Title: whitebox_AES

Description: Find the key in the Tbox table!

인하대 IXPLOIT 남동현

Write-up

analyze

White-Box AES의 라운드는 먼저 ShiftRows 연산을 한 후, 나머지 세 연산을 가능한 모든 경우에서 연산 해놓은 T-Boxes들로부터 적당한 값을 취하는 방식이다.

strategy

> 1. tbox 배열에서 우선 r번째 라운드에서, 아무 byte x를 정하고, 해당 라운드 키를 구하기위해 16byte값을 모두 모은다.
> 2. 구한 16byte들을 각각 sbox에서 찾아 행과 열을 이용해 16byte를 구하고, x와 xor하여 tbox에 들어오기 전 16byte를 구한다.
> 3. 2에서 구한 16byte를 주어진 ShiftRows(sr)함수에 3번 적용시킨다. (sr의 역함수)
> 4. Key Schedule 알고리즘을 거슬러가며 최초 키를 구한다.

implementation

편의상 키 스케줄링 알고리즘을 거슬러 올라갈 필요가 없는 0번째 라운드 키를 사용하였다.

```
#include "wbAES.h"
#include <stdio.h>
#include <stdint.h>

uint8_t reverse_sbox(uint8_t a){
    for (int i = 0; i < 16; i++){
        for (int j = 0; j < 16; j++){
            if (sbox[i][j] == a) return i*0x10+j;
        }
    }
}
void reverse_sr(unsigned char out[16]){     // sr(sr(sr(sr(a)))) = a
    for (int i = 0; i < 3; i++) sr(out);
    }

int main(){
```

```
    uint8_t key[16] = {};
    uint8_t x = 0x00;
    for(int i = 0; i < 16; i++){
        key[i] = reverse_sbox(tbox[0][i][x])^x;     // collect 0th round key's ith
byte( xor wt x)
    }
    reverse_sr(key);
    for(int i = 0; i < 16; i++){
        printf("%c", key[i]);
    }
    printf("\n");

}
```

key (= flag)

```
INCO{0buc4t3d?}
```