정보보호개론 Assignment 1

2016310936 우승민

이번 과제는 vigenere variant cipher를 사용하여 암호화 시킨 plain text를 해독하는 tool을 만드는 과제입니다.

우선 암호화 코드는 아래와 같습니다.

```
#include <stdio.h>
#define KEY_LENGTH 5
int main(){
    unsigned char ch;
    FILE *fpIn, *fpOut;

    unsigned char key[KEY_LENGTH]={ 0x45, 0x01, 0x22, 0x02, 0x03};

    fpIn = fopen("plaintext.txt", "r");
    fpOut = fopen("hw1_input.txt", "wb");
    for(int i=0; fscanf(fpIn, "%c", &ch) != EOF; ++i){
        ch ^= key[i % KEY_LENGTH];
        fwrite(&ch, sizeof(ch), 1, fpOut);
    }

    fclose(fpIn);
    fclose(fpOut);

    return 0;
}
```

KEY_LENGTH 와 그만큼의 key 값이 주어지면 입력 받은 plaintext를 key와 xor시켜 hw1_input으로 암호화합니다.

이 코드의 암호해독 코드를 만들기 위해서는 크게 2가지 일을 수행해야합니다.

1. KEY_LENGTH 값 찾기

2. 사용된 KEY 찾기

위 2가지 일을 수행하면 찾은 KEY_LENGTH 와 KEY를 통해 암호화된 text를 해독할 수 있습니다.

우선 제 코드의 main 문과 전역변수부터 설명하겠습니다.

```
#include <stdio.h>$
$
char origin[5005]; // input file original$
char temp[5005];$
int alp[26]; // alphabet 사용 횟수$
float al[26]; // alphabet 빈도수$
int text;^I^I// input text 크기$
$
float pp[10]; // 혹시 빈도수가 0.065 정도로 안나왔을 때 대비하여$
$
unsigned char key[10];$
```

각 변수에 대해 설명하면 origin 은 암호화된 text 를 저장하는 용도로 사용되고, temp 는 나중에 key 를 찾으면서 변형된 text 를 저장하기 위해 사용됩니다.

alp 와 al array 는 암호화된 text 에서 alphabet 빈도수를 측정하기 위해 사용됩니다.

text 변수는 입력 받은 text 의 크기를 저장하는데 사용됩니다.

pp array 는 KEY_LENGTH 에 따라 합산된 빈도수를 저장하기 위해 사용됩니다.

key array 는 찾게 된 KEY_VALUE 값을 저장하기 위해 사용됩니다.

다음으로 main 에 대해 설명하겠습니다.

```
int main()$
^I{$
^I^Iunsigned char ch;$
^I^IFILE *fpIn, *fpOut;$
$
^I^IfpIn = fopen("hw1_input.txt", "r");$
^I^IfpOut = fopen("hw1_output.txt", "wb");$
^I^Iint i=0;$
^I^Ifor(i=0; fscanf(fpIn, "%c", &ch) != EOF; ++i){$
^I^I^I^Iorigin[i]=ch;$
^I^I}$
^I^Iorigin[i]= '₩0';$
^I^Itext = i;$
^I^IfclIse(fpIn);$
^I^I$
^I^Iint key_length = find_length();$
$
^I^Ifind_key(key_length);$
```

우선 입력 받은 text 를 origin 에 저장하고 key_length 에 find_length() 함수를 통해 KEY_LENGTH 를 찾은 후 저장하게 하였고, 이어서 find_key(int) 함수를 통해 KEY_VALUE 값들을 찾게 해주었습니다.

```
find_key(key_length);
char value[5];
for(i=0; i< key_length; i++){                    //output 파일 첫 줄에 value값 출력
    sprintf(value, "0x%02x", key[i]);
    value[4]=' ';
    fwrite(&value,sizeof(value), 1, fpOut);
}
ch = '₩n';
fwrite(&ch, sizeof(ch), 1, fpOut);

for(i=0; i<text; i++){              // output 파일 두번 째 줄부터 변환된 값 출력
    ch = key[i % key_length] ^ origin[i];
    fwrite(&ch, sizeof(ch), 1, fpOut);
}
fclose(fpOut);

return 0;
}
```

KEY_VALUE 값을 전부 찾으면, output 파일 상단에 순서대로 출력한 이후 다음 줄부터 해독한 text 를 출력하게 만들어주었습니다.

다음으로 KEY_LENGTH 값을 찾는 함수인 find_length() 대해 설명하겠습니다.

```
int find_length(){
    for(int i=1; i<=10; i++){
        int cnt = 1; // 읽는 글 위치를 넘겨줌
        int sum = 0; // alphabet 총 횟수

        while(cnt < text){              // 각 alphabet 횟수를 세고 총 alphabet 수도 샌다.
            if('a' <= origin[cnt] && origin[cnt] <= 'z'){
                alp[origin[cnt]-97]++;
                sum++;
            }
            if('A' <= origin[cnt] && origin[cnt] <= 'Z'){
                alp[origin[cnt]-65]++;
                sum++;
            }
            cnt = cnt + i; // length 단위
        }
```

우선 KEY_LENGTH가 1부터 10까지 가능하기 때문에 for문도 1부터 10까지 반복하게 해주었고, origin에 저장한 text를 예상 KEY_LENGTH (i) 번째 글자마다의 alphabet 사용 횟수를 저장해주었습니다. sum은 총 alphabet 사용 횟수입니다.

```
|float s = 0;$
|for(int j=0; j<26; j++){      // alphabet의 빈도수를 체크$
|^|al[j] = (float) alp[j] / (float) sum;$
|^|al[j] = al[j] * al[j];$
|^|s = s + al[j];$
|^|al[j]=0;$
|^|alp[j]=0;$
|}$
|$
|if(sum > 300 && s < 0.07 && s > 0.06) return i;  //빈도수 합이 0.065에 가까우면 length이므로 return$
|sum = 0;$
```

이후 al array에 (alphabet 마다의 사용 횟수) / (총 alphabet 사용 횟수) = 빈도 를 저장해 주었고, 제곱하였습니다. s에 빈도 제곱 값을 총합을 저장하고, 만약 0.065에 가까운 값이 나오면 빠른 해독을 위해 바로 return하여 함수를 끝내도록 하였습니다.

만약 sum이 300보다 작다면 표본이 너무 적다고 판단되기에 return하지 않도록 하였습니다.

```
^|^|pp[i-1] = s;$
^|}$
$
^|float y = 0.038;$
^|int ans;$
^|for(int i=0; i<10; i++){$
^|^|if(pp[i] > y){$
^|^|^|y = pp[i];$
^|^|^|ans = i+1;$
^|^|}$
^|}$
^|return ans;  //만약 빈도수가 0.065에 가까운 경우가 없으면 가장 큰 경우를 length로 판단$
}$
$
```

만약 0.065에 가까운 값이 없으면 pp array에 빈도 수 합을 저장하고 for문이 끝나면 1 부터 10까지의 빈도 수 중 가장 크게 나온 값을 KEY_LENGTH로 정하게 하였습니다.

다음으로 find_key(int)은 아래와 같습니다.

```
void find_key(int k){$
^|for(int i=0; i<k; i++){
^|^|key[i] = find_value(k,i);$
^|}$
}$
```

KEY_LENGTH만큼 for문을 돌며 각 위치에 맞는 value를 찾기 위해 find_value (int, int) 함수를 사용하게 됩니다. 처음 넘겨주는 인자는 length의 길이이고, 두번째 인자는 몇 번째의 value 인지를 넘겨줍니다.

다음으로 KEY_VALUE 값을 찾는 함수인 find_value(int)에 대해 설명하겠습니다.

```
int find_value(int length, int position){           // key 개수를 알았다면, 위치마다의 value값 찾기$
^|float val[256]={0,};                  // 0x00 ~ 0xff 변환 후 빈도수 저장$
^|float e[256]={0,};                    // 0x00 ~ 0xff 변환 후 e 사용 횟수 저장$
$
^|for(int i=0; i<256; i++){$
^|^|int cnt = position;$
^|^|int sum = 0;$
^|^|int ccnt = 0;$
^|$
^|^|while(cnt < text){              // 0x00 ~ 0xff 중 for문을 반복하며 변환 값을 temp에 저장$
^|^|^|temp[ccnt] = origin[cnt] ^ i;$
^|^|^|ccnt++;$
^|^|^|cnt = cnt + length;$
^|^|}$
^|^|temp[ccnt]='₩0';$
$
^|^|for(int j=0; j<ccnt; j++){              // 변환 후 alphabet 횟수 저장$
^|^|^|if('a' <= temp[j] && temp[j] <= 'z'){$
^|^|^|^|alp[temp[j]-97]++;$
^|^|^|^|sum++;$
^|^|^|}$
^|^|}$
```

우선 초반 부분은 find_length() 함수와 비슷하게 진행됩니다. 다른 점은 temp에 position 번째의 글자를 시작으로 length마다의 글자를 i(0x00~0xff)와 xor한 값을 저장하게 하였습니다. 그 후 find_length()와 동일하게 alphabet 사용횟수와 총 alphabet 사용횟수를 저장하게 해주었습니다.

새로운 array인 val은 0x00~0xff까지의 값에 따른 빈도수 합을 저장하기 위해 사용하였고, e는 alphabet 'e'의 사용 횟수를 저장하기 위해 사용되었습니다.

```
float s = 0;$
e[i] = (float) alp[4] / (float) sum;       // (e 사용 횟수)/(alphabet 총 횟수) 저장$
if(e[i] > 0.5) e[i]=0;                    // 빈도수가 지나칠 정도로 많으면 잘못된 것으로 판단하여 0으로 저장
for(int j=0; j<26; j++){      $
^|al[j] = (float) alp[j] / (float) sum;$
^|al[j] = al[j] * al[j];$
^|s = s + al[j];$
^|al[j]=0;$
^|alp[j]=0;$
}$
sum = 0;$
val[i] = s;                         // 변환 후 빈도수 저장$
if(val[i] > 0.09) e[i]=0;           // 빈도수가 과도할 정도로 높으면 잘못된 것으로 판단$

cnt = 0;$
while(cnt < ccnt){        // 변환 전으로 다시 변환$
^|temp[cnt] ^= i;$
^|cnt++;$
}$
```

그 후 e array에 alphabet 'e'의 사용 빈도를 저장하였습니다. 만약 0.5보다 클 경우는 과도하게 나왔기 때문에 표본이 부족하여 생긴 결과일 수도 있으므로 0으로 바꾸었습니다. 이후엔 (alphabet 사용 횟수) / (총 alphabet 사용 횟수) = 빈도 를 구한 후 빈도의 총합을 제곱하여 저장해주었습니다. 여기서는 val array에 저장하였습니다. 만약 빈도 총합이 0.09 초과하면 이것 또한 표본의 문제로 발생한 것이라 판단해 e array에 저장한 값을 0으로 하였습니다.

그 후 다시 temp에 변형 전 상태를 저장하였습니다.

```
^|int ans=0;$
^|float max=0.05;$
^|for(int i=0; i<256; i++){ // 변환 된
^|^|$
^|^|if(e[i] > max && val[i] < 0.08){$
^|^|^|if(val[i] < 0.05) continue;$
^|^|^|max = e[i];$
^|^|^|ans = i;$
^|^|}$
^|}$
^|return ans;$
}$
```

마지막으로 for문 (0x00 ~ 0xff) 까지 빈도수 제곱 합 'e' 사용 빈도 측정이 끝나면, 전체의 값을 다시 비교하여 KEY_VALUE를 찾는 과정입니다.

제가 선택한 value는 'e'의 사용횟수가 가장 크고, 빈도수 제곱 합이 0.05~0.08 안에 들어오면 적절하다고 생각하였습니다.

제 code에 대해 평가하자면 처음 입력되는 plaintext의 크기가 1000 bytes에서 5000 bytes 사이의 값으로 제한되어 있기에 사용된 text에 따라 표본이 부족하여 정확도가 부족하다고 생각합니다.

실행화면을 끝으로 설명 마치겠습니다.

```
Sixteen years had Miss Taylor been in Mr. Woodhouse's family, less as a$
governess than a friend, very fond of both daughters, but particularly$
of Emma. Between _them_ it was more the intimacy of sisters. Even before$
Miss Taylor had ceased to hold the nominal office of governess, the$
mildness of her temper had hardly allowed her to impose any restraint;$
and the shadow of authority being now long passed away, they had been$
living together as friend and friend very mutually attached, and Emma$
doing just what she liked; highly esteeming Miss Taylor's judgment, but$
directed chiefly by her own.$
$
The real evils, indeed, of Emma's situation were the power of having$
rather too much her own way, and a disposition to think a little too$
well of herself; these were the disadvantages which threatened alloy to$
her many enjoyments. The danger, however, was at present so unperceived,$
that they did not by any means rank as misfortunes with her.$
$
Sorrow came--a gentle sorrow--but not at all in the shape of any$
disagreeable consciousness.--Miss Taylor married. It was Miss Taylor's$
loss which first brought grief. It was on the wedding-day of this$
beloved friend that Emma first sat in mournful thought of any$
continuance. The wedding over, and the bride-people gone, her father and$
herself were left to dine together, with no prospect of a third to cheer$
a long evening. Her father composed himself to sleep after dinner, as$
usual, and she had then only to sit and think of what she had lost.$
$
The event had every promise of happiness for her friend. Mr. Weston$
was a man of unexceptionable character, easy fortune, suitable age, and$
pleasant manners; and there was some satisfaction in considering$
with what self-denying, generous friendship she had always wished and$
promoted the match; but it was a black morning's work for her. The want$
of Miss Taylor would be felt every hour of every day. She recalled her$
past kindness--the kindness, the affection of sixteen years--how she had$
taught and how she had played with her from five years old--how she had$
devoted all her powers to attach and amuse her in health--and how$
nursed her through the various illnesses of childhood. A large debt of$
gratitude was owing here; but the intercourse of the last seven$
years, the equal footing and perfect unreserve which had soon followed$
Isabella's marriage, on their being left to each other, was yet a$
dearer, tenderer recollection. She had been a friend and companion such$
as few possessed: intelligent, well-informed, useful, gentle, knowing$
all the ways of the family, interested in all its concerns, and$
peculiarly interested in herself, in every pleasure, every scheme of$
```

```c
#include <stdio.h>
#define KEY_LENGTH 5
int main(){
    unsigned char ch;
    FILE *fpIn, *fpOut;
    unsigned char key[KEY_LENGTH]={ 0x45, 0x01, 0x22, 0x02, 0x03};
    fpIn = fopen("plaintext.txt", "r");
    fpOut = fopen("hw1_input.txt", "wb");
    for(int i=0; fscanf(fpIn, "%c", &ch) != EOF; ++i){
        ch ^= key[i % KEY_LENGTH];
        fwrite(&ch, sizeof(ch), 1, fpOut);
    }
    fclose(fpIn);
    fclose(fpOut);
    return 0;
}
```

암호화 code

암호화된 text 입니다.

```
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ ls
1.c  2016310936_hw1.c  plaintext.txt
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ vi plaintext.txt
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ gcc 1.c
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ ./a.out
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ ls
1.c  2016310936_hw1.c  a.out  hw1_input.txt  plaintext.txt
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ gcc 2016310936_hw1.c
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ ./a.out
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$ ls
1.c  2016310936_hw1.c  a.out  hw1_input.txt  hw1_output.txt  plaintext.txt
saumsung@DESKTOP-IC1EI9F:/mnt/c/Users/wsm42/Desktop/4-1/정보개/1$
```

실행 화면입니다.

해독한 text 입니다.

```
0x45 0x01 0x22 0x02 0x03 $
Sixteen years had Miss Taylor been in Mr. Woodhouse's family, less as a$
governess than a friend, very fond of both daughters, but particularly$
of Emma. Between _them_ it was more the intimacy of sisters. Even before$
Miss Taylor had ceased to hold the nominal office of governess, the$
mildness of her temper had hardly allowed her to impose any restraint;$
and the shadow of authority being now long passed away, they had been$
living together as friend and friend very mutually attached, and Emma$
doing just what she liked; highly esteeming Miss Taylor's judgment, but$
directed chiefly by her own.$
$
The real evils, indeed, of Emma's situation were the power of having$
rather too much her own way, and a disposition to think a little too$
well of herself; these were the disadvantages which threatened alloy to$
her many enjoyments. The danger, however, was at present so unperceived,$
that they did not by any means rank as misfortunes with her.$
$
Sorrow came--a gentle sorrow--but not at all in the shape of any$
disagreeable consciousness.--Miss Taylor married. It was Miss Taylor's$
loss which first brought grief. It was on the wedding-day of this$
beloved friend that Emma first sat in mournful thought of any$
continuance. The wedding over, and the bride-people gone, her father and$
herself were left to dine together, with no prospect of a third to cheer$
a long evening. Her father composed himself to sleep after dinner, as$
usual, and she had then only to sit and think of what she had lost.$
$
The event had every promise of happiness for her friend. Mr. Weston$
was a man of unexceptionable character, easy fortune, suitable age, and$
pleasant manners; and there was some satisfaction in considering$
with what self-denying, generous friendship she had always wished and$
promoted the match; but it was a black morning's work for her. The want$
of Miss Taylor would be felt every hour of every day. She recalled her$
past kindness--the kindness, the affection of sixteen years--how she had$
taught and how she had played with her from five years old--how she had$
devoted all her powers to attach and amuse her in health--and how$
nursed her through the various illnesses of childhood. A large debt of$
gratitude was owing here; but the intercourse of the last seven$
years, the equal footing and perfect unreserve which had soon followed$
Isabella's marriage, on their being left to each other, was yet a$
dearer, tenderer recollection. She had been a friend and companion such$
as few possessed: intelligent, well-informed, useful, gentle, knowing$
all the ways of the family, interested in all its concerns, and$
```