Side Channel Analysis Contest 2022			
7. Fault			
학교	국민대학교	이름	이현호

1. Fault model 분석

본 라이트업에서 사용할 기호와 그에 대한 정의는 다음 [표 1] 과 같다.

기호	정의
m	평문
c	암호문
d	비밀키
<N, e>	공개키 쌍

[표 1] 기호 및 정의

Square-and-multiply 지수승 알고리즘은 [표 2]와 같다. [표 2]의 $d=(d_{t-1},\dots,d_0)_2$에서 $d_i(0\le i<t-1)$값들은 0 또는 1의 값을 가진다. [표 2]에서 볼 수 있듯이 Square-and-multiply 지수승 알고리즘에서 3번 줄의 제곱 연산은 항상 수행하며 4번 줄의 조건문에 의해 $d_i=$ 1일 때만 곱셈 연산을 수행한다.

입력 : $x\in(\mathbb{Z}/N\mathbb{Z})^{\times}$, $d=(d_{t-1},\dots,d_0)_2$
출력 : $y=x^d$
1. $R_0\leftarrow 1$, $R_1\leftarrow x$
2. for $i=t-1$ down to 0 do
3. $\quad R_0\leftarrow R_0^2$
4. $\quad$ if $d_i=1$ then
5. $\quad\quad R_0\leftarrow R_0\cdot R_1$
6. return R_0

[표 2] Square-and-multiply 지수승 알고리즘

만약 [표 2]의 반복문에서 $i=j$일 때 단 한번 제곱 연산을 skip 하였다면 반복문의 연산 결과 $\hat{y_j}$는 [수식 1] 과 같다.

$$\hat{y}_j=\prod_{i=j+1}^{t-1}x^{d_i2^{i-1}}\times\prod_{i=0}^{j}x^{d_i2^i}$$

[수식 1]

[수식 1]의 결과로부터 [수식 2]의 결과를 도출해 낼 수 있다.

$$\hat{y_j}=\begin{cases}\hat{y}_{j-1} & \text{for } d_j=0\\ x^{2^{j-1}}\ \hat{y}_{j-1} & \text{for } d_j=1\end{cases}$$

[수식 2]

2. RSA 비밀키 복구

악의적인 공격자는 각 i에 대해 3번 줄 연산을 한 번만 skip하여 2046개의 새로운 메시지를 얻었다는 것으로 보아 비밀키의 크기가 2046 비트라는 사실을 알 수 있다. 그렇다면 [표 2]에서 $t=2046$ 일 것이다. 또한 $i=2045$일 때 $R_0=1$이므로 R_0을 제곱 연산을 한 값과 연산을 skip한 3번 줄 연산 결과는 $R_0^2=R_0=1^2=1$으로 같다. 따라서 $\hat{y}_{2045}=x^d$임을 쉽게 알 수 있다.

$x^d \bmod N=m$이기 때문에 $\hat{y}_{2045}\bmod N=x^d\bmod N=m$ 이다. 또한 [수식 2]를 [수식 3]으로 응용 가능하다.

$$\hat{y}_j \bmod N = \begin{cases} \hat{y}_{j-1} \bmod N & \text{for } d_j = 0 \\ x^{2^{j-1}} \hat{y}_{j-1} \bmod N & \text{for } d_j = 1 \end{cases}$$

[수식 3]

$\hat{y}_{2045} \bmod N = m$ 이라는 사실과 [수식 3]를 이용하여 복구한 비밀키 d는 다음과 같다.

$d=$
0b11101100110011001101000011001000100000100001001110010111110111100110001011111010011111001000
10010100000000100011111000000000111001101101100111000010100110000000100101110001111001011101111
0
01001010101101100110010011011001100001000110000101101011111111000110001110001111000000001000110
00011101111000100111010000001000011101011011010011111100011111110011000111101010000100111101110
00001011100110000010100110011111111001111000111000111011010001110110101001011111001011011111001
00001011010010001011110001111111000000111111010111011110011011111000100010100000000010101101111
00001010001010111000100011100101011011001001010011110001010011001101101100010001000000111001000
00001011111110010111110100111001101111011000111101000000111010001110110001111010110100110011001
10110111011010111101011011100011110000001000111000000000000111100010111101100010000110101001101
1
01000100000010010101100100001101101100001100011111010011011110011011110101011000011011010010100
00101001000010000101110011011011101111000110110000000010000000101110011000110011001010111101101
00111000011100001001010111100011100111010010010110100000101100111010000111101001110110101100000
11101100001101001110000111111001101001000000100110101001110011001111000110110101100111101000000
01000000011101111111001110011110101111111101010100010011001011101001111110010010000000110001101
01010100010000100011111100101100101110010101101001011100100110100000001000101001001000000001111
10101100101111011001111000111000000000101000001100000111111001011000101111101001110101010101111
00110001011100001010011100111110100111101011110001000110011010110010000111000000100011010100111
11001110101011110001110001001100011001011000010100110100111110010000010010010100000000100101010
11011001100011111011110001010001111011010001100010101010010100100000001100000111001000010000010
11110001010110100000010000000101000000001101011100100100001101010010111100011100111010011001011
11
01110101000101100101100101110010101110000111110001001011111010100110001000000000100110001011111
1010001010110001011111010010111000101110110001111011010000001110110010100
1

3. 위의 복구한 d를 이용해 $(d \times 123) \bmod 2^{128}$을 계산한 hex 값(Flag)은 다음과 같다.

b8 79 9d 16 92 c7 cc ad 1c 94 c1 e7 ce 0c 6c b3