

(a)

```

14 def Extract_RK(P,C):
15     out1=[0,0,0,0]
16     out2=[0,0,0,0]
17     key=[0,0,0,0]
18
19     out1=TC20.LM(C)
20     out2=TC20.ISB(out1)
21     key=TC20.AR(P,out2)
22
23     return key
    
```

$$\begin{aligned}
 &P \oplus \text{RK} = X \\
 &X = \text{ISB}(\text{LM}(C)) \\
 \Rightarrow &\text{RK} = P \oplus \text{ISB}(\text{LM}(C))
 \end{aligned}$$

위의 방법으로 라운드 키를 찾아냄

```

25 PT=[1,2,3,4]
26 key=[5,6,7,8]
27 CT=TC20.Enc_Round(PT,key)
28 find_RK=Extract_RK(PT,CT)
29 print("Round key=",find_RK)
    
```

확인을 위해 라운드 함수의 입력(PT)와 출력(C)를 key를 통해 만들어봄

```
Round key= [5, 6, 7, 8]
```

옳은 key값이 출력되었다는 것을 알 수 있음

(b)

```

33 def IsSlidPair(P1,C1,P2,C2):
34     key=Extract_RK(P1,P2)
35     if C2==TC20.Enc_Round(C1,key):
36         return True
37     else:
38         return False
    
```

$$\begin{aligned}
 &\text{Suppose } (P_1, C_1), (P_2, C_2) \text{ is Slid Pair} \\
 &P_1 \xrightarrow{\text{RK}} X_1 \xrightarrow{\text{RK}} X_2 \dots X_n = C_1 \\
 &P_2 \xrightarrow{\text{RK}} Y_1 \dots Y_q \xrightarrow{\text{RK}} Y_w = C_2 \\
 &\Rightarrow X_1 = F(P_1, \text{RK}) \Rightarrow \text{RK} = \text{Extract_RK}(P_1, P_2) \\
 &\Rightarrow C_2 = F(Y_q, \text{RK}) \Rightarrow C_2 = F(C_1, \text{RK})
 \end{aligned}$$

따라서, P1, P2로 만든 key로 C1을 암호화한 값이 C2라면, 두 쌍은 Slid Pair

```

40 key=[5,6,7,8]
41 PT1=[0,1,2,3]
42 CT1=TC20.TC20_Enc(PT1,key)
43 PT2=TC20.Enc_Round(PT1,key)
44 CT2=TC20.TC20_Enc(PT2,key)
45 print(IsSlidPair(PT1,CT1,PT2,CT2))

```

True

PT1과 PT1을 라운드 함수에 집어넣은 PT2를 만들어 Slid Pair를 만듦

실행 결과 True가 나옴

(c)

n 개의 (평문, 암호문) 쌍에서 k 개의 쌍을 선택

Slid Pair가 있을 확률: $P_{n,k}$

$P_{n,k} = P(E(P_i) = P_j \text{인 서로 다른 } i, j \text{가 있을 확률})$

$= 1 - P(\text{모든 서로 다른 } E(P_i) \neq P_j \text{인 확률})$

$= 1 - \left\{ \frac{n}{n} \times \frac{n-1}{n} \times \frac{n-2}{n} \times \dots \times \frac{n-(k-1)}{n} \right\}$

$= 1 - \left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \dots \left(1 - \frac{k-1}{n}\right)$

$e^x = 1 + x + \frac{x^2}{2!} + \dots = \sum_{j=0}^{\infty} \frac{x^j}{j!}$ (테일러 정리)

$e^x \approx 1 + x \quad (-1 < x < 1)$

For all $j \in [1, n), -\frac{j}{n} \in (-1, 1) (j \in \mathbb{Z}) \Rightarrow 1 - \frac{j}{n} \approx e^{-\frac{j}{n}}$

$\Rightarrow P_{n,k} \approx 1 - e^{\frac{1}{n}} \cdot e^{-\frac{k}{n}} = 1 - e^{-\frac{k-1}{n}} = 1 - e^{-\frac{(k-1)k}{2n}}$
 $\approx 1 - e^{-\frac{k^2}{2n}}$

$P_{n,k}$ 가 $\frac{1}{2}$ 이상이 되는 k 의 최소값:

$\frac{1}{2} \leq 1 - e^{-\frac{k^2}{2n}} \Rightarrow \frac{1}{2} \geq e^{-\frac{k^2}{2n}} \Rightarrow -\ln \frac{1}{2} \leq \frac{k^2}{2n}$

$\Rightarrow (2 \ln 2) n \leq k^2 \Rightarrow \lceil \sqrt{(2 \ln 2) n} \rceil \leq k$

$\therefore \lceil \sqrt{n} \rceil$ 개 ($\lceil \sqrt{n} \rceil \leq 4\sqrt{n}$) 만큼의 쌍을 선택하면, 그 중

Slid pair가 있을 확률이 $\frac{1}{2}$ 이상이 되므로

\sqrt{n} 의 입력 길이는 32비트 정도 된다

생일 문제에 의해, \sqrt{n} 개의 쌍을 선택하면, Slid Pair가 나오는 것을 기대할 수 있음

평문과 암호문의 조합: 2^{32} 개

$\rightarrow \sqrt{n} = \sqrt{2^{32}} = 2^{16}$ 이므로, 2^{16} 개의 쌍을 모으면, Slid Pair가 포함될 것으로 기대할 수 있음

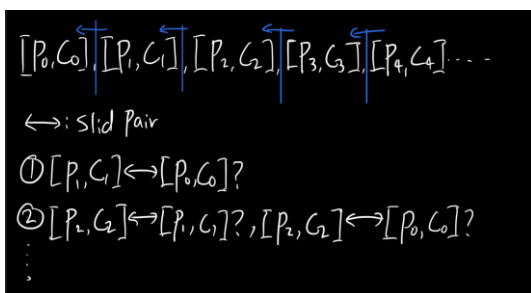
(d)

```

49 file_name = 'known_ptct.var'
50 item= load_var_from_file(file_name)
51
52 right_key=[]
53
54 for i in range(len(item)):
55     print("attack ",i+1)
56     print(item[i])
57     for j in range(0,i):
58         if IsSlidPair(item[i][0],item[i][1],item[j][0],item[j][1])==True:
59             #Slid 쌍인지 확인
60             print("find key!")
61             right_key=Extract_RK(item[i][0],item[j][0])
62             #Slid Pair라면, 라운드 키값을 획득
63             break
64     if right_key:
65         break
66     else:
67         print("didn't find slide pair")
68
69 print("key=",right_key)

```

'known_ptct.var'에서 pt와 ct쌍을 가져옴



첫번째 쌍부터 왼쪽으로 한칸씩 이동하여 Slid pair인지 확인

Slid pair이면, 라운드 키를 찾아 종료

```

attack 45126
[[0, 85, 255, 0], [185, 108, 109, 7]]
didn't find slide pair
attack 45127
[[0, 182, 197, 113], [225, 93, 200, 180]]
didn't find slide pair
attack 45128
[[0, 218, 8, 106], [18, 239, 139, 172]]
didn't find slide pair
attack 45129
[[0, 48, 50, 219], [18, 140, 68, 67]]
find key!
key= [0, 5, 0, 9]
PS C:\Users\Jin_Cheol\Desktop\HW2_20192233박진철\2번>

```

공격을 통해 라운드 키 [0.5.0.9]획득

```

72  ran_int=random.randint(0,len(item))
73  PT=item[ran_int][0]
74  print('PT=',PT)
75  CT=item[ran_int][1]
76  print('CT=',CT)
77  right_key=[0,5,0,9]
78  En_CT=TC20.TC20_Enc(PT,right_key)
79  print('En_CT=',En_CT)
80  if CT==En_CT:
81      print("success")
82  else:
83      print("Fail")

```

주어진 파일 속 아무 평문 암호문 쌍을 가져와 키가 맞는지 검증

```

PT= [0, 41, 167, 233]
CT= [53, 169, 141, 8]
En_CT= [53, 169, 141, 8]
success

```

얻어낸 키 값이 맞는 키라는 것을 알 수 있음

(e)

키 전수조사의 경우, 2^n 의 공격량이 필요하지만, Slide Attack은 $2^{(n/2)}$ 의 공격량으로 라운드키를 획득할 수 있으므로, 키 전수조사보다 적은 공격량으로 해독할 수 있음