# 2018 암호분석경진대회 : 4번 문제 답안

**비밀 키 d :**

$(100100000000111001000001011100110000001011010011011110011101101111110001010110100000001000100011010 1)_2$

**개요 :**

1. 주어진 key_sequences를 분석해보면 모든 AD수열의 앞부분은

'DDADDADADADDDADDDDADDDDDDDDADDADAD' 또는 'DDADDADADDDDADDDDADDDDDDDDADDADAD'

으로 이루어져 있음을 알 수 있다. 앞부분이 2가지로만 생겨난다는 사실을 이용해 비밀 키의 일부분을 유추할 수 있다. 그리고 그 과정을 통해 무한원점과 연산되어 실제 연산을 하지 않는 부분이 어디인지 알 수 있다. 이러한 정보들과 함께 아래와 같은 순서로 비밀 키 후보들을 찾는다.

2. key_sequences에서 주어진 AD수열과 state를 이용해 50개의 AD수열에서 각각 나올 수 있는 키 후보들을 찾는다.
3. 2번에서 찾은 50묶음의 키 후보에서 가장 많이 나온 후보들을 추린다.
4. 3번의 키 후보들 중에서 50개의 AD수열이 나올 수 있는지 확인하여 비밀 키를 찾는다.

**풀이 :**

**1. 주어진 AD수열을 이용해 비밀 키의 일부분을 찾는다.**

주어진 AD수열의 앞부분이 모두

'DDADDADADADDDADDDDADDDDDDDDADDADAD' 또는 'DDADDADADDDDADDDDADDDDDDDDADDADAD'

으로 이루어져 있다. 이것을 좀 더 자세히 분석해 보면 10번째에서만 달라진다.

'DDADDADAD ADDD ADDDDADDDDDDDDADDADAD'
'DDADDADAD DDD ADDDDADDDDDDDDADDADAD'

AD수열과 state를 사용해 키의 끝을 00,01,10,11로 두고 하나씩 Algorithm 1을 수행해보자.

| | | | | |
|---|---|---|---|---|
| ①<br>비밀 키가<br>'00'으로<br>끝나는 경우 | | $S[1] \leftarrow O$ | $S[2] \leftarrow P$ | $state = 0$ |
| | $d_0 = 0$ | | $S[2] \leftarrow 2S[2] (= 2P)$ | $state = 0$ |
| | $d_1 = 0$ | | $S[2] \leftarrow 2S[2] (= 4P)$ | $state = 0$ |
| | $d_2 = 1$ | $S[1] \leftarrow S[1] + S[2] (= O + 4P)$ | $S[2] \leftarrow 2S[2] (= 8P)$ | |

표와 같이 $d_2 = 1$ 일 때, 무한원점과의 연산은 수행되지 않아 AD수열로 표현하면 DDD가 된다.

$d_2 = 0$ 인 경우도 마찬가지로 DDD가 됨을 쉽게 확인할 수 있다. 따라서 비밀 키가 00으로 끝나는 경우에는 key_sequences안에 있는 AD수열을 만들어 낼 수 없다. 즉, 비밀 키는 00으로 시작하지 않는다.

# 2018 암호분석경진대회 : 4번 문제 답안

| | | | | |
|---|---|---|---|---|
| ②<br>비밀 키가<br>'11'으로<br>끝나는 경우 | | $S[1] \leftarrow O$ | $S[2] \leftarrow P$ | $state = 0$ |
| | $d_0 = 1$ | $S[1] \leftarrow S[1] + S[2] (= O + P)$ | $S[2] \leftarrow 2S[2] (= 2P)$ | $state = 1$ |
| | $d_1 = 1$ | $S[1] \leftarrow S[1] + S[2] (= P + 2P)$ | $S[2] \leftarrow 2S[2] (= 4P)$ | |

비밀 키가 11으로 끝나는 경우에는 (A)DAD가 되어 key_sequences에 있는 AD수열을 만들 수 없다.

| | | | | |
|---|---|---|---|---|
| ③<br>비밀 키가<br>'10'으로<br>끝나는 경우 | | $S[1] \leftarrow O$ | $S[2] \leftarrow P$ | $state = 0$ |
| | $d_0 = 0$ | | $S[2] \leftarrow 2S[2] (= 2P)$ | $state = 0$ |
| | $d_1 = 1$ | $S[1] \leftarrow S[1] + S[2] (= O + 2P)$ | $S[2] \leftarrow 2S[2] (= 4P)$ | $state = 1$ |
| | $d_2 = 1$ | $S[1] \leftarrow S[1] + S[2] (= P + 4P)$ | $S[2] \leftarrow 2S[2] (= 8P)$ | $state = 1,11$ |

비밀 키가 10으로 끝나는 경우에는 $d_2 = 1$일 때, D(A)DAD가 만들어지지만 $d_2$에서 state가 1과 11로 나뉘면서 $d_3$가 어떤 것이 되더라도 key_sequences안에 AD수열 앞부분이 2가지 외에 더 많이 생긴다. 따라서 비밀 키가 00, 10, 11로 끝나게 되면 key_sequences 정보에 어긋나게 되어 비밀 키는 01로 끝나게 됨을 알 수 있다.

비밀 키가 01로 끝남을 알았으니 AD수열을 이용해 비밀 키의 일부를 유추해보자.

'DDADDADAD DDD ADDDDADDDDDDDDADDADAD'
'DDADDADAD ADDD ADDDDADDDDDDDDADDADAD'

| | | | | |
|---|---|---|---|---|
| | $S[1] \leftarrow O$ | $S[2] \leftarrow P$ | $state = 0$ | |
| $d_0 = 1$ | $S[1] \leftarrow S[1] + S[2] (= O + P)$ | $S[2] \leftarrow 2S[2] (= 2P)$ | $state = 1$ | |
| $d_1 = 0$ | | $S[2] \leftarrow 2S[2] (= 4P)$ | $state = 0$ | |
| $d_2 = 1$ | $S[1] \leftarrow S[1] + S[2] (= P + 4P)$ | $S[2] \leftarrow 2S[2] (= 8P)$ | $state = 1$ | |
| $d_3 = 0$ | | $S[2] \leftarrow 2S[2] (= 16P)$ | $state = 0$ | |
| $d_4 = 1$ | $S[1] \leftarrow S[1] + S[2]$ | $S[2] \leftarrow 2S[2]$ | $state = 1$ | |
| $d_5 = 1$ | $S[1] \leftarrow S[1] + S[2]$ | $S[2] \leftarrow 2S[2]$ | $state = 1,11$ | e=1 or e=0 2가지 경우로 나뉜다. |

2가지 수열 모두 'DDADDADAD'로 시작하기 때문에 $d_4$까지는 $state$가 1가지로 유지되어야 한다. 그 다음에는 'ADDD' or 'DDD'로 나뉘기 때문에 $d_5$에서 $state$가 2가지로 나뉠 수 있도록 $d_5 = 1$을 잡아준다.

# 2018 암호분석경진대회 : 4번 문제 답안

e=1

| | | | |
|---|---|---|---|
| $d_6 = 0$ | | $S[2]$ | $state = 0$ |
| $d_7 = 0$ | | $S[2]$ | $state = 0$ |
| $d_8 = 0$ | | $S[2]$ | $state = 0$ |
| $d_9 = 1$ | $S[1]$ | $S[2]$ | $state = 1$ |
| $d_{10} = 0$ | | $S[2]$ | $state = 0$ |
| $d_{11} = 0$ | | $S[2]$ | $state = 0$ |
| $d_{12} = 0$ | | $S[2]$ | $state = 0$ |
| $d_{13} = 1$ | $S[1]$ | $S[2]$ | $state = 1$ |
| $d_{14} = 0$ | | $S[2]$ | $state = 0$ |
| $d_{15} = 0$ | | $S[2]$ | $state = 0$ |

e=0

| | | |
|---|---|---|
| $S[1]$ | $S[2]$ | $state = 1$ |
| | $S[2]$ | $state = 0$ |
| | $S[2]$ | $state = 0$ |
| $S[1]$ | $S[2]$ | $state = 1$ |
| | $S[2]$ | $state = 0$ |
| | $S[2]$ | $state = 0$ |
| | $S[2]$ | $state = 0$ |
| $S[1]$ | $S[2]$ | $state = 1$ |
| | $S[2]$ | $state = 0$ |
| | $S[2]$ | $state = 0$ |

위와 같이 $state$ 와 AD수열을 종합하여 비밀 키의 일부분을 유추할 수 있고, 비밀 키의 뒷부분은 … 010001000110101 로 끝내게 된다.

이는 비밀 키의 빨간색 부분을 찾은 것이다.

$(1001000000001110010000010111001100000001011010011011110011101101111110001010110100000001000100011010 1)_2$

# 2018 암호분석경진대회 : 4번 문제 답안

## 2. key_sequences에서 주어진 AD수열과 state를 이용해 50개의 AD수열에서 각각 나올 수 있는 키 후보들을 찾는다.

state와 AD수열을 종합해 비트가 0이었는지 1이었는지 유추할 수 있다. 그것들을 종합해 코드로 구현하고 실행시키면 가능한 키수열 후보들을 얻을 수 있다.

ex. state가 0일 때,

```
if state==0:
        if arr[i][j]=='D':
            state=0
            String+='0'
            j+=1
        else:#arr[i][j]=='A'
            state=1
            String+='1'
            j+=2
```

문제에 주어진 Algorithm 1을 보면 키수열의 비트가 0이면서 state가 0일 때 'D'가 출력되고, 비트가 1이면서 state가 0일 때 'AD'가 출력됨을 알 수 있다. 그러므로 state가 0이고 그때 AD수열의 상태가 'D'이면, 비트는 0이었음을 유추할 수 있다. 마찬가지로 state가 0이고 그때 AD수열의 상태가 'A'이면, 비트는 1이었음을 유추할 수 있다.

state가 1,11일 때도 비슷하게 구현할 수 있다.

이 코드를 이용해 AD수열 1개당 나올 수 있는 키수열 후보들을 찾는다. 여기서 가능한 키수열 후보들이 아주 많기 때문에 모두 찾는 것은 불가능했다. 그래서 적절하게 반복 횟수를 제한하여 실행시켰다. 그렇게 찾은 50묶음의 키수열 후보들을 $k_1, k_2, \cdots, k_{50}$ 이라고 하고, 2개의 AD수열끼리 키수열 후보들을 비교해 공통으로 가지고 있는 키수열 후보들을 모아 $k_{(1,2)}, k_{(3,4)}, \cdots, k_{(49,50)}$ ($k_{(i,i+1)}$ 는 $k_i, k_{k+1}$에 공통으로 들어있는 키수열 리스트)이라고 하자.

비교하는 코드를 실행해 본 결과 다음과 같이 14개의 리스트가 만들어졌다.

$$k_{(1,2)}, k_{(3,4)}, k_{(5,6)}, k_{(9,10)}, k_{(11,12)}, k_{(25,26)}, k_{(27,28)}, k_{(29,30)}, k_{(31,32)}, k_{(33,34)}, k_{(39,40)}, k_{(43,44)}, k_{(45,46)}, k_{(47,48)}$$

반복횟수를 제한했기 때문에 모든 키 후보를 찾을 수는 없었다.

## 3. 2번에서 찾은 50묶음의 키 후보에서 가장 많이 나온 후보들을 추린다.

2번에서 나온 14개의 리스트들을 비교해(2개의 리스트를 비교하는데 걸리는 시간 복잡도는 $O(n^2)$이다.) 가장 많이 나온 키수열을 찾는다. 결과적으로

$$k_1, k_2, k_3, k_4, k_9, k_{10}, k_{25}, k_{26}, k_{31}, k_{32}$$

에서 공통으로 나온 키수열과 1번에서 찾은 키의 일부분을 종합하여 아래의 키 후보를 찾을 수 있었다.

키 후보 :

$(10010000000011100100000101110011000000101101001101111001110110111111000101011010000000100010001101 01)_2$

# 2018 암호분석경진대회 : 4번 문제 답안

## 4. 3번에서 추린 키 후보들로 50개의 AD수열이 나올 수 있는지 확인하여 비밀 키를 찾는다.

Algorithm 1을 이용해 3번에서 찾은 키 후보가 비밀 키가 될 수 있는지 확인한다.

$(1001000000001110010000010111001100000010110100110111100111011011111100010101101000000010001000110101)_2$

```
/* Algorithm 1 */
    for key in data:
        if key=='0':
            if State==11:
                String+='AD'
                State=1
            else:
                String+='D'
                State=0
                  •
                  •
                  •
```

```
/* 만들어진 AD수열이 key_sequences에 있는지 */
 count = -1
 for a in arr:
     count+=1
     if a in lst:     #있으면 print
         print("[%d] ok" %(count))
```

위처럼 Algorithm 1을 적절히 변경하여 AD수열을 만드는 코드를 구현하였다. 위 코드는 만들어진 AD수열이 key_sequences에 있으면 '[만들어진 수열과 같은 수열의 위치] ok'를 출력하여 50개를 모두 만들 수 있는지 확인한다. 실행시켜 본 결과 34번째 AD수열을 제외한 49개의 AD수열은 만들어질 수 있음을 확인하였고, 34번째는 손으로 직접 가능함을 확인하였다. 따라서 key_sequences안에 모든 AD수열을 만들어내기 때문에 비밀 키가 된다.