

DES 개발 2019253022 홍한울

목표 : DES 알고리즘을 구현한다.

구현 동작 : 평문1, 평문2를 각각 다른 랜덤 64-bit key를 생성하고, 암호화, 복호화를 진행하여 최종적으로 암호문1,2 복호문1,2 를 만든다.

코드 및 주요 함수

- **Main 함수**

- 호출 함수 :
- process_file(input_filename1, cy_filename1, de_filename1);
- process_file(input_filename2, cy_filename2, de_filename2);

```
397 int main() {  
398     srand(_Seed:(unsigned int)time(_Time:NULL));  
399  
400     const char *input_filename1 = "Plain Text 1.txt";  
401     const char *input_filename2 = "Plain Text 2.txt";  
402     const char *cy_filename1 = "Cypher Text 1.txt";  
403     const char *cy_filename2 = "Cypher Text 2.txt";  
404     const char *de_filename1 = "Decryp Text 1.txt";  
405     const char *de_filename2 = "Decryp Text 2.txt";  
406  
407     //new code  
408     process_file(input_filename1, cy_filename1, de_filename1);  
409     process_file(input_filename2, cy_filename2, de_filename2);  
410  
411     return 0;  
412 }
```

파일 2개의 처리로 process_file 함수에서 각 파일들에 대한 입, 출력 및 암호화, 복호화 처리가 이루어짐.

- **void process_file**

- 동작 정의
 - ◆ 파일 입출력.
 - ◆ 랜덤키 생성 및 sub키 목록 생성
 - ◆ 암호화 복호화 수행
- 호출 함수:
 - ◆ generate_random_64bit()
 - ◆ generate_subkeys(key, subkeys)

- ◆ test_text_to_uint64(plaintext);
- ◆ des_encryption() (암호화 동작)
- ◆ des_decryption() (복호화 동작)
- ◆ uint64_to_text(decryption, decrypted_text)

- **uint64_t des_encryption(), uint64_t des_decryption()**

■ 동작 정의:

- ◆ 초기, 마지막 permutation 진행
- ◆ round진행

■ 호출 함수:

- ◆ permutation(binary, IP, 64);
- ◆ round(output, subkeys[i],i,0); or round(output, subkeys[i],i,-1);
- ◆ permutation(output, FP, 64);

- **round(uint64_t binary,uint64_t subkey,int count,int flag)**

■ 동작정의:

- ◆ Mixer 동작
- ◆ Swapper 동작 (암호화, 복호화는 각각 순행, 역행으로 서브키 진행 및 마지막 Swapper flag를 통한 동작 구분 구현)

■ 호출 함수:

- ◆ des_function(R, subkey)

- **des_function(uint32_t right, uint64_t subkey)**

■ 동작 정의:

- ◆ Expansion P-Box 수행
- ◆ R와 subkey의 Xor 연산 수행
- ◆ S-Box 수행
- ◆ Straight P-Box 수행

■ 호출 함수:

- ◆ expansion_pbox(right)
- ◆ sbox(xor_result)
- ◆ straight_pbox

개발과정

test.h test.c를 통해서 main에서 구현할 기능에 대해서 먼저 작성 후 test 값들을 활용하여 code test를 진행하여 검증된 코드를 des.c에서 개발을 진행함.

-테스트 예시

Initial permutation 검증

```
plaintext : 4E6F772069732074
test_key  : 0123456789ABCDEF
parity    : 00F0CCAA0AACCF00
init perm : B7A4873600FE1327
```

Parity 검증

```
test_key : 0123456789ABCDEF
parity   : 00F0CCAA0AACCF00
init perm : B7A4873600FE1327
```

Compression P-Box 검증

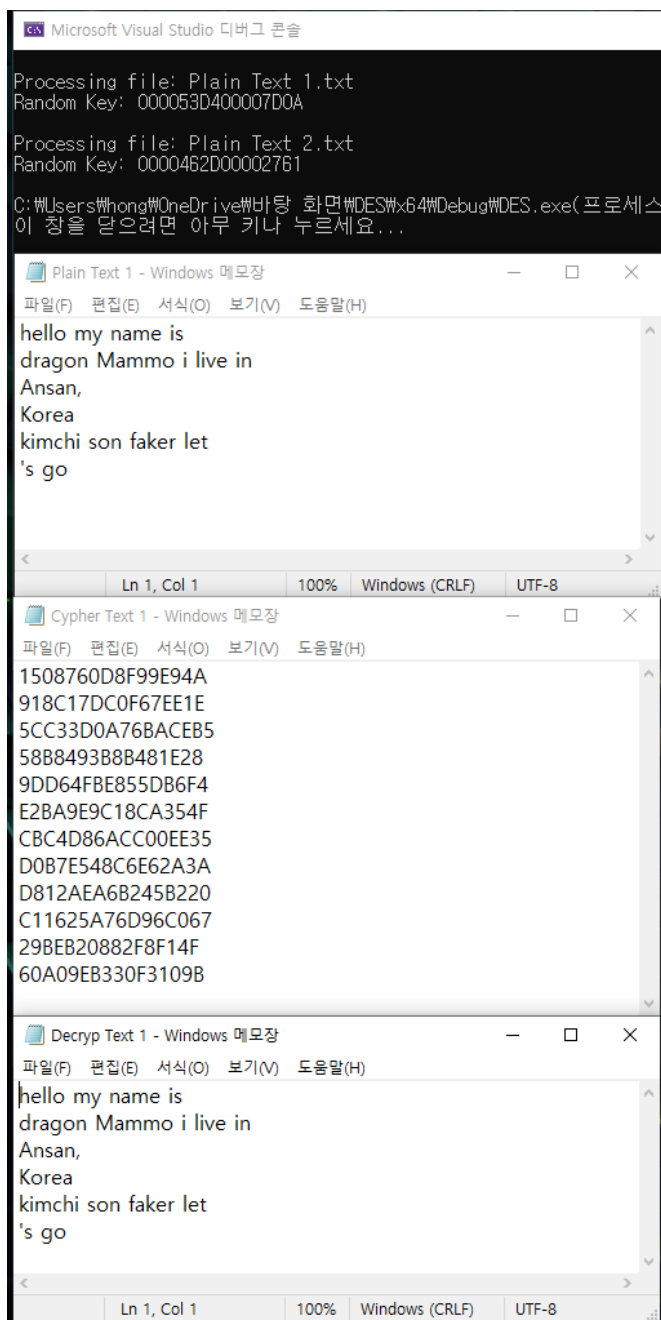
```
test_key : 0123456789ABCDEF
parity   : 00F0CCAA0AACCF00
compressed : 00000B02679B49A5
compressed : 000069A659256A26
compressed : 000045D48AB428D2
compressed : 00007289D2A58257
compressed : 00003CE80317A6C2
compressed : 000023251E3C8545
compressed : 00006C04950AE4C6
compressed : 00005788386CE581
compressed : 0000C0C9E926B839
compressed : 000091E307631D72
compressed : 0000211F830D893A
compressed : 00007130E5455C54
compressed : 000091C4D04980FC
compressed : 00005443B681DC8D
compressed : 0000B691050A16B5
compressed : 0000CA3D03B87032
```

총 동작 테스트

```
test_key : 0123456789ABCDEF
plaintext : 4E6F772069732074
Cyphertext : 86E8D6BFC69B6CB8
deCyphertext : 4E6F772069732074
```

결과:

Enter로 구분된 평문에 대한 결과 (Plain Text 1.txt)



```
Microsoft Visual Studio 디버그 콘솔
Processing file: Plain Text 1.txt
Random Key: 000053D400007D0A
Processing file: Plain Text 2.txt
Random Key: 0000462D00002761
C:\Users\hong\OneDrive\바탕 화면\DES\x64\Debug\DES.exe( 프로세스
이 창을 닫으려면 아무 키나 누르세요...)

Plain Text 1 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
hello my name is
dragon Mammo i live in
Ansan,
Korea
kimchi son faker let
's go

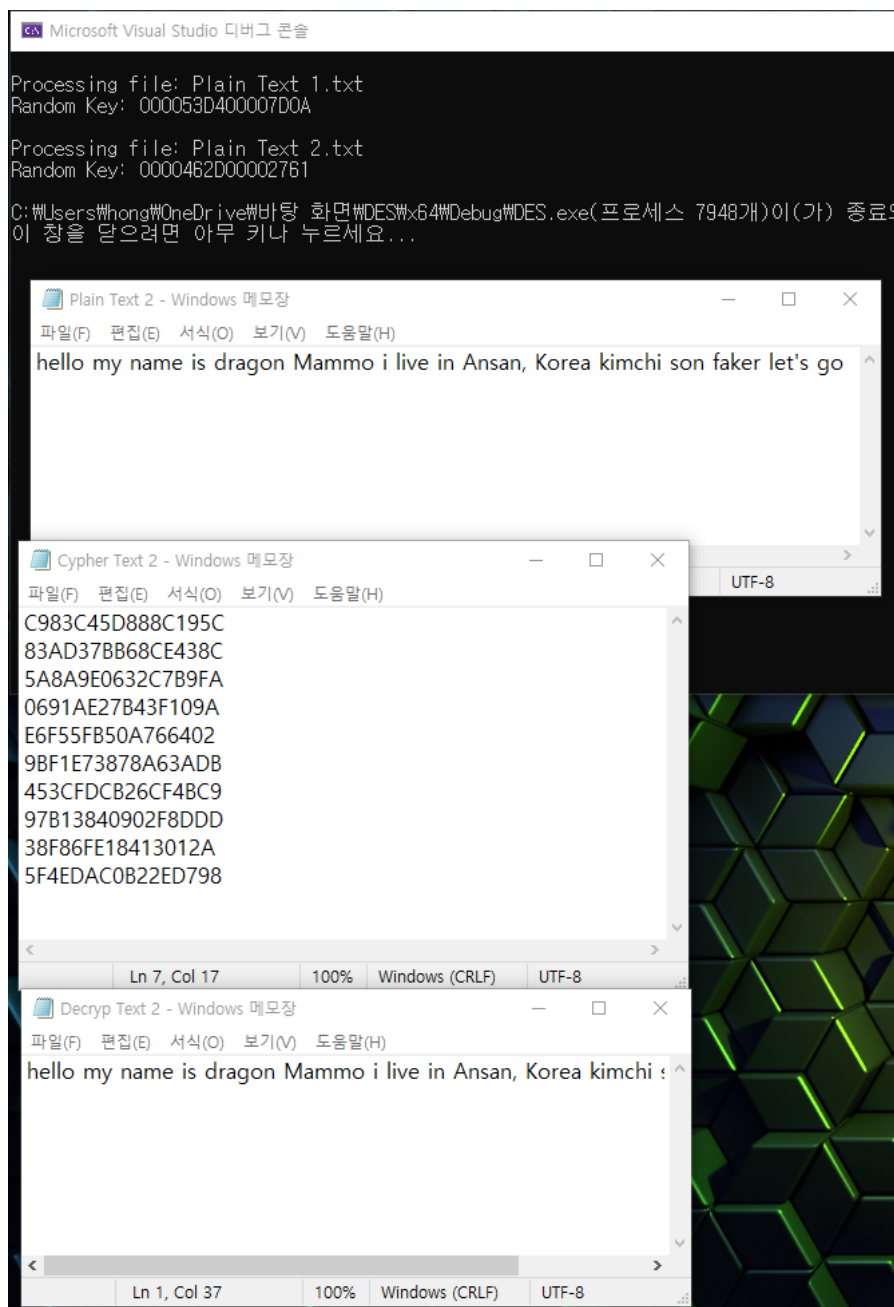
Ln 1, Col 1 100% Windows (CRLF) UTF-8

Cypher Text 1 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1508760D8F99E94A
918C17DC0F67EE1E
5CC33D0A76BACEB5
58B8493B8B481E28
9DD64FBE855DB6F4
E2BA9E9C18CA354F
CBC4D86ACC00EE35
D0B7E548C6E62A3A
D812AEA6B245B220
C11625A76D96C067
29BEB20882F8F14F
60A09EB330F3109B

Decryp Text 1 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
hello my name is
dragon Mammo i live in
Ansan,
Korea
kimchi son faker let
's go

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Enter로 구분되지 않은 평문에 대한 결과 (Plain Text 2.txt)



Issue: random 64bit 키 생성시 결과 사진과 같이 0000XXXX0000XXXX 와 같은 패턴으로 키가 생성되지만,

테스트 결과 key는 위 패턴이 아닌 무작위 패턴에도 프로그램 정상작동을 확인

-실행 시 추가적인 입출력 없이 평문1,2 파일이름만 맞으면 프로그램 실행가능.

Plain Text 1 / Plain Text 2