해서는 안 된다
  - 명언, 유명한 인용구, 사전의 예문, 웹에서 찾은 말, 키보드의 배열을 이용한 문자열(qwert, asdfghjkl 등), 무지개 색, 혹성의 이름, 성좌, 달 이름, 요일 이름 등

## 7.2 복수 패스워드를 사용할 것

- 하나의 패스워드를 다양한 용도에 사용해서는 안됨
- 정보의 가치에 따라 패스워드를 구별해서 사용
- 패스워드의 일부만을 바꾸어 복수의 패스워드로 나누어 사용해서는 안됨

– 예:

- 회사 컴퓨터의 로그인용: tUniJw1
- 집 컴퓨터의 로그인용 : tUniJw2
- 메일의 디지털 서명용 : tUniJw3
- 온라인 쇼핑용 : tUniJw4

## 7.3 메모를 유효하게 사용할 것

- 패스워드를 메모에 써 넣고, 컴퓨터 모니터에 붙여 놓아서는 안 됨
- 메모를 유용하게 사용하는 것은 결코 나쁘지 않다
- 메모를 물리적인 키와 동일하게 취급
- 패스워드의 일부분만을 메모해 두는 것은 특히 유효

## 7.4 패스워드의 한계

- 가정: 영어 알파벳과 숫자열중의 8문자로 한정
- 62개 문자  
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
- 영어 알파벳과 숫자 8문자로 된 문자열의 가능성
$$62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62$$
$$= 62^8$$
$$= 218340105584896$$
- 약 218조 종류

# 패스워드의 한계

- 키의 비트 수로 말하면 48비트 정도에 지나지 않음
- 이 정도의 길이는 전사 공격이 가능한 길이
- 만약 적극적 공격자의 컴퓨터가 1초간에 1억 개의 패스워드를 만들어서 시험할 수 있다면, 약 25일에 모든 패스워드를 체크할 수 있음



## 7.5 패스워드 생성/관리 툴을 사용

- 수많은 웹 사이트를 이용
  - 다수의 ID와 패스워드 필요
- 패스워드생성/관리 툴을 이용하는 것이 바람직
  - 난수를 사용해서 추측이 어려운 패스워드를 생성
  - 브라우저와 연계해서 웹 사이트의 패스워드 입력을 지원
  - 툴이 사용자의 패스워드를 마음대로 이용하지 않도록 주의
  - 툴과 그 개발처를 「신뢰」 할 수 있는지가 중요

# 12장 키

연습문제 풀이

# 연습문제

1. 어느 암호 시스템에서 가능한 모든 키들을 모아놓은 공간을 \_\_\_\_\_이라고 한다.

- ① Key Space
- ② Key Warehouse
- ③ Key Set
- ④ Key Family
- ⑤ Key Collection

2. 대칭 암호 시스템인 DES, DES-EDE2, DES-EDE3 의 실제 키 길이를 순서대로 알맞게 쓴 것은?

- ① 56, 112, 168
- ② 56, 64, 128
- ③ 64, 128, 192
- ④ 64, 112, 192
- ⑤ 56, 128, 168

## 3. 키와 평문에 대한 설명으로 적합한 것은?

- ① 키는 길이가 짧기 때문에 평문보다 그 가치가 적다.
- ② 동일한 평문에 키를 적용할 때 키의 길이만 같으면 동일한 암호문을 생성한다.
- ③ 키의 가치와 평문의 가치는 동일하다고 할 수 있다.
- ④ 평문의 길이보다 키의 길이는 항상 짧아야 한다.
- ⑤ 평문의 길이와 키를 사용하여 생성한 암호문은 길이가 동일하다.

## 4. 메시지 인증에 사용하는 키와 디지털 서명에 사용하는 키의 차이점에 대해 올바른 설명은?

- ① 메시지 인증에 사용하는 키는 송신자와 수신자가 동일한 키를 사용한다.
- ② 메시지 인증에는 공개 키 방법을 사용한다.
- ③ 메시지 인증에는 키 교환에 대해 신경 쓸 필요가 없다.
- ④ 디지털 서명을 검증할 수 있는 사람은 서명자의 개인 키를 알고 있는 사람 뿐이다.
- ⑤ 디지털 서명에서는 송신자와 수신자가 동일한 공개 키를 사용한다.

5. 메시지 인증 코드나 디지털 서명에 사용하는 키의 목적에 대한 올바른 설명은?

- ① 기밀성을 유지하기 위한 키이다.
- ② 인증을 수행하기 위한 키이다.
- ③ 대칭 키 교환을 위한 키이다.
- ④ 송신자와 수신자 모두 동일한 키를 사용하는 방법이다.
- ⑤ 바른 키를 모르고 있어도 목적을 달성할 수 있는 방법이다.

6. 통신 때마다 새로운 키를 사용하는 경우에 이 키를 \_\_\_\_\_라고 하고, 반복적으로 사용되는 키를 \_\_\_\_\_라고 한다.

- ① 일회용 키, 반복 키
- ② 세션 키, 마스터 키
- ③ 일회용 키, 마스터 키
- ④ 세션 키, 공통 키
- ⑤ 단순 키, 공통 키



7. 통상적으로 사용자가 직접 이용하는 정보가 암호화의 대상이 된다. 이 때에 사용하는 키를 \_\_\_\_\_라 부른다. 이것과는 달리 「키를 암호화하는 키」를 \_\_\_\_\_라 부른다.

- ① KEK, IEK
- ② IEK, KEK
- ③ CEK, IEK
- ④ CEK, KEK
- ⑤ KEK, CEK

8. 키를 생성할 때 키가 갖추어야 할 가장 중요한 성질은?

- ① 반복성
- ② 규칙성
- ③ 무작위성
- ④ 주기성
- ⑤ 예측성

9. 패스워드로부터 키를 만들 때에는 사전 공격이라는 공격을 막기 위해 패스워드에 \_\_\_\_\_(이)라 불리는 난수를 부가해서 일방향 해시 함수에 입력한다. 이 방법을 \_\_\_\_\_(이)라 부른다.

- ① 솔트, KEK
- ② 패딩, CEK
- ③ 솔트, PKI
- ④ 패딩, PBE
- ⑤ 솔트, PBE

10. 키 교환 알고리즘 중의 하나로서 타인에게 알려져도 상관없는 정보를 두 사람이 교환하는 것만으로 공통의 비밀 값을 만들어내어 대칭 키로 사용하는 방법은 다음 중 어느 것인가?

- ① AES 키 교환
- ② Rabin 키 교환
- ③ PKI 키 교환
- ④ Diffie-Hellman 키 교환
- ⑤ RSA 키 교환

11. 유한체 상의 이산대수문제를 수학적으로 풀기 어렵다는 것이 \_\_\_\_\_(를)을 이론적으로 뒷받침한다.

- ① PBE
- ② PKI
- ③ RSA 알고리즘
- ④ Diffie-Hellman 키 교환
- ⑤ 의사난수 생성

12. 패스워드에 솔트를 추가하게 되면 생성하는 KEK의 값을 변화시킬 수 있는데 이 솔트를 사용하게 되면 \_\_\_\_\_이라는 공격을 막을 수 있다.

- ① 중간자 공격
- ② 재생 공격
- ③ 전사 공격
- ④ 사전 공격
- ⑤ 생일 공격

13. PBE에서 KEK를 만들 때 안전성을 높이기 위해 어떠한 방법을 사용하는가?

- ① 암호화를 중복해서 수행한다.
- ② 키의 길이를 길게 한다.
- ③ 길이가 긴 솔트를 추가한다.
- ④ 길이가 긴 패딩을 추가한다.
- ⑤ 해시 함수를 여러 번 통과하도록 한다.

## 14. 안전한 패스워드를 유지하기 위한 방법으로 옳바르지 않은 것은?

- ① 자신만이 알 수 있는 정보를 사용한다.
- ② 여러 개의 패스워드를 사용한다.
- ③ 메모를 유효하게 사용한다.
- ④ 자신과 관련된 것은 사용하지 않는다.
- ⑤ 잘 알려지지 않은 혹성의 이름 같은 것을 사용한다.



# 연습문제

Diffie-Hellmant 키 교환의 예를 하나 들어보기로 하자. 이때  $P=7$ 을 선택하고  $G=5$ 를 선택하였다고 하자. 앨리스가 선택한 랜덤한 수는  $A=4$ 라고 하고 밥이 선택한 랜덤한 수는  $B=3$ 이라고 한다. (문제 15 ~ 19)

15. 앨리스가 계산해야 하는 값  $G^A \bmod 7$ 은 무엇인가?

$$G^A \bmod 7 = 5^4 \bmod 7 = 2$$

# 연습문제

Diffie-Hellmant 키 교환의 예를 하나 들어보기로 하자. 이때  $P=7$ 을 선택하고  $G=5$ 를 선택하였다고 하자. 앨리스가 선택한 랜덤한 수는  $A=4$ 라고 하고 밥이 선택한 랜덤한 수는  $B=3$ 이라고 한다. (문제 15 ~ 19)

16. 밥이 계산해야 하는 값  $G^B \bmod 7$ 은 무엇인가?

$$G^B \bmod 7 = 5^3 \bmod 7 = 6$$

# 연습문제

Diffie-Hellmant 키 교환의 예를 하나 들어보기로 하자. 이때  $P=7$ 을 선택하고  $G=5$ 를 선택하였다고 하자. 앨리스가 선택한 랜덤한 수는  $A=4$ 라고 하고 밥이 선택한 랜덤한 수는  $B=3$ 이라고 한다. (문제 15 ~ 19)

17. 앨리스는 밥이 보내준 값인  $G^B \bmod 7$ 를  $A=4$  제공해서  $\bmod 7$ 을 취한다. 그 결과는 무엇인가?

$$6^4 \bmod 7 = 1$$

# 연습문제

Diffie-Hellmant 키 교환의 예를 하나 들어보기로 하자. 이때  $P=7$ 을 선택하고  $G=5$ 를 선택하였다고 하자. 앨리스가 선택한 랜덤한 수는  $A=4$ 라고 하고 밥이 선택한 랜덤한 수는  $B=3$ 이라고 한다. (문제 15 ~ 19)

18. 밥은 앨리스가 보내준 값인  $G^A \bmod 7$ 를  $B=3$  제공해서  $\bmod 7$ 을 취한다. 그 결과는 무엇인가?

$$2^3 \bmod 7 = 1$$

# 연습문제

Diffie-Hellmant 키 교환의 예를 하나 들어보기로 하자. 이때  $P=7$ 을 선택하고  $G=5$ 를 선택하였다고 하자. 앨리스가 선택한 랜덤한 수는  $A=4$ 라고 하고 밥이 선택한 랜덤한 수는  $B=3$ 이라고 한다. (문제 15 ~ 20)

19. 문제 19와 문제 20에서 구한 값이 같은지 아닌지 확인 해보시오.

**동일하다.**

- 암호학과 네트워크 보안, Behrouz A. Forouzan 지음, 이재광외 3인 역, 한티미디어
- 컴퓨터 보안과 암호, WILLIAM STALLINGS 지음, 최용락외 2인 역, 그린출판사

**Q & A**

**Thanks!**