2020 암호분석경진대회

2번 문제 : 암호구현 및 최적화

경량 사물인터넷 환경 상에서의 안전하고 효율적인 대칭키 암호 구현은 사물인터넷 서비스의 보안성과 가용성 확보를 위해 매우 중요하다. 아래에는 Addition Rotation eXclusive-or (ARX)를 기반으로 한 대칭키 암호에 대한 레퍼런스 코드가 기술되어 있다. 해당 암호를 8-비트 사물인터넷 환경에 맞게 최적화 구현하시오.

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

#define ROUND_NUM 128

typedef unsigned char u8;
typedef unsigned short u16;

u8 ROL8(u8 x, u8 n) {
  return ((u8) ((x) << (n)) | (u8) ((x) >> (8 - (n))));
}

u16 ROL16(u16 x, u16 n) {
  return ((u16) ((x) << (n)) | (u16) ((x) >> (16 - (n))));
}

void key_gen(u8 *rnd, u8 *key) {
  u8 key_in[2];

  u8 tmp1, tmp2;
  u16 *key_p;
  u16 con = 0x9ABC;

  key_in[0] = key[0];
  key_in[1] = key[1];

  key_p = (u16*) key_in;

  int i;
  for (i = 0; i < ROUND_NUM; i++) {

    if (i % 2 == 0) {
      key_in[0] = ROL8(key_in[0], 1) + ROL8(key_in[0], 5);
      key_in[1] = ROL8(key_in[1], 3) + ROL8(key_in[1], 7);
    } else {
      *key_p = ROL16(*key_p, 1) + ROL16(*key_p, 9) + ROL16(con, (i%16));
    }

    tmp1 = key_in[0] + key_in[1];
    tmp2 = key_in[0] ^ key_in[1];
```

```
    key_in[0] = tmp1;
    key_in[1] = tmp2;

    rnd[i * 2 + 0] = key_in[0];
    rnd[i * 2 + 1] = key_in[1];
  }
}

void enc(u8 *text, u8 *rnd) {
  u8 text_in[2];

  u16 *text_p;

  text_in[0] = text[0];
  text_in[1] = text[1];

  text_p = (u16*) text_in;

  int i;
  for (i = 0; i < ROUND_NUM; i++) {
    if (i % 2 == 0) {
      *text_p = ROL16(*text_p, 4);
    } else {
      *text_p = ROL16(*text_p, 8);
    }
    text_in[0] = text_in[0] + rnd[i * 2 + 0];
    text_in[1] = text_in[1] ^ rnd[i * 2 + 1];
  }
  text[0] = text_in[0];
  text[1] = text_in[1];
}

u8 TEST_VECTOR(u8 *in, u8 *answer) {
  return (in[0] == answer[0] && in[1] == answer[1]);
}

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

  u8 key[3][2]       = { {0x12, 0x34}, {0x9A, 0xBD}, {0x11, 0x22} };
  u8 rnd[ROUND_NUM * 2] = { 0, };
  u8 text[3][2]      = { {0x56, 0x78}, {0xDE, 0xF0}, {0x33, 0x44} };
  u8 out_text[3][2] = { {0x50, 0x3F}, {0x88, 0x28}, {0x7F, 0x33} };

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

```
  Serial.println("-----------------");

  for(int i=0; i<3; i++) {
    key_gen(rnd, key[i]);
    enc(text[i], rnd);

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

  u8 key_bench[2]      = { 0x12, 0x34 };
  u8 text_bench[2]     = { 0x56, 0x78 };

  u32 time1;
  u32 time2;
  time1 = millis();
  for(int i=0; i<4096; i++) {
    key_gen(rnd,key_bench);
    enc(text_bench,rnd);

    if(text_bench[0]>0x80){
      digitalWrite(LED_BUILTIN, HIGH);
    }else{
      digitalWrite(LED_BUILTIN, LOW);
    }
  }

  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 (key_gen 그리고 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 프로그래밍 하는 경우에도 두 함수 (key_gen 그리고 enc)의 내부만을 수정하도록 한다.

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

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

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