2021 암호경진대회

6번 문제 : 암호구현

Timing Attack은 비밀정보에 의존하는 암호화 연산 수행 시 발생하는 시간 정보를 이용하여 비밀정보를 추출하는 공격 기법을 의미한다. 대칭키에 대한 Timing Attack을 방어하기 위해서는 비트슬라이싱 기법을 통해 시간 의존적 암호화 연산을 제거하는 것이 중요하다. 아래에는 비트슬라이싱이 적용되지 않은 대칭키 암호화에 대한 레퍼런스 코드가 제시되어 있다. 해당 암호에 비트슬라이싱 기법을 적용하고 8-비트 저전력 프로세서에 맞게 최적화 고속구현하시오.

```
#pragma GCC optimize ("-O3")

typedef unsigned char    u8;
typedef unsigned short   u16;
//typedef unsigned int   u32;

// round: #128
#define NUM_ROUND    128

// key_gen constant
#define CONST_VAL    0xab00cd00

// rotation left
#define ROL(X) ((X<<1) | (X>>31))

// s_box: 4-bit x #16
u32 s_box[16] = {0x3, 0x9, 0x6, 0xf, 0xe, 0x5, 0xd, 0x4, 0xc, 0x7, 0xa, 0x2, 0xb, 0x1, 0x8, 0x0};

// key generation
void new_bs_keygen(u32* r_key, u32 m_key){
  int i=0;
  u32 constant_value = CONST_VAL;

  for(i=0;i<NUM_ROUND;i++){
    r_key[i] = (m_key + i) ^ constant_value;
    constant_value = ROL(constant_value);
  }
}

u32 s_box_gen(u32 text){
  u32 output = 0;
  u32 temp = 0;
  int i = 0;

  for(i=0;i<8;i++){
    temp = ( ( text >> (4*i) ) & 0x0000000F );
    temp = s_box[temp];
```

```
    output = (output | ( temp << (4*i) ) );
  }

  return output;
}

u32 p_box1_gen(u32 text){
  u32 p_idx[32] = {0, 8, 16, 24, 1, 9, 17, 25, 2, 10, 18, 26, 3, 11, 19, 27, 4, 12, 20, 28, 5, 13, 21,
29, 6, 14, 22, 30, 7, 15, 23, 31};
  u32 output = 0;
  u32 temp = 0;
  int i = 0;

  for(i=0;i<32;i++){
    temp = ((text>>i) & 0x00000001);
    temp = (temp << p_idx[i]);
    output = (output | temp);
  }

  return output;
}

u32 p_box2_gen(u32 text){
  u32 p_idx[32] = {27, 1, 23, 30, 7, 22, 29, 16, 0, 4, 13, 18, 25, 17, 28, 31, 10, 14, 3, 5, 6, 2, 12,
11, 9, 8, 19, 26, 24, 20, 15, 21};
  u32 output = 0;
  u32 temp = 0;
  int i = 0;

  for(i=0;i<32;i++){
    temp = ((text>>i) & 0x00000001);
    temp = (temp << p_idx[i]);
    output = (output | temp);
  }

  return output;
}

void new_bs_enc(u32* r_key, u32* text){
  u32* r_key_in  = r_key;
  u32 text_in = *text;
  int i;

  for(i=0;i<NUM_ROUND;i++){
    text_in = s_box_gen(text_in);
    text_in = p_box1_gen(text_in);
    text_in = text_in^r_key_in[i];
```

```
    text_in = p_box2_gen(text_in);
  }
  *text = text_in;
}


u8 TEST_VECTOR(u32 in, u32 answer) {
  return (in == answer);
}

void setup() {
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);

  // r_key : 32-bit x #128
  u32 r_key[128] = {0, };

  // m_key : 32-bit
  u32 m_key[3]    = {0x12345678,0x01020304,0x55667788};

  // text: 32-bit
  u32 text[3]     = {0x90ABCDEF,0x0A0B0C0D,0xFFEEDDCC};
  u32 out_text[3]= {0xE4DE2FF8,0xE7F54BDC,0x53485E4F};

  Serial.println("-----------------");
  Serial.println("    TEST VECTOR ");
  Serial.println("-----------------");

  for(int i=0; i<3; i++) {
    new_bs_keygen(r_key, m_key[i]);
    new_bs_enc(r_key, &text[i]);

    if(TEST_VECTOR(text[i], out_text[i])){
      Serial.println(">> CORRECT");
    }else{
      Serial.println(">> WRONG");
    }
  }
  Serial.println("-----------------");
  Serial.println("     BENCHMARK ");
  Serial.println("-----------------");

  // m_key : 32-bit
  u32 m_key_bench = 0x12345678;

  // text: 32-bit
  u32 text_bench  = 0x90ABCDEF;
```

```
  u32 time1;
  u32 time2;
  time1 = millis();
  for(int i=0; i<64; i++) {

    new_bs_keygen(r_key, m_key_bench);
    new_bs_enc(r_key, &text_bench);
  }

  time2 = millis();
  Serial.print(">> ");
  Serial.println((time2-time1));
  Serial.println("-----------------");
}

void loop() {
}
```

주의사항

1) 구현 타겟 플랫폼은 아두이노 우도 (8-비트 AVR 프로세서)이며 상세한 정보는 웹사이트를 참고하도록 한다.
https://store.arduino.cc/usa/arduino-uno-rev3

2) 제시된 레퍼런스 코드에서 두 함수 (new_bs_keygen 그리고 new_bs_enc)의 내부만을 수정하도록 한다.
(두 함수의 입력 인자와 출력 인자의 수와 형식에 대한 변경도 허용하지 않는다.).
그 외의 코드에 대한 수정은 허용하지 않는다. 단 개발 용이성을 위한 새로운 함수 추가는 허용한다.

3) 인라인 어셈블리 혹은 어셈블리 코드로 프로그래밍하는 경우에도 두 함수 (new_bs_keygen 그리고 new_bs_enc)의 내부만을 수정하도록 한다.

4) 비트슬라이싱 기법을 활용하여 S-BOX 연산을 수행해야 한다. 그렇지 않은 경우 0점이 부여된다.

5) 프로그래밍은 Arduino IDE를 사용하도록 한다. 결과물은 (*.ino) 형식만을 허용한다.
https://www.arduino.cc/en/main/software

6) 결과물은 다음 2종을 포함한다.
- 아두이노 코드 (테스트 벡터 확인 과정, 벤치마크 과정)
- 문서 (구현 기법 상세, 테스트 벡터 확인 결과, 벤치마크 결과)

7) 평가방법은 다음과 같다.
- 테스트 벡터 통과 (30점, 절대 평가)
- 문서화 (30점, 절대평가)
- 벤치마크 결과 (40점, 상대평가)