

운영체제 보안 2

- 암호화 프로그래밍



컴퓨터소프트웨어학과 김병국 교수

목차



- □암호화/복호화 개발 환경 구축
- □RC4 암호화/복호화
- □AES 암호화/복호화

1. 암호화/복호화 개발 환경 구축



□개발 환경 구축

- 라이브러리 설치
 - 명령어: sudo apt-get install libssl-dev
 - OpenSSL관련 기능을 프로그래밍할 수 있는 API들을 제공
 - 최신화 관련하여 오류 메시<u>지가 나타날 경우, 아래의 "최</u>신화"를 실행 후 이 부분을 재시도
- 설치 확인
 - 명령어 : sudo apt-get list libssI-dev
- 칼리 리눅스 최신화(옵션)
 - 아래 1과 2를 순차적으로 실행
 - 1. 명령어 : sudo apt-get update
 - 설치된 패키지들에 대하여 최신 상태를 확인
 - 2. 명령어 : sudo apt-get upgrade
 - 최신 상태로 변경(소요시간 : 수 십분

17

2. RC4 암호화/복호화



□RC4 기술

- RC(Ron's Code 혹은 Rivest's Cypher의 약자)
- 1987년 RSA 시큐리티의 로널드 라이베스(Ronald Lorin Rivest)가 설계
- 바이트(옥텟) 스트림 단위 처리
- 빠른 처리
- 바이트단위 동일 패턴이기 때문에, 하나가 뚫리면 큰일

3. RC4 관련 함수 (1/2)



□ RC4 함수

- 함수 : RC4_set_key()
 - 암호화/복호화를 위한 키(128bits = 16bytes)를 생성
 - RC4를 위한 키는 rc4_key_st 구조체 형태를 가짐
 - 해당 함수의 사용을 위해서는 openssl/rc4.h 헤더를 추가해야 함
 - 예: #include <openssl/rc4.h>

• 인자:

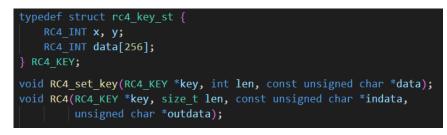
- *key : 암호화/복호화용 키

- len : 키(*data) 값의 길이

- *data : 키 값

🌞 💍 빌드 옵션

- gcc를 이용한 빌드 시 옵션에 관련 라이브러리 추가 필요
- 옵션: -lcrypto (-l(소문자 "L")은 라이브러리 추가 옵션임)



[RC4관련 함수들]



3. RC4 관련 함수 (2/2)



☐RC4 함수

- 함수 : RC4 ()
 - RC4 방식의 암호화/복호화기능을 수행
 - 인자
 - *key : 암호화/복호화를 위한 키
 - len : 입력데이터(*indata)의 길이
 - *indata : 입력데이터(평문 또는 암호문)
 - *outdata : 출력데이터(암호문 또는 평문)

🌞 빌드 옵션

- gcc를 이용한 빌드 시 옵션에 관련 라이브러리 추가 필요
- 옵션: -lcrypto (-l(소문자 "L")은 라이브러리 추가 옵션임)



[RC4관련 함수들]



4. RC4 응용 프로그래밍

printf("KEY=");

printf("\n");

for (int i = 0; i < 16; i++) {

printf("%02X", p cKey[i]);

nFd In = open(argv[1], O RDONLY);

nFd Out = open(argv[2], O WRONLY | O CREAT, 0644);

```
#include <sys/types.h>
□RC4 실습
                                                #include <sys/stat.h>
                                                #include <fcntl.h>
     int main(int argc, char *argv[])
                                                #include <openssl/rc4.h>
 11
                                                #include <string.h>
         int nFd In = -1;
                                                #include <unistd.h>
         int nFd Out = -1;
                                                #include <stdbool.h>
         unsigned char p cKey[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
         int nDataLen:
         unsigned char p cInputData[BUFSIZ] = {0};
         unsigned char p cOutputData[BUFSIZ] = {0};
         RC4 KEY stRc4Key;
         if (argc != 3) {
             printf("Usage: cmd <in-file> <out-file>\n");
             return -1;
```

```
3 printf("set key.\n");
    RC4 set key(&stRc4Key, 16, p cKey);
    printf("Start to encrypt.\n");
    while (1) {
        nDataLen = read(nFd In, p cInputData, BUFSIZ);
        if (nDataLen <= 0)
            break;
        RC4(&stRc4Key, nDataLen, p cInputData, p cOutputData);
        if (write(nFd Out, p cOutputData, nDataLen) <= 0)</pre>
            break;
    close(nFd In);
    close(nFd Out);
    printf("En/De-cryption is done.\n");
    return 0;
                                          [파일명: rc4.c]
```

•컴파일: qcc rc4.c -lcrypto -o rc4

#include <stdio.h>

•실행 예: ./rc4 test1.txt test2.txt.enc

5. AES 암호화/복호화



□AES 기술

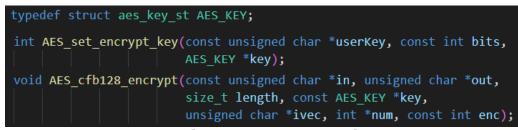
- AES (Advanced Encryption Standard) 암호화 알고리즘
- NIST(National Institute of Standards and Technology: 미 표준 기술 연구소) 에서 개발
- 대칭형 암호화 알고리즘
- 암호화를 위해 128, 192, 256 비트의 키를 지원
- 현재 가장 많이 사용되고 있는 암호화 알고리즘

6. AES 암호화/복호화 함수 (1/2)



☐ AES 함수

- 함수: AES_set_encrypt_key()
 - 암호화/복호화를 위한 키(128bits = 16bytes)를 생성
 - AES를 위한 키는 aes_key_st 구조체(명칭: AES_KEY) 형태를 가짐
 - 해당 함수의 사용을 위해서는 openssl/aes.h 헤더를 추가해야 함
 - 예: #include <openssl/aes.h>
 - 인자:
 - *userKey : 사용자 지정키(비밀번호)
 - bits : 암호화 비트
 - *key : 생성될 AES 키
 - 결과 값:
 - 성공: 0. 실패: -1 또는 -2



[AES 128 관련 함수들]



- qcc를 이용한 빌드 시 옵션에 관련 라이브러리 추가 필요
- 옵션: -lcrypto (-l(소문자 "L")은 라이브러리 추가 옵션임)

6. AES 암호화/복호화 함수 (2/2)



☐ AES 함수

- 함수: AES_cfb128_encrypt()
 - AES 128의 암호화/복호화 기능을 수행
 - 블록 암호화 모드로 CFB(Cipher-FeedBack)를 활용

• 인자:

- *in : 입력데이터

- *out : 출력데이터

- length : 입력데이터(in)의 길이

- *kev : AES 키

- *ivec : 벡터값

- *num : 블록의 위치

- enc : 암호화(1)/복호화(0)

```
typedef struct aes key st AES KEY;
int AES set encrypt key(const unsigned char *userKey, const int bits,
                        AES KEY *key);
void AES cfb128 encrypt(const unsigned char *in, unsigned char *out,
                        size_t length, const AES_KEY *key,
                        unsigned char *ivec, int *num, const int enc);
```

[AES 128 관련 함수들]



- qcc를 이용한 빌드 시 옵션에 관련 라이브러리 추가 필요
- 옵션: -lcrypto (-l(소문자 "L")은 라이브러리 추가 옵션임)

```
if (argc != 4) {
7. AES 응용 (1/2)
                                                              printf("Usage: %s <-e/-d> <in-file> <out-file>\n", argv[0]);
                                                              return -1;
                                                   29
□AES 기술 실습
                                                          while((nOpt=getopt(argc, argv, "ed"))>0) {
                                                              switch (nOpt) {
        #include <stdio.h>
                                                                  case 'e':
        #include <sys/types.h>
                                                                     bEncryption = true;
        #include <sys/stat.h>
                                                                     break:
        #include <fcntl.h>
                                                                  case 'd':
        #include <openssl/aes.h>
                                                                     bEncryption = false;
        #include <string.h>
                                                                     break;
        #include <getopt.h>
                                                                  default:
        #include <unistd.h>
                                                                     printf("Usage: %s <-e/-d> <in-file> <out-file>\n", argv[0]);
        #include <stdbool.h>
                                                                     return -1;
                                                  42
    11
        int main(int argc, char *argv[])
    12
            int nFd In = -1;
    13
            int nFd Out = -1;
    14
            int nRead, nWritten;
    15
            unsigned char p cInputData[AES BLOCK SIZE];
```

[파일명: aes_cfb128.c (1/2)]

unsigned char p cOutputData[AES BLOCK_SIZE];

unsigned char p_ckey[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};

unsigned char p cIvec[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};

int bEncryption = false;

int nOpt = -1:

AES KEY stKey;

17

19

21

22 23

24

7. AES 응용 (2/2)

□AES 기술 실습

```
int nBlockIndex = 0;
        while (1) {
            nRead = read(nFd In, p cInputData, AES BLOCK SIZE);
            if (nRead <= 0)
                break;
70
            if (bEncryption) {
                AES cfb128 encrypt(p cInputData, p cOutputData, nRead,
                                   &stKey, p cIvec, &nBlockIndex,
                                   AES ENCRYPT);
78
                AES cfb128 encrypt(p cInputData, p cOutputData, nRead,
                                   &stKey, p cIvec, &nBlockIndex,
                                   AES DECRYPT);
            nWritten = write(nFd Out, p cOutputData, nRead);
        close(nFd In);
        close(nFd Out);
        return 0;
                                      [파일명: aes_cfb128.c (2/2)]
```

```
printf("CKEY=");
for (int i = 0; i < 16; i++) {
    printf("%02X", p_ckey[i]);
}
printf("\n");

printf("IVEC=");
for (int i = 0; i < 16; i++) {
    printf("%02X", p_cIvec[i]);
}
printf("\n");

printf("Input File: %s\n", argv[optind]);
printf("Output File: %s\n", argv[optind + 1]);

nFd_In = open(argv[optind], O_RDONLY);
nFd_Out = open(argv[optind + 1], O_WRONLY | O_CREAT, 0644);

/* set the encryption stKey */
AES_set_encrypt_key(p_ckey, 128, &stKey);</pre>
```

수고하셨습니다.

