# 암호분석 2023 - 과제 2

2022년 5월 20일까지

1. 수업시간에 사용한 32비트 암호 TC20에 다음과 같이 24비트 암호키 $[0, k_1, k_2, k_3]$를 사용할 때,

$$PT = [p_0, p_1, p_2, p_3] \rightarrow \boxed{E} \rightarrow CT = [c_0, c_1, c_2, c_3]$$

$CT = E(PT, key)$로 쓸 수 있다. 선택평문 $PT = [1, 1, 1, 1]$을 고정하고, 대응되는 암호문 $[c_0, c_1, c_2, c_3]$을 얻었을 때 사용된 암호키 $[0, k_1, k_2, k_3]$를 찾는 TMTO 공격을 생각하자.

(1) 랜덤한 $m$개 시작점(암호키)을 선택한다.

(2) 각 시작점에 대하여 길이 $t$인 체인을 만든다.

(3) 과정 (1)과 (2)를 반복하여 $\ell$개의 Hellman 테이블을 만든다.

$$X_{i,j+1} = f(X_{i,j}) = R(E(PT, X_{i,j})), \quad (i = 1, 2, \ldots, m,\ j = 0, 1, \ldots, t-1).$$

여기서 $R$은 32비트(암호문)를 24비트(암호키)로 변환하는 함수이다.

$$R[x_0, x_1, x_2, x_3] = [0, x_1, x_2, x_3].$$

$$\begin{array}{|l|l|l|}
\hline
\mathrm{SP}_1 & = X_{1,0} \rightarrow X_{1,1} \rightarrow X_{1,2} \rightarrow \cdots \rightarrow X_{1,t-1} \rightarrow X_{1,t} = & \mathrm{EP}_1 \\
\mathrm{SP}_2 & = X_{2,0} \rightarrow X_{2,1} \rightarrow X_{2,2} \rightarrow \cdots \rightarrow X_{2,t-1} \rightarrow X_{2,t} = & \mathrm{EP}_2 \\
\mathrm{SP}_i & = X_{i,0} \rightarrow X_{i,1} \rightarrow X_{i,2} \rightarrow \cdots \rightarrow X_{i,t-1} \rightarrow X_{i,t} = & \mathrm{EP}_i \\
\mathrm{SP}_m & = X_{m,0} \rightarrow X_{m,1} \rightarrow X_{m,2} \rightarrow \cdots \rightarrow X_{m,t-1} \rightarrow X_{m,t} = & \mathrm{EP}_m \\
\hline
\end{array}$$

(a) 공격에 필요한 파라미터로 $m = 2^8$, $t = 2^8$, $\ell = 2^8$을 사용하여, TMTO 테이블을 만들고 공격하는 프로그램을 작성하고, 단계별 설명과 함께 실행결과를 화면 캡처로 제출하라. 단, 사용한 프로그램 소스 `TMTO24.py`는 별도로 첨부한다.

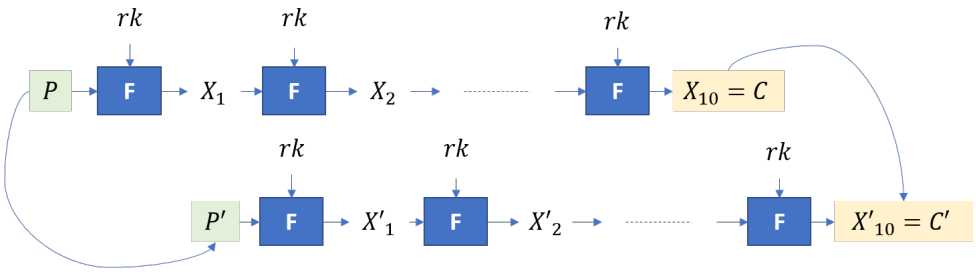
(b) TMTO 테이블은 고정하고 (즉, 고정된 선택평문에 대하여) 암호키를 다양하게 바꿔 가면서 만든 암호문으로 실험하여 (a) 공격의 성공율을 계산하라. (100번 중 몇 번 정도 성공하는가?)

(c) 하나의 TMTO 테이블에 포함된 $m \times t$개 암호키 중 서로 다른 암호키의 비율을 ECR(Expected Coverage Rate)라고 한다. (a)에서 사용한 파라미터로 TMTO 테이블을 만들고 ($\ell = 256$개 테이블이 생성됨) 각 테이블의 ECR을 출력하라. ECR 그래프를 그리고 평균값을 계산하라.

(d) 암호키로 32비트 $[k_0, k_1, k_2, k_3]$를 사용하는 TC20에 대한 TMTO 공격을 수행하라. 공격에 필요한 파라미터 $m$, $t$, $\ell$을 적절히 설정하고 공격 성공률을 추정하라.

(e) (d)에서 설정한 파라미터의 근거를 제시하고, 공격 성공률의 추정값과 실험결과를 비교, 분석하라.

2. TC20과 같이 모든 라운드에서 같은 라운드 키를 사용하는 블록암호는 안전하지 않음을 보여주는 공격으로 'Slide Attack'을 생각할 수 있다.




고정된 암호키를 사용한다고 가정하자. 위 그림과 같이 두 개의 (평문, 암호문) 쌍 $(P, C)$, $(P', C')$이 다음 조건을 만족하면 **'Slid Pair'**라고 한다.

$$P' = F(P, K), \quad C' = F(C, K),$$

여기서 $F$는 동일한 라운드 키를 사용하는 라운드 함수를 의미한다.

다음과 같은 단계로 TC20의 Slide Attack을 구현하라.

(a) 라운드 함수 $F$의 입력($P$)과 출력($C$)이 주어질 때, 사용된 라운드 키($K$)를 출력하는 함수 `Extract_RK(P,C)`를 작성하라. 즉, $C = F(P, K)$의 관계가 있을 때, 주어진 $P$, $C$로부터 $K$를 구하는 함수이다.

(b) 두 개의 (평문, 암호문) 쌍 $(P_1, C_1)$, $(P_2, C_2)$를 입력으로 하여 두 개의 쌍이 Slid Pair이면 True를 아니면 False를 출력하는 함수 `IsSlidPair(P1, C1, P2, C2)`를 작성하라.
$(P_1, P_2)$으로부터 라운드 키 $K$를 찾고, $C_2 = F(C_1, K)$를 만족하는지 확인하는 방법으로 구현하면 된다.

(c) 블록암호 TC20이 32비트 암호이므로 $2^{16}$개의 (평문, 암호문) 쌍을 모으면, 하나의 Slid Pair가 포함될 것으로 기대할 수 있음을 설명하라.

(d) 다음 코드로 (평문, 암호문)을 생성하고 결과를 파일 `known_ptct.var`에 저장했다고 가정하자.

```
def Generate_Known_pt_ct(num_pair, key):
    list_pool = []
    print('Generating PT-CT pairs', end='')
    for i in range(0,num_pair):
        PT = [0, random.randint(0,255), random.randint(0,255), random.randint(0,255)]
        CT = TC20.TC20_Enc(PT, key)
        item = copy.deepcopy([PT,CT])
        list_pool.append(item)
        if (i % (1<<13)) == 0:
            print('.', end='')
    print('  Done! \n')
    return list_pool

# 공격용 데이터를 만들때 사용하는 key는 공격자가 모른다고 가정한다.
pool = Generate_Known_pt_ct(1<<16, key)
save_var_to_file(pool, 'known_ptct.var')
```

이 파일을 이용하여 Slide Attack을 수행하는 프로그램을 작성하고 실행결과를 출력하라.

(e) Slide Attack은 모든 라운드에서 사용한 라운드 키가 동일하다면 암호 알고리즘의 구조와는 무관한 공격 (이를 generic attack 이라고 한다) 이다. 키 전수조사와 비교할 때, Slide Attack의 장점은 무엇인가?

* 각 문항에 대한 답을 하나의 보고서에 작성하고 필요한 파일(소스코드, 데이터 등)은 사용설명과 함께 별도로 첨부합니다. 구현에 성공하지 못한 경우에도 시도한 내용을 포함하여 보고서를 제출하기 바랍니다.