정보보호

HW05 : DES,AES-CBC Cipher

제 출 일	2018년 10월 18일
담당교수	류재철
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. 공통과정

■ 과제해결 과정

저번 과제와 동일한 작동 방식으로 구현을 하였다.
암복호화 선택, 입력 파일명, 키 입력, 암호화 형식 선택으로 입력을 받는다

```
void main()
{
   char inputFileName[256];
   char outputFileName[256];
   FILE *input_FD;
   FILE *output_FD;

   int mode;
   char key[30]={0, };
   int cipher;
   int t=0;
   char buff[BLOCK_SIZE]={0, };

   printf(">> Input mode [ 1 : ENC , 2 : DEC ]");
   scanf("%d",&mode);//암복호화 선택
   printf(">> Input file name : ");
   scanf("%s",inputFileName);
   printf("input key : ");
   scanf("%s",key);//평행이동할 키값을 받음
   printf(">> Input Cipher mode [ 1 : aes-cbc , 2 : aes-ctr 3 : des ]");
   scanf("%d",&cipher);//암호화 형식선택

   if (mode==1)
   {
      sprintf(outputFileName,"plain.enc");
   }
   else if(mode == 2)
   {
      sprintf(outputFileName,"plain.enc.dec");
   }
   else{
      printf("[!] Mode Error!\n");
      exit(1);
   }

   input_FD=fopen(inputFileName,"rb");
   output_FD=fopen(outputFileName,"wb");
```

암복호화에 맞게 파일을 생성해준다.

```
while(0<(t=fread(&buff,sizeof(char),BLOCK_SIZE,input_FD))){
    int res=0;
    if(cipher==1){
        res = aes_cbc(buff,key,t,mode);//1번일 경우 cbc진행
    }
    else if(cipher==2){
        //2번 ctr은 미구현
    }
    else if(cipher==3){
        res = des(buff,key,t,mode);//3번일 경우 des 진행
    }
    else printf("[!] Cipher mode Error!\n");

    if(res>0&&res<BLOCK_SIZE)
    fwrite(&buff,sizeof(char),res,output_FD);
    else fwrite(&buff,sizeof(char),BLOCK_SIZE,output_FD);
    //리턴된 res값을 이용하여서 파일에 써준다
    //블럭사이즈보다 작고 0보다 크다면 해당하는 숫자를
    //그외에는 블럭사이즈로 한다 블럭사이즈는 128로 지정하였다.
    memset(buff,0,sizeof(char)*BLOCK_SIZE);
}

fclose(output_FD);
fclose(input_FD);
printf("[!] Cipher Complete\n");
```

선택한 암호화 방식에 따라 진행한다.

블록사이즈 만큼 읽으면서 진행하여서 방식에따라 암,복호화 진행후 만들어진 값의 길이를 리턴한다.

리턴된 값을 이용하여서 출력파일에 작성하였다.

2. AES-CBC

■ 과제해결 과정

```
unsigned int aes_cbc(unsigned char * msg,unsigned char *key, unsigned int msg_len, int mode)
{
   AES_KEY  aes_ks;
   unsigned char iv[AES_BLOCK_SIZE];
   unsigned int i,result,padding;
   unsigned char block_in[BLOCK_SIZE] = {0, };
   unsigned char block_out[BLOCK_SIZE] = {0, };
   result=0;
   memset(iv, 0, sizeof(iv));
   memcpy(block_in, msg, msg_len);
   if(mode==1){
      if(msg_len <BLOCK_SIZE){
         padding = BLOCK_SIZE - msg_len;
         int count=padding;
         while(count>=1){
         block_in[BLOCK_SIZE -count] = padding;
         count--;
         }//들어온 입력값이 블럭사이즈보다 작다면 나머지 칸을 패딩으로 다 채워준다
      }//맨마지막만 채우는 식으로 패딩을하면 복호화시 꽉찬 블럭을 복호화할때 에러 발생해서 패딩 방법
을 바꿨다.
      AES_set_encrypt_key(key,BLOCK_SIZE,&aes_ks);
      AES_cbc_encrypt(block_in,block_out,BLOCK_SIZE,&aes_ks,iv,AES_ENCRYPT);
      result=BLOCK_SIZE;//암호화하고 나온값은 항상 블럭사이즈이다.
   }
   else if(mode ==2){
      AES_set_decrypt_key(key,BLOCK_SIZE,&aes_ks);
      AES_cbc_encrypt(block_in,block_out,BLOCK_SIZE,&aes_ks,iv,AES_DECRYPT);
      //복호화는 무조건 블럭사이즈로해서 일단 바로 복호화를 진행한다.
      padding = block_out[BLOCK_SIZE-1];//맨 마지막 비트를 패딩이라고 가정
      int count=padding;
      while(count>=2){
      if( block_out[BLOCK_SIZE-count]!=block_out[BLOCK_SIZE-count+1]) break;
      count--;
      }
      //패딩값 만큼의 비트수가 패딩값으로 채워져 있는지 확인. 다르다면 break문으로 중단
      if(count==1)
      result = BLOCK_SIZE-padding;
      else result=BLOCK_SIZE;
   }//패딩이 있다면 count가 1까지 감소 했을 것임으로 result는 블럭사이즈에서 패딩값을 빼준다.
   memcpy(msg,block_out,BLOCK_SIZE);
   return result;
}
```

aes.h자료를 참고하여서 들어가야할 변수의 타입들을 확인하여 알맞게 IV벡터와 문자열들을 생성했다.

암호화를 진행할 경우 패딩을 생각하여야하는데 마지막칸만 패딩으로 채워서 확인할경우에는 복호화시 꽉찬 블록일 경우에 오류가 생기는 것을 발견하여서 패딩블럭에 모두 패딩값으로 채우고 복호화시 다 같은지 확인하는 형식으로 구현하였다.

AES 암호화 키를 생성하여 aes_ks로 넣고 cbc암호화를 진행하면 결과값이 block_out에 저장된다.

복호화시에는 AES 복호화 키를 생성하여 aes_ks로 넣고 cbc 복호화를 진행하면 결과 값이 block_out에 저장되는데 이때는 패딩이 있을 가능성을 생각해야한다.

맨 마지막 비트의 값만큼의 같은수가 동일한 비트로 채워져있는지 확인하여서 다 같다면 패딩이 맞고 아니라면 패딩이 아니다.

패딩이라면 최종 리턴값에 블록사이즈에서 패딩값을 뺀 만큼을 리턴하여서 파일 출력이 올바르게 되도록 하였다.

■ 실행결과 화면

블록 단위는 128, 입력 파일의 길이는 154

1. AES-CBC 암호화

```
shin@shin:~/Desktop/homework/4$ ls
main.c  plain  sample  view.py
shin@shin:~/Desktop/homework/4$ ./sample
>> Input mode [ 1 : ENC , 2 : DEC ]1
>> Input file name : plain
input key : qwer
>> Input Cipher mode [ 1 : aes-cbc , 2 : aes-ctr 3 : des ]1
[!] Cipher Complete
shin@shin:~/Desktop/homework/4$ python view.py
The size of the file "plain" is 154.
--------------------------------------------------------------------------------
0000 0000:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0010:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 31 31  |od over!! nice11|
0000 0020:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0030:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 32 32  |od over!! nice22|
0000 0040:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0050:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 33 33  |od over!! nice33|
0000 0060:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0070:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 34 34  |od over!! nice44|
0000 0080:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0090:  6F 64 20 6F 76 65 72 21-21 0A                    |od over!!.      |
--------------------------------------------------------------------------------

The size of the file "plain.enc" is 256.
--------------------------------------------------------------------------------
0000 0000:  F1 1B E5 98 84 13 E7 28-D2 F0 7F 0D 4A BC 6D 8A  |.......(...J.m.|
0000 0010:  B3 50 A5 92 32 5B 90 2F-83 62 07 83 E9 C4 C5 69  |.P..2[./.b.....i|
0000 0020:  29 30 4D 49 59 79 0D 87-EB 1F DE E9 E5 ED EE BE  |)0MIYy..........|
0000 0030:  88 53 C5 59 DC DE F2 97-21 84 2A B8 3C F1 D0 BF  |.S.Y....!.*.<...|
0000 0040:  DD E1 58 4F 4F 72 BD 25-5A 7F BB EE DF 41 0B B1  |..XOOr.%Z...A..|
0000 0050:  6B A9 B4 C2 84 3E E1 AD-EB 0C 01 33 A9 4A 33 C5  |k....>.....3.J3.|
0000 0060:  BE 9E DF 65 7B DC B6 B9-95 AB F5 E1 D7 2E 13 A9  |...e{...........|
0000 0070:  96 16 4B C4 01 B6 02 FC-D4 77 94 32 3B 5C 5E 0F  |..K......w.2;\^.|
0000 0080:  F1 1B E5 98 84 13 E7 28-D2 F0 7F 0D 4A BC 6D 8A  |.......(...J.m.|
0000 0090:  A9 F4 D5 00 0D E6 50 15-01 5F 4B 48 B9 41 A8 82  |......P.._KH.A..|
0000 00A0:  84 46 36 D6 C2 10 85 BA-3F C4 60 62 7D 65 A1 4A  |.F6.....?.`b}e.J|
0000 00B0:  F4 27 EC 3F 7D A8 3A 10-6F 97 61 53 06 6D 49 EF  |.'.?}.:.o.aS.mI.|
0000 00C0:  09 3D 6A A0 6E AF 07 33-3A 7C B9 1A 09 7D 1C 2A  |.=j.n..3:|...}.*|
0000 00D0:  D3 80 0F 98 6B 9D BB E1-B2 46 69 C6 C9 2D 24 E5  |....k....Fi..-$.|
0000 00E0:  87 F1 2E 51 86 E2 A7 EB-99 85 E4 BC 97 8D 74 31  |...Q..........t1|
0000 00F0:  6C B5 E4 85 1E F8 AD A4-80 EE 81 38 39 81 F0 E8  |l..........89...|
--------------------------------------------------------------------------------

The file 'plain.enc.dec' does not exist.

shin@shin:~/Desktop/homework/4$
```

2. AES-CBC 복호화

```
shin@shin:~/Desktop/homework/4$ python view.py
The size of the file "plain" is 154.
------------------------------------------------------------------------------
0000 0000:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0010:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 31 31  |od over!! nice11|
0000 0020:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0030:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 32 32  |od over!! nice22|
0000 0040:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0050:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 33 33  |od over!! nice33|
0000 0060:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0070:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 34 34  |od over!! nice44|
0000 0080:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0090:  6F 64 20 6F 76 65 72 21-21 0A                    |od over!!.      |
------------------------------------------------------------------------------

The size of the file "plain.enc" is 256.
------------------------------------------------------------------------------
0000 0000:  F1 1B E5 98 84 13 E7 28-D2 F0 7F 0D 4A BC 6D 8A  |.......(...J.m.|
0000 0010:  B3 50 A5 92 32 5B 90 2F-83 62 07 83 E9 C4 C5 69  |.P..2[./.b.....i|
0000 0020:  29 30 4D 49 59 79 0D 87-EB 1F DE E9 E5 ED EE BE  |)0MIYy..........|
0000 0030:  88 53 C5 59 DC DE F2 97-21 84 2A B8 3C F1 D0 BF  |.S.Y....!.*.<...|
0000 0040:  DD E1 58 4F 4F 72 BD 25-5A 7F BB EE DF 41 0B B1  |..XOOr.%Z...A..|
0000 0050:  6B A9 B4 C2 84 3E E1 AD-EB 0C 01 33 A9 4A 33 C5  |k....>.....3.J3.|
0000 0060:  BE 9E DF 65 7B DC B6 B9-95 AB F5 E1 D7 2E 13 A9  |...e{...........|
0000 0070:  96 16 4B C4 01 B6 02 FC-D4 77 94 32 3B 5C 5E 0F  |..K......w.2;\^.|
0000 0080:  F1 1B E5 98 84 13 E7 28-D2 F0 7F 0D 4A BC 6D 8A  |.......(...J.m.|
0000 0090:  A9 F4 D5 00 0D E6 50 15-01 5F 4B 48 B9 41 A8 82  |......P.._KH.A..|
0000 00A0:  84 46 36 D6 C2 10 85 BA-3F C4 60 62 7D 65 A1 4A  |.F6.....?.`b}e.J|
0000 00B0:  F4 27 EC 3F 7D A8 3A 10-6F 97 61 53 06 6D 49 EF  |.'.?}.:.o.aS.mI.|
0000 00C0:  09 3D 6A A0 6E AF 07 33-3A 7C B9 1A 09 7D 1C 2A  |.=j.n..3:|...}.*|
0000 00D0:  D3 80 0F 98 6B 9D BB E1-B2 46 69 C6 C9 2D 24 E5  |....k....Fi..-$.|
0000 00E0:  87 F1 2E 51 86 E2 A7 EB-99 85 E4 BC 97 8D 74 31  |...Q..........t1|
0000 00F0:  6C B5 E4 85 1E F8 AD A4-80 EE 81 38 39 81 F0 E8  |l..........89...|
------------------------------------------------------------------------------

The size of the file "plain.enc.dec" is 154.
------------------------------------------------------------------------------
0000 0000:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0010:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 31 31  |od over!! nice11|
0000 0020:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0030:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 32 32  |od over!! nice22|
0000 0040:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0050:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 33 33  |od over!! nice33|
0000 0060:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0070:  6F 64 20 6F 76 65 72 21-21 20 6E 69 63 65 34 34  |od over!! nice44|
0000 0080:  74 65 6D 70 2E 74 65 73-74 20 31 32 33 2E 67 6F  |temp.test 123.go|
0000 0090:  6F 64 20 6F 76 65 72 21-21 0A                    |od over!!.      |
------------------------------------------------------------------------------
```

3. DES

■ 과제해결 과정

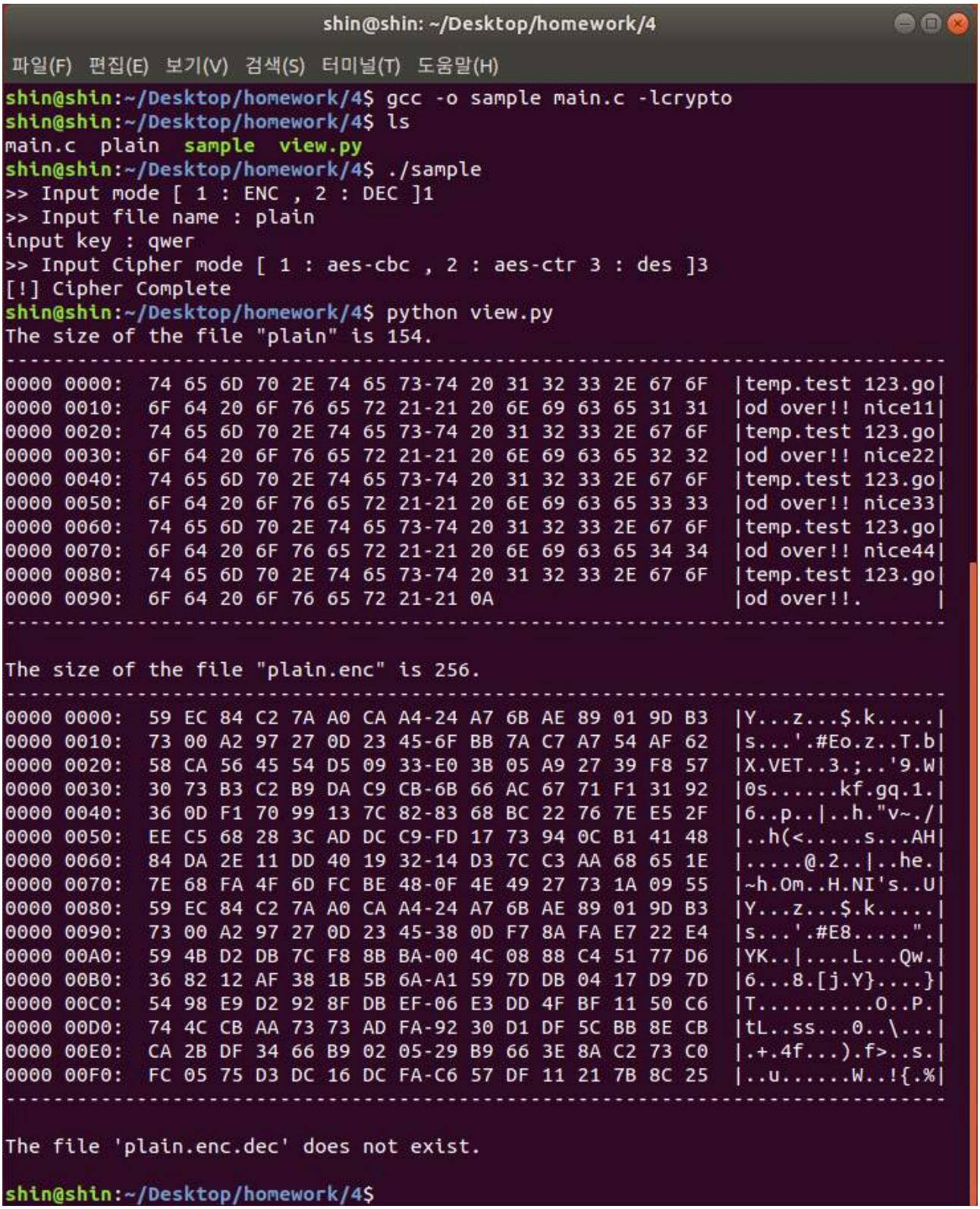
```
unsigned int des(unsigned char * msg,unsigned char *key, unsigned int msg_len, int mode)
{
   DES_key_schedule des_ks;
   DES_cblock  des_key = {0, };
   DES_cblock  iv = {0, };
   unsigned int i,result, padding;

   unsigned char block_in[BLOCK_SIZE] = {0, };
   unsigned char block_out[BLOCK_SIZE] = {0, };

   DES_string_to_key(key,&des_key);
   DES_set_key_checked(&des_key,&des_ks);

   memcpy(block_in, msg, msg_len);

   if(mode==1){
      if(msg_len <BLOCK_SIZE){
         padding = BLOCK_SIZE - msg_len;
         int count=padding;
         while(count>=1){
         block_in[BLOCK_SIZE -count] = padding;
         count--;
         }
      }//cbc와 동일하게 패딩
   DES_ncbc_encrypt(block_in,block_out,BLOCK_SIZE,&des_ks,&iv,DES_ENCRYPT);
      result=BLOCK_SIZE;
   }//암호화 할땐  결과가 항상 블럭사이즈
   else if(mode==2){
   DES_ncbc_encrypt(block_in,block_out,BLOCK_SIZE,&des_ks,&iv,DES_DECRYPT);
      padding = block_out[BLOCK_SIZE-1];
      int count=padding;
      while(count>=2){
      if( block_out[BLOCK_SIZE-count]!=block_out[BLOCK_SIZE-count+1]) break;
      count--;
      }//복호화 할때도 cbc와 동일하게 패딩 확인 후 제거
      if(count==1)
      result = BLOCK_SIZE-padding;
      else result =BLOCK_SIZE;
   }
   memcpy(msg,block_out,BLOCK_SIZE);

   return result;
}
```

cbc와 같은 형식이지만 des.h를 참고하여 암호화할 때 들어가는 타입들이 다르기 때문에 iv벡터와 des_key를 알맞은 타입을 설정해주었다.
암호화 과정시에 cbc와 동일하게 패딩을 해주고 복호화 과정에도 동일하게 패딩확인후 제거하였다.

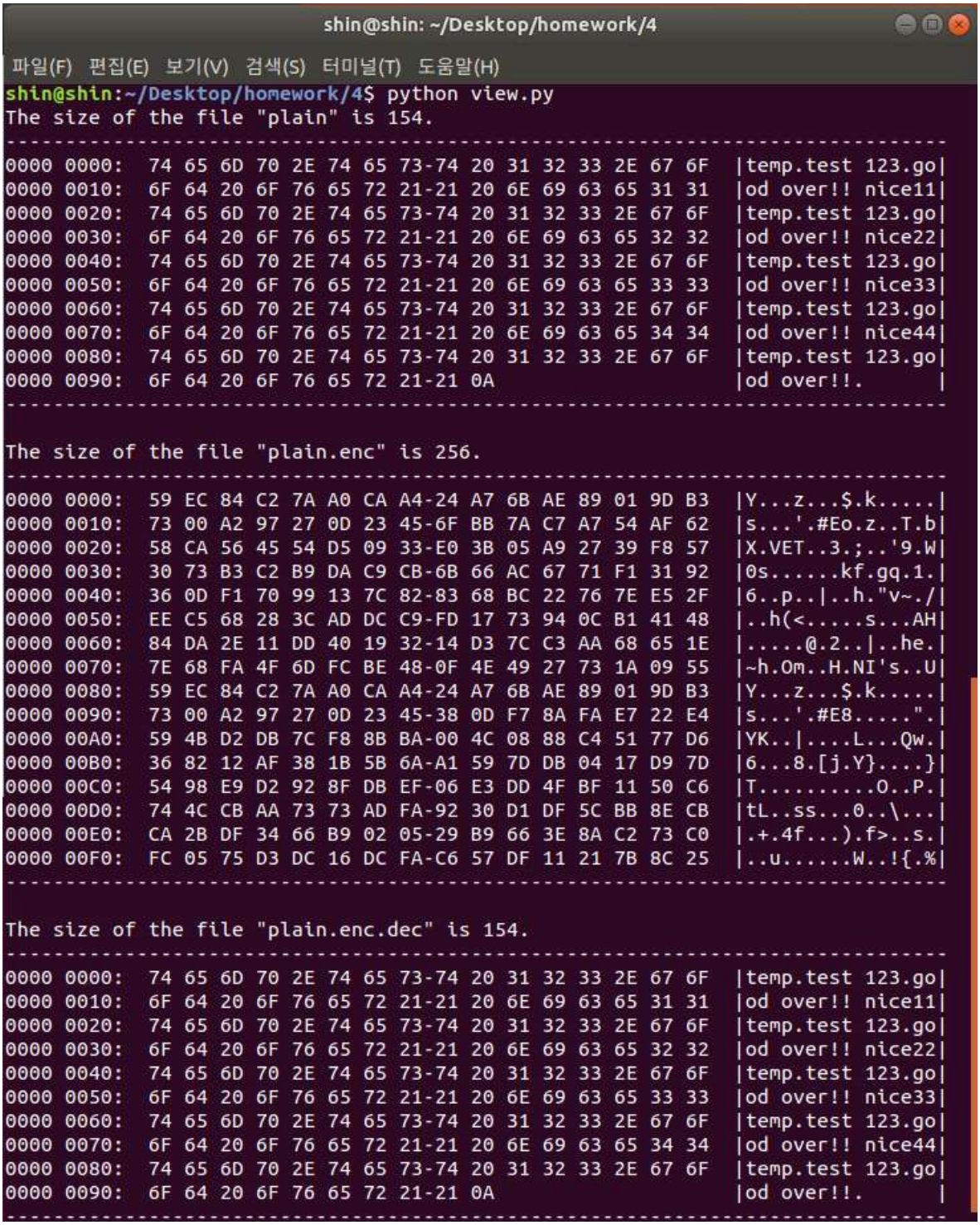
■ 실행결과 화면

블록 단위는 128, 입력 파일의 길이는 154

1. DES 암호화



2. Des 복호화



느낀점 : 처음에 BLOCK_SIZE에 대한 값에 이해가 잘 안 되서 막혔었는데 예시를 보고 이해를 하고 빨리 과제를 완료하였다.