# 2020 암호분석경진대회 답안제출

2020. 08. 31

| | | |
|---|---|---|
| **참가자 1** | **성 명** | 이창원 |
| | **소속** | 서울시립대학교 수학과 |
| | **휴대폰** | |
| | **E-mail** | |

| | | |
|---|---|---|
| **참가자 2** | **성 명** | 장호빈 |
| | **소속** | 서울시립대학교 수학과 |
| | **휴대폰** | |
| | **E-mail** | |

| | | |
|---|---|---|
| **참가자 3** | **성 명** | 김인성 |
| | **소속** | 서울시립대학교 수학과 |
| | **휴대폰** | |
| | **E-mail** | |

# 3번 문제 답안

## 정답:

Recommendations for preventing novel coronavirus infection

1. Wash your hands with soap and running water.
2. Cover your mouth with your sleeve when coughing!
3. If you experience respiratory symptoms such as cough, must wear a mask.
4. Inform medical staffs of your travel history when visiting selected clinics.
5. Consult with your local public health center or call '1339' or 'Are a Code+120' if you are suspicious of contract an infectious disease.

## 풀이:

ARIA-128을 COA에 대해 바로 복호화하는 것은 어렵다. 그래서 블록암호 운영모드가 적용될 때 사용되는 패딩 기법을 통해 공격을 진행할 수 있고 이때 쓰이는 공격 방법은 oracle padding attack 이다.

암호문 C를 16바이트씩 29개의 블록으로 나누어 각각 $C_0$, $C_1$, ..., $C_{28}$이라고 하자. n번째 암호문 $C_n$을 복호화하여 나온 값을 $M_n$이라고 할 때, $D_k(C_n)=M_n$이다. 그러면 첫 번째 16바이트 블록에서 $P_0=M_0 \oplus IV$ 이 되고 우리가 이 처음 16바이트 평문 블록만 암호화 되며 맨 끝 한자리가 패딩 되었다고 가정하면 $IV[15] \oplus M_0[15]=0x80$ 이 된다.

여기서 IV[15]의 값을 주어진 패딩이 올바른지 확인하는 API를 사용하여 0x00부터 0xFF까지 $2^8$번의 전수조사를 하게 되면 특정한 값에서 올바른 패딩이 되었다고 출력이 된다. 이 문제에서는 0x82가 이를 만족하였고 이에 따라 $0x82 \oplus M_0[15]=0x80$임을 알 수 있어 $M_0[15]$가 0x02임을 알 수 있다.

이제 맨 끝 한자리가 아닌 두 번째 자리까지 패딩이 되었다고 가정하자. 패딩 방식이 처음엔 0x80을 붙이고 그 이후에는 0x00을 부족한 만큼 채우는 방식이기 때문에 맨 끝자리는 올바르게 패딩이 될 수 있도록 IV[15]의 값을 $M_0[15]$와 xor 했을 때 0x00이 되도록 0x02로 바꾸고 IV[14]값에 대해 위와 마찬가지로 0x00부터 0xFF까지 조사를 하면 마찬가지로 어느 특정 값에서 올바른 패딩 값이 나오게 되는 것을 알 수 있다. 그리고 이를 통하여 위와 마찬가지로 $M_0[14]$의 값을 알 수 있다. 이 과정을 반복하면 첫 번째 16바이트 블록의 $M_0$을 알 수 있고 원래의 IV값과 xor 연산을 하면 평문 $P_0$을 알 수 있게 된다.

n 번째 블록(n>1)부터는 암호화 과정에서 원래의 IV가 이전 블록에서 사용된 평문과 그것을 암호화한 암호문을 xor 한 값인 $G_{n-1}$을 사용하기 때문에 위의 과정 중 마지막에 원래의 IV값과 xor 하는 것 대신 $G_{n-1}$을 xor하여 평문을 구한다. 이를 반복하여 $P_0$부터 $P_{28}$까지 구할 수 있고 맨 마지막 $P_{28}$ 블록에서 5번째 바이트부터 80||00||…||00 이 등장하여 패딩이 되었음을 알 수 있고 올바른 평문을 구했음을 알 수 있다.

이렇게 $P_0$부터 $P_{28}$에 해당하는 16바이트 블록 내의 데이터를 ASCII 코드를 통해 영어로 변환하면 위와 같은 결과를 얻을 수 있다.

이에 대한 알고리즘은 아래와 갔다.

# 3번 문제 답안

Oracle Padding attack algorithm

1. ciphertext C, IV
2. C를 29개의 16byte 크기의 블록으로 나눈다. ($C_0$, $C_1$, …, $C_{28}$)
3. G=IV
4. for r=0 to r=28
    for i=0 to i=15
        for j=0 to j=255
            T[15 - i] = j
            while(i != 0)
                i = i - 1
                T[15 - i] = $M_r$[15 - I]  // 패딩을 맞추기 위해 타겟인 1바이트 뒷부분은 0으로 만듦
            v = API($C_r$, T)
            if(v==0)
                $M_r$[15 - i] = 0x80 ⊕ T[15 - i]
                break
    for i=0 to i=15
        $P_r$[i]=G[i] ⊕ $M_r$[i]
        G[i]=$P_r$[i] ⊕ $C_r$[i]

// API($C_R$,IV) : $C_R$과 IV를 검사하여 올바르게 패딩이 되었는지 확인하는 함수(결과값 - 0: valid, 1: invalid)