정보보호

대칭키 암호

정보보호 연구실

이동섭

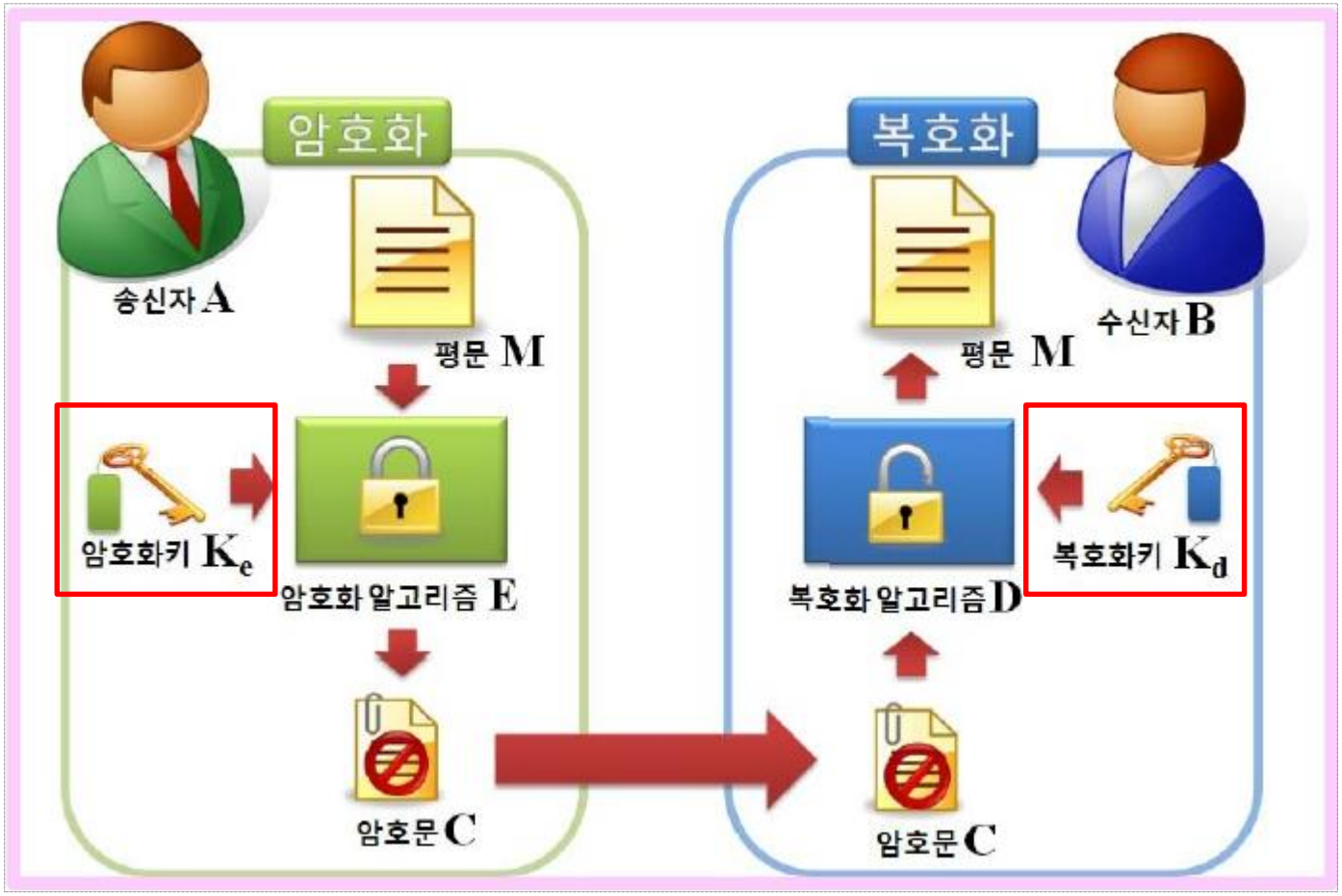
ISLAB
INFORMATION SECURITY LABORATORY CNU

실습 계획

주차	날짜	실습 내용	과제	이론
1	09.02	Introduce	-	암호 소개
2	09.09	기초 암호기술 및 암호문 해독	시저, 비즈네르 암호문 해독	고전 암호 / 블록 암호
3	09.18	블록 암호 실습	메시지 AES 암호화	블록 암호
4	09.25	암호화 채팅프로그램 구현	채팅 프로그램 구현	공개키 암호
5	10.02	공개키 암호	RSA 실습	
6	10.09	한글날	-	전자서명
7	10.16	전자서명	전자서명 실습	
8	10.23	중간고사 (예정)	-	
9	10.30	해시	Sha 256 실습	해시
10	11.06	하이브리드 암호 시스템 구현	암호화 채팅 프로그램 구현	하이브리드 암호 시스템
11	11.13	블록체인 기초	환경 세팅	블록체인
12	11.20	블록체인 지갑주소 생성	지갑주소 생성	
13	11.27	블록체인 거래	블록체인 거래	SSL
14	12.04	블록 해더 확인	블록 해더 확인	
15	12.11	기말고사 (예정)	-	

암호

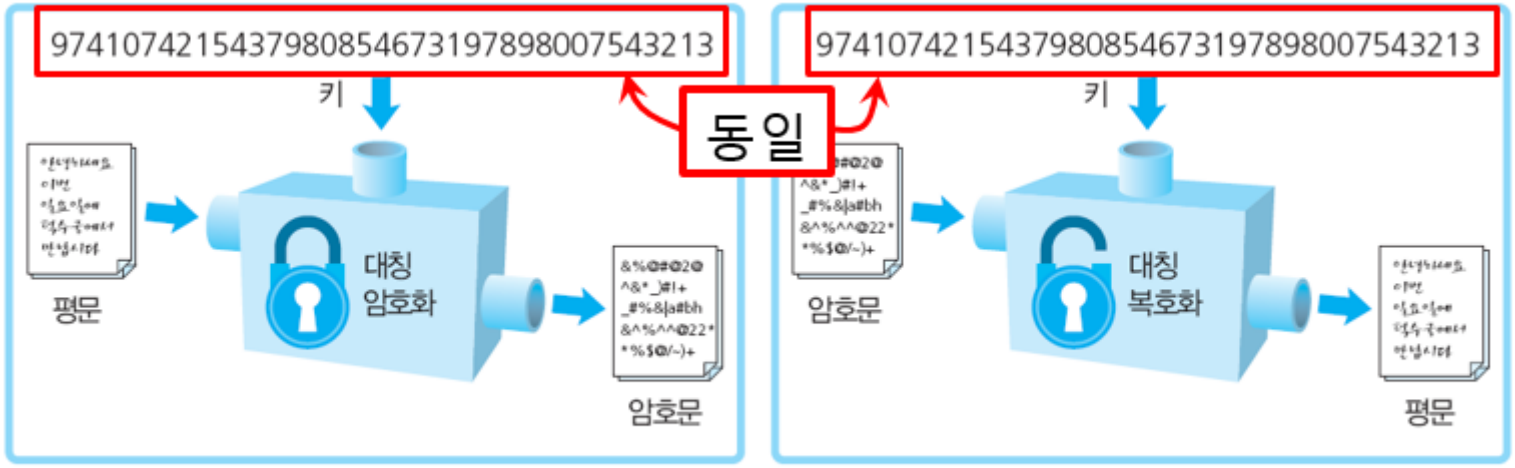
- 암호화 키 = 복호화키 ?


암호화
송신자 A
평문 M
암호화키 K_e
암호화 알고리즘 E
암호문 C
복호화
수신자 B
평문 M
복호화키 K_d
복호화 알고리즘 D
암호문 C

대칭키 암호

■ 암호화 키 = 복호화 키

- ▶ 대칭키 암호(Symmetric-key cryptosystem)
- ▶ = 관용 암호
- ▶ = 비밀키 암호(Secret-key cryptosystem)



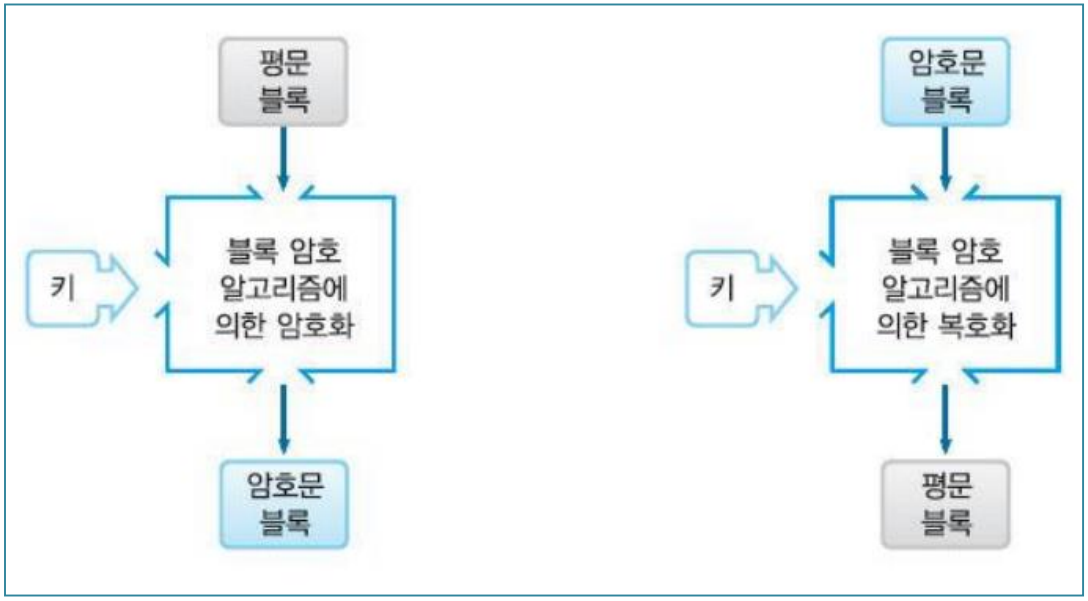
(a) 대칭 암호: 암호화와 복호화에 동일한 키를 사용

- ● 블록 암호 (Block Cipher)
- ● 스트림 암호 (Stream Cipher)

대칭키 암호

■ 블록 암호 (Block Cipher)

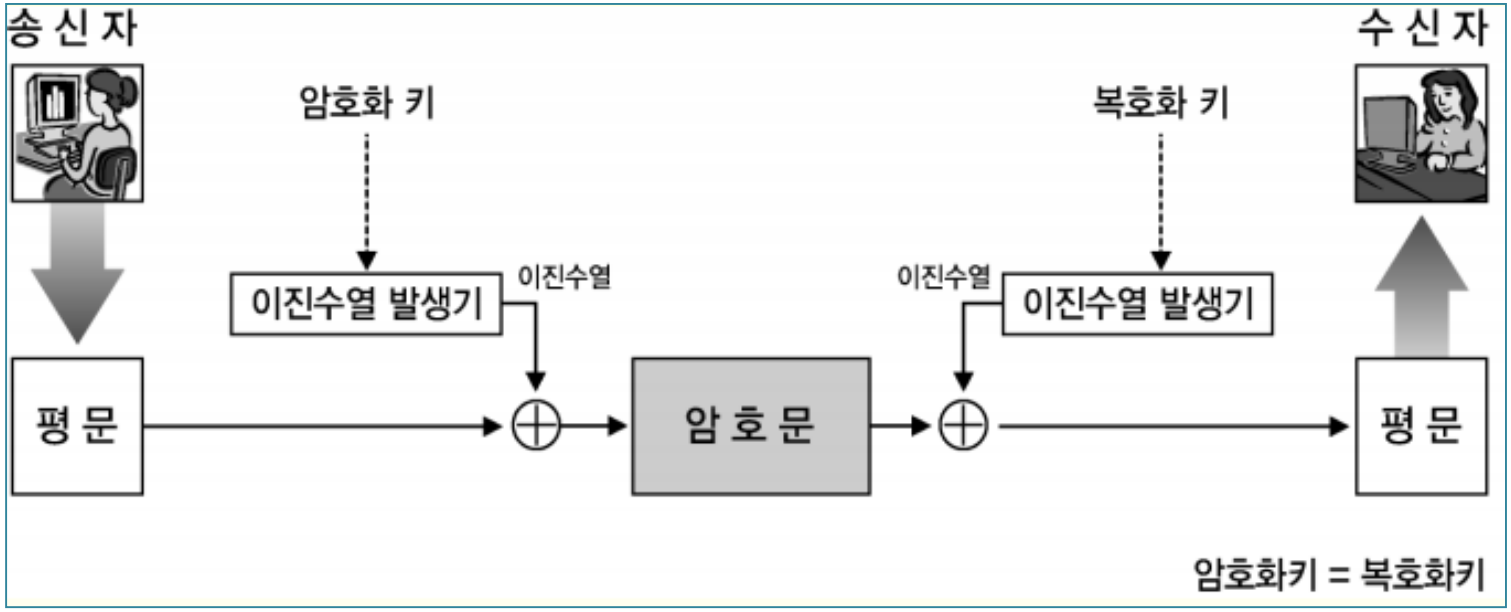
- ▶ 어느 특정 비트 수의 ‘집합’을 한 번에 처리하는 암호 알고리즘
  - ● 집합을 블록(Block)이라고 함
  - ● 블록의 비트 수를 블록 길이 (Block length)라고 함
    - ▪ DES나 트리플 DES의 블록 길이는 64비트
    - ▪ DES : 64비트 평문, 64비트 암호문
    - ▪ AES : 블록 길이는 128비트, 192비트, 256비트



대칭키 암호

■ 스트림 암호 (Stream Cipher)

- 데이터의 흐름(스트림)을 순차적으로 처리해가는 암호 알고리즘
- 평문 스트림과 이진 키 스트림의 XOR 연산으로 암호문 생성
- 블록 암호보다도 빠른 특성이 있음
  - RC4, A5/1 등



XOR

■ XOR 특징

- 같은 값끼리 XOR을 취할 경우 0이 됨
- AND 연산이나 OR 연산과는 다른 특징
- ‘A’ XOR ‘C’ 를 하는 경우

```
          A => 0x41 => 0 1 0 0   0 0 0 1
XOR       C => 0x43 => 0 1 0 0   0 0 1 1
 ---------------------------------------
                       0 0 0 0   0 0 1 0
```

- ‘A’ XOR ‘C’ 결과에 다시 XOR ‘C’ 를 하는 경우

```
                       0 0 0 0   0 0 1 0
XOR       C => 0x43 => 0 1 0 0   0 0 1 1
 ---------------------------------------
          A => 0x41 => 0 1 0 0   0 0 0 1
```

INPUT		OUTPUT
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

간단한 XOR 암복호화 구현

■ 간단한 XOR 암복호화 구현

- 평문 Hello Security

```
Hello Security
```

- 키 ‘sky’를 이용해 암호화를 진행

```
H e l l o   S e c u r i t y
s k y s k y s k y s k y s k
---------------------------
? ? ? ? ? ? ? ? ? ? ? ? ? ?
```

- 복호화는 XOR 연산을 한번 더 진행하여 평문 획득

간단한 XOR 암복호화 구현

■ 간단한 XOR 암호화 구현

- ▶ Input : 3의 배수의 글자
- ▶ Key : 3글자

```
(base) dajaba@dajaba:~/lab/02$ cat test
HelloSecurity!!
(base) dajaba@dajaba:~/lab/02$ python xor.py
Enter either "encrypt" or "e" or "decrypt" or "d".
e
Enter the key
key
Enter your file name:
test
En(De)cryption complete
(base) dajaba@dajaba:~/lab/02$ python xor.py
Enter either "encrypt" or "e" or "decrypt" or "d".
d
Enter the key
key
Enter your file name:
encrypt.txt
En(De)cryption complete
(base) dajaba@dajaba:~/lab/02$ cat decrypt.txt
HelloSecurisy!!(base) dajaba@dajaba:~/lab/02$
```

간단한 XOR 암복호화 구현

■ 간단한 XOR 암복호화 구현

```
mode = getMode()
key = getKey()
fileName = getFileName()
encrypt(mode, fileName, key)
```

```
def getMode():
    while True:
        print('Enter either "encrypt" or "e" or "decrypt" or "d".')
        mode = input().lower()
        if mode in 'encrypt e decrypt d'.split():
            return mode
        else:
            print('The value you entered its invalid')

def getFileName():
    print('Enter your file name:')
    return input()

def getKey():
    key = 0
    while True:
        print('Enter the key')
        key = input()
        return key
```

간단한 XOR 암복호화 구현

■ 간단한 XOR 암복호화 구현

```
def encrypt(mode, fileName, key):
    keyValue = ''

    outputFileName = 'encrypt.txt'
    if mode[0] == 'd':
        outputFileName = 'decrypt.txt'

    translated = ''
    outputFile = open(outputFileName, 'w')

    inputFile = open(fileName, 'r')
    message = inputFile.read()

    for i in range(len(message)//3):
        keyValue += key

    translated = str_xor(keyValue, message)
    outputFile.write(translated)
    outputFile.close()
    inputFile.close()
    print('En(De)cryption complete')
```

Brute-Force Attack

■ Brute-Force Attack 구현

- ▶ 입력 : 무작위 답
  - ● ‘aaa’ 부터 ‘zzz’ 까지 모두 대입

```
(base) dajaba@dajaba:~/lab/02$ python bruteForceAttac.py
Enter either "encrypt" or "e" or "decrypt" or "d".
d
Enter your file name:
encrypt.txt
En(De)cryption complete
```

```
HetloKecmriky!9
HewloHecnrihy!:
HevloIecoriiy!;
HeqloNechriny!<
HeploOecirioy!=
HesloLecjrily!>
HerloMeckrimy!?
He}loBecdriby!0
He|loCecericy!1
He^?lo@ecfri`y!2
He~loAecgriay!3
HeyloFec`rify!4
HexloGecarigy!5
He{loDecbridy!6
HezloEeccriey!7
HeeloZec|rizy!(
Hedlo[ec}ri{y!)
HegloXec~rixy!*
HefloYec^?riyy!+
Healo^ecxri~y!,
He`lo_ecyri^?y!-
Heclo\eczri|y!.
Heblo]ec{ri}y!/
HemloRectriry!
HelloSecurisy!!
HeoloPecvripy!"
HftllKe`mrjky"9
HfwllHe`nrjhy":
/Hello
```

제출 요령

보고서 (*.pdf)

- 문서는 PDF로 변환하여 제출
- 과제 해결 과정
  - 과제를 어떻게 이해했는지
  - 어떻게 해결했는지

소스코드 (*.py)

- 과제 해결에 작성한 코드

❖ 소스코드

1) 실습 부분 코드

- XOR 암복호화 코드

2) 과제 부분 코드

- Brute-Force Attack 코드

제출 요령

■ 제출 방법

- ▶ 사이버 캠퍼스 (e-learn.cnu.ac.kr)
- ▶ 파일명 [IS00]_03_학번_이름
- ▶ 강의 〉 과제제출
  - ● 사이버 캠퍼스로 제출한 과제만 인정

■ 제출 기한

- ▶ 2019 09. 18 10:00:00 ~ 2019 09. 24 23:59:59