Programming Assignment #3
(파일 이름: **`crypt.c`**, Due: 7월 4일 23:59, Hard Deadline)

암/복호화 프로그램 만들기

1. 개요

C 언어로 16바이트의 메시지를 8바이트의 키로 암호화하고, 암호화한 데이터를 복호화할 수 있는 프로그램을 만들어본다.

2. 암호화 방법

본 프로젝트에서 메시지 암호화는 다음과 같은 두 단계로 이루어져 있다:

A. 치환 (Substitution)

치환을 위한 함수는 **`sub`** 이다. 여기에 입력으로 들어온 16바이트의 값을 $i_0, i_1, \cdots, i_{15}$라고 하고, 8바이트의 키 값을 $k_0, k_1, \cdots, k_7$이라고 하자. 그리고 출력으로 나갈 16바이트의 값을 $o_0, o_1, \cdots, o_{15}$라고 하자. 그러면 다음과 같은 식이 성립한다 (단, $\oplus$는 XOR이다):

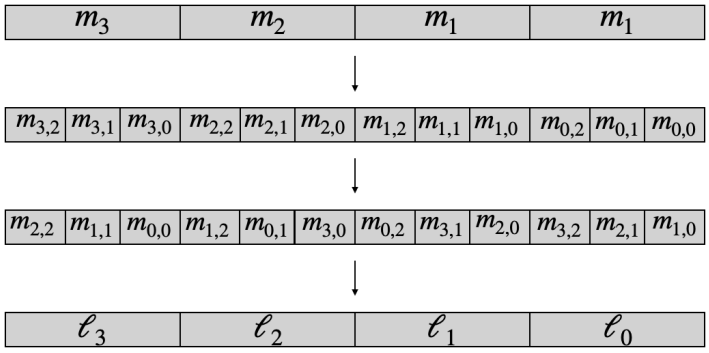
$o_n = ((i_n \times 23) \mod 256) \oplus k_n$ ($n < 8$일 때)
$o_n = ((i_n \times 23) \mod 256) \oplus k_{n-8}$ ($n \geq 8$일 때)

B. 비트 섞기 (Mixing bits)

비트 섞기를 위한 함수는 **`mix`** 이다. 여기에 입력으로 들어온 16바이트의 int형 값을 m_0, m_1, m_2, m_3이라고 하자. 그리고 출력으로 나갈 16바이트의 int형 값을 $\ell_0, \ell_1, \ell_2, \ell_3$이라고 하자. 또한, $i = 0, 1, 2, 3$일 때, m_i를 11비트, 10비트, 11비트로 각각 쪼갠 것을 각각 $m_{i,2}, m_{i,1}, m_{i,0}$라고 하자 (단, m_i에서 $m_{i,2}$이 왼쪽 11개의 비트, $m_{i,0}$가 오른쪽 11개의 비트, 그리고 $m_{i,1}$은 가운데 10개의 비트). 그러면 다음과 같이 출력이 결정된다:

$$\ell_0 = m_{3,2} \| m_{2,1} \| m_{1,0},\ \ell_1 = m_{0,2} \| m_{3,1} \| m_{2,1},\ \ell_2 = m_{1,2} \| m_{0,1} \| m_{3,0},\ \ell_3 = m_{2,2} \| m_{1,1} \| m_{0,0}$$

그림으로 나타내면 다음과 같다:



3. 복호화 방법

복호화는 암호화의 반대 과정으로 하면 되며, 다음과 같이 두 단계로 이루어져 있다.

A. Reverse Mixing Bits

이를 위한 함수는 **revmix** 이다. 이 과정은 **mix** 함수의 과정을 거꾸로 하면 된다.

B. Reverse Substitution

이를 위한 함수는 **revsub**이다. 여기에 입력으로 들어온 16바이트의 값을 $i_0, i_1, \cdots, i_{15}$라고 하고, 8바이트의 키 값을 $k_0, k_1, \cdots, k_7$이라고 하자. 그리고 출력으로 나갈 16바이트의 값을 $o_0, o_1, \cdots, o_{15}$라고 하자. 그러면 다음과 같은 식이 성립한다 (단, $\oplus$는 XOR이다):

$o_n = ((i_n \oplus k_n) \times 167) \mod 256$ ($n < 8$일 때)
$o_n = ((i_n \oplus k_{n-8}) \times 167) \mod 256$ ($n \geq 8$일 때)

4. 소스코드 설명

우선, 여러분의 계정에 **PA3** 디렉토리와 **crypt.c**라는 파일이 있을 것이다. 이 파일 안에서 주석으로 작성하라고 되어있는 부분을 작성하면 된다.

TEXT 타입: 평문(암호화하기 전의 문장)과 암호문을 표현하기 위한 타입으로, 16바이트의 값을 character로, 혹은 integer로 표현 가능하게 하기 위하여 union으로 작성되었다.

KEY 타입: 키를 표현하기 위한 타입으로, 8바이트의 값을 character로, 혹은 integer로 표현 가능하게 하기 위하여 union으로 작성되었다.

main 함수: **main**함수는 매개 변수를 실행 파일 이름 제외하고 세 개 받으며, 다음과 같이 실행한다. (실행 파일 이름이 **crypt**일 때)

암호화 할 때: **$ ./crypt [평문] [비밀번호] 1**
복호화 할 때: **$ ./crypt [16진수 암호문] [비밀번호] 2**

암호화를 위해서는 세 번째 인자를 1이라고 쓴다. 평문의 길이는 무조건 16바이트이어야 하며, 16바이트가 아니면 아무 메시지도 내지 않고 종료한다. 또한, 비밀번호도 무조건 8바이트이어야 하며, 8바이트가 아니면 아무 메시지도 발생시키지 않고 종료한다.

복호화를 위해서는 세 번째 인자를 2라고 쓴다. 암호문의 길이는 16바이트이지만, 각 바이트를 16진수로 나타내므로 실제 길이는 32가 된다. 따라서 16진수 암호문은 16진법 32자리 수가 될 것이다. 이렇게 입력하지 않으면 역시 아무 메시지도 내지 않고 종료시킨다. 또한 비밀번호도 마찬가지로 무조건 8바이트이어야 하며, 8바이트가 아니면 아무 메시지도 발생시키지 않고 종료한다.

암호화를 성공적으로 수행하면 암호문을 보여주는데, 16진수로 보여준다 (16진수로 출력하는 것 자체는 이미 구현되어 있으므로 추가로 구현할 필요는 없을 것이다). 암호문 역시 16바이트인데 16진수로 표현되므로 실제로는 32자리가 나올 것이다.

복호화할 때는 앞서 언급 했듯이 32자리의 16진수를 첫 번째 인자로 입력하고, 두 번째 인자에 비밀번호를 입력한다. 만일 암호화할 때와 같은 비밀번호를 입력했다면 성공적으로 암호가 풀려야 할 것이고, 화면에는 복호화된 결과 문자열을 보여줘야 할 것이다.

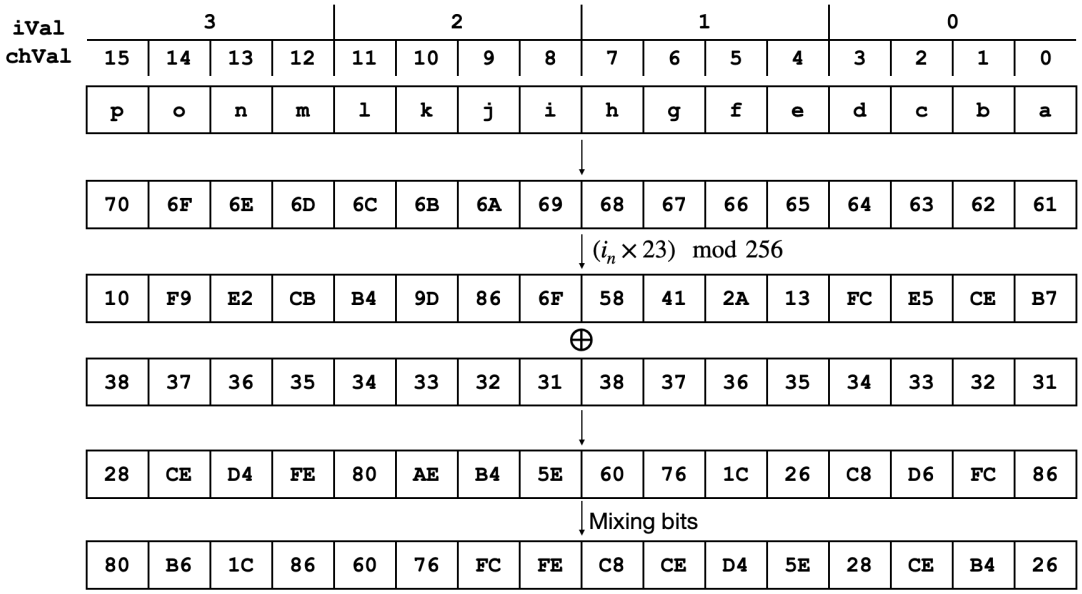
sub, **mix**, **revsub**, **revmix** 함수: 모두 **out**이라고 하는 변수를 반환한다. 여러분들의 역할은 **out**의 멤버 변수에 값을 채워넣는 것이다.

5. 실시 예

실행 파일 이름이 **crypt**라고 했을 때, 다음과 같이 실행할 수 있다:

```
$ ./crypt abcdefghijklmnop 12345678 1
26B4CE285ED4CEC8FEFC7660861CB680
$ ./crypt 26B4CE285ED4CEC8FEFC7660861CB680 12345678 2
abcdefghijklmnop
$
```

여기서 위와 같은 결과가 나오는 이유는 다음과 같다:



6. 참고 사항

1. 만일 **crypt.c** 를 다시 받고 싶다면 교수에게 요청할 것

2. Delay: 0점.

3. Copy나 다른 사람이 짜 준 경우 0점