

운영체제 보안 2

- 암호화 프로그래밍

컴퓨터소프트웨어학과

김병국 교수



□ 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`
 - OpenSSL 관련 기능을 프로그래밍할 수 있는 API들을 제공
 - 최신화 관련하여 오류 메시지가 나타날 경우, 아래의 “최신화”를 실행 후 이 부분을 재시도

■ 설치 확인

- 명령어 : `sudo apt-get list libssl-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)



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 틀리면 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

```
typedef struct rc4_key_st {
    RC4_INT x, y;
    RC4_INT data[256];
} RC4_KEY;

void RC4_set_key(RC4_KEY *key, int len, const unsigned char *data);
void RC4(RC4_KEY *key, size_t len, const unsigned char *indata,
         unsigned char *outdata);
```

[RC4관련 함수들]

⚙️ 빌드 옵션

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



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)

```
typedef struct rc4_key_st {
    RC4_INT x, y;
    RC4_INT data[256];
} RC4_KEY;

void RC4_set_key(RC4_KEY *key, int len, const unsigned char *data);
void RC4(RC4_KEY *key, size_t len, const unsigned char *indata,
         unsigned char *outdata);
```

[RC4관련 함수들]

빌드 옵션

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



4. RC4 응용 프로그래밍

RC4 실습

```
1 1 #include <stdio.h>
2 2 #include <sys/types.h>
3 3 #include <sys/stat.h>
4 4 #include <fcntl.h>
5 5 #include <openssl/rc4.h>
6 6 #include <string.h>
7 7 #include <unistd.h>
8 8 #include <stdbool.h>
9
10 int main(int argc, char *argv[])
11 {
12 2 int nFd_In = -1;
13 int nFd_Out = -1;
14
15 unsigned char p_cKey[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
16
17 int nDataLen;
18 unsigned char p_cInputData[BUFSIZ] = {0};
19 unsigned char p_cOutputData[BUFSIZ] = {0};
20
21 RC4_KEY stRc4Key;
22
23 if (argc != 3) {
24     printf("Usage: cmd <in-file> <out-file>\n");
25     return -1;
26 }
27
28 printf("KEY=");
29 for (int i = 0; i < 16; i++) {
30     printf("%02X", p_cKey[i]);
31 }
32 printf("\n");
33
34 nFd_In = open(argv[1], O_RDONLY);
35 nFd_Out = open(argv[2], O_WRONLY | O_CREAT, 0644);
36
```

```
36
37 3 printf("set key.\n");
38 RC4_set_key(&stRc4Key, 16, p_cKey);
39
40 printf("Start to encrypt.\n");
41 while (1) {
42     nDataLen = read(nFd_In, p_cInputData, BUFSIZ);
43     if (nDataLen <= 0)
44         break;
45
46     RC4(&stRc4Key, nDataLen, p_cInputData, p_cOutputData);
47
48     if (write(nFd_Out, p_cOutputData, nDataLen) <= 0)
49         break;
50 }
51
52 close(nFd_In);
53 close(nFd_Out);
54
55 printf("En/De-cryption is done.\n");
56
57 return 0;
58 }
```

[파일명: rc4.c]

- 컴파일: gcc rc4.c -lcrypto -o rc4
- 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

```
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]

⚙️ 빌드 옵션

- gcc를 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
- *key : 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]

⚙️ 빌드 옵션

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



7. AES 응용 (1/2)

□ AES 기술 실습

1

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <sys/stat.h>
4  #include <fcntl.h>
5  #include <openssl/aes.h>
6  #include <string.h>
7  #include <getopt.h>
8  #include <unistd.h>
9  #include <stdbool.h>
10
11 int main(int argc, char *argv[])
12 {
13     int nFd_In = -1;
14     int nFd_Out = -1;
15     int nRead, nWritten;
16     unsigned char p_cInputData[AES_BLOCK_SIZE];
17     unsigned char p_cOutputData[AES_BLOCK_SIZE];
18     int bEncryption = false;
19     int nOpt = -1;
20
21     unsigned char p_ckey[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
22     unsigned char p_cIvec[16] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
23
24     AES_KEY stKey;
```

[파일명: aes_cfb128.c (1/2)]

2

```
26 if (argc != 4) {
27     printf("Usage: %s <-e/-d> <in-file> <out-file>\n", argv[0]);
28     return -1;
29 }
30
31 while((nOpt=getopt(argc, argv, "ed"))>0) {
32     switch (nOpt) {
33         case 'e':
34             bEncryption = true;
35             break;
36         case 'd':
37             bEncryption = false;
38             break;
39         default:
40             printf("Usage: %s <-e/-d> <in-file> <out-file>\n", argv[0]);
41             return -1;
42     }
43 }
44 ..
```



7. AES 응용 (2/2)

□ AES 기술 실습

```
66 4 int nBlockIndex = 0;
67 while (1) {
68     nRead = read(nFd_In, p_cInputData, AES_BLOCK_SIZE);
69     if (nRead <= 0)
70         break;
71
72     if (bEncryption) {
73         AES_cfb128_encrypt(p_cInputData, p_cOutputData, nRead,
74                             &stKey, p_cIvec, &nBlockIndex,
75                             AES_ENCRYPT);
76     }
77     else {
78         AES_cfb128_encrypt(p_cInputData, p_cOutputData, nRead,
79                             &stKey, p_cIvec, &nBlockIndex,
80                             AES_DECRYPT);
81     }
82
83     nWritten = write(nFd_Out, p_cOutputData, nRead);
84 }
85
86 close(nFd_In);
87 close(nFd_Out);
88
89 return 0;
90 }
```

[파일명: aes_cfb128.c (2/2)]

```
45 3 printf("CKEY=");
46 for (int i = 0; i < 16; i++) {
47     printf("%02X", p_ckey[i]);
48 }
49 printf("\n");
50
51 printf("IVC=");
52 for (int i = 0; i < 16; i++) {
53     printf("%02X", p_cIvec[i]);
54 }
55 printf("\n");
56
57 printf("Input File: %s\n", argv[optind]);
58 printf("Output File: %s\n", argv[optind + 1]);
59
60 nFd_In = open(argv[optind], O_RDONLY);
61 nFd_Out = open(argv[optind + 1], O_WRONLY | O_CREAT, 0644);
62
63 /* set the encryption stKey */
64 AES_set_encrypt_key(p_cke, 128, &stKey);
```



수고하셨습니다.

