HIGHT Quantum implementation

https://youtu.be/ODiy9G2mCvk

장경배

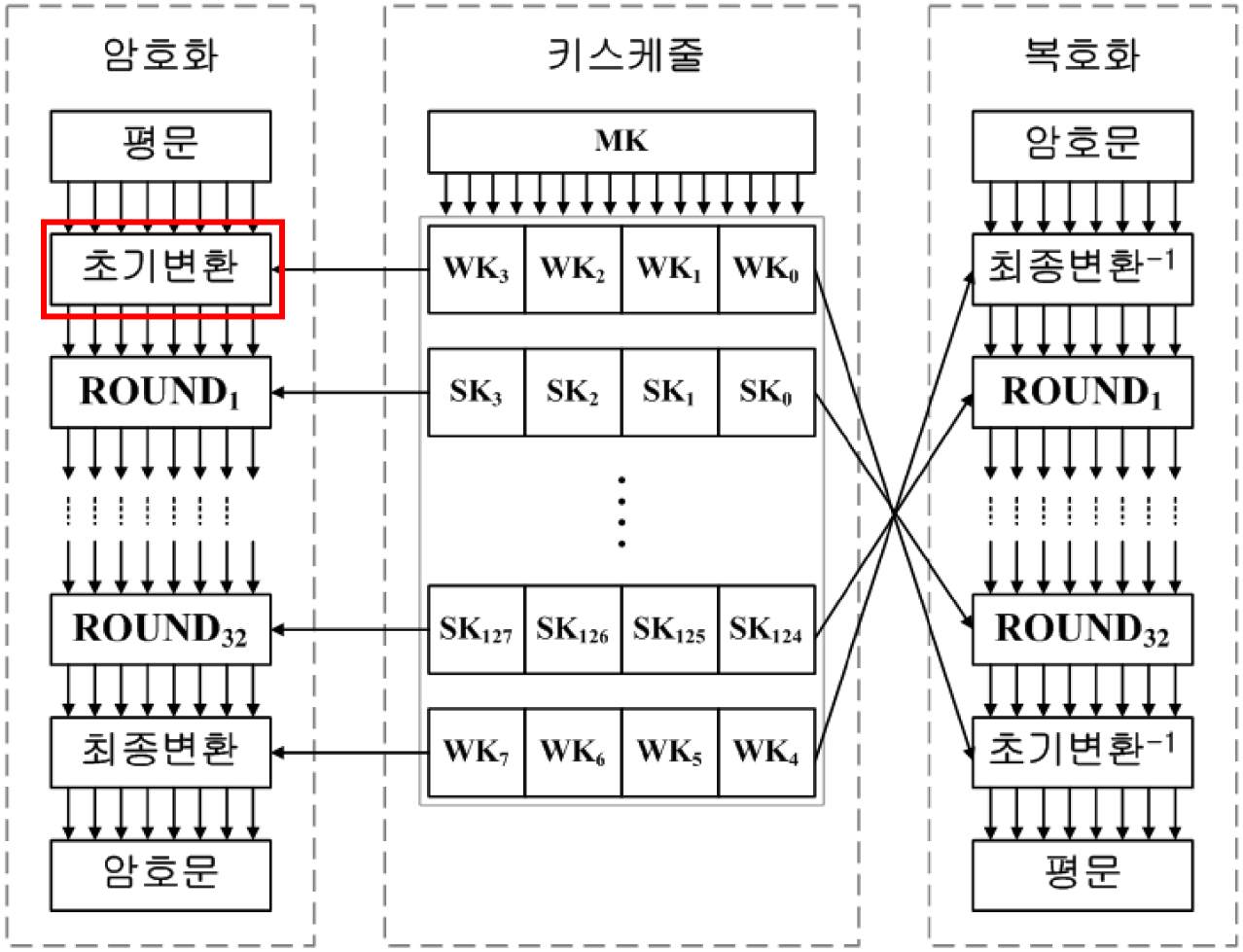
한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

HIGHT

- ARX 연산으로 구성 된, 128 - bit 마스터키, 64-bit의 평문, 암호문
- 제한적 자원을 갖는 환경에서 구현될 수 있도록 8-bit 단위의 ARX연산으로 설계
- **CHES** 06

HIGHT



HIGHT 전체구조

HIGHT 초기변환

- 마스터키를 활용해서 화이트닝 키 생성하고

$$\mathrm{WK_i} = \begin{cases} \mathrm{MK_{i+12}} & , \ 0 \le \mathrm{i} \le 3 \\ \mathrm{MK_{i-4}} & , \ 4 \le \mathrm{i} \le 7 \end{cases}$$

- 화이트닝키를 사용하여 입력 평문 ***P*** 를 ***X*** 로 변환

$$\begin{aligned} X_{0,i} &= P_i, \quad i = 1, 3, 5, 7 \\ X_{0,0} &= P_0 \boxplus \mathrm{WK}_0 \\ X_{0,2} &= P_2 \oplus \mathrm{WK}_1 \\ X_{0,4} &= P_4 \boxplus \mathrm{WK}_2 \\ X_{0,6} &= P_6 \oplus \mathrm{WK}_3 \end{aligned}$$

HIGHT 초기변환

- 마스터키를 활용해서 화이트닝 키 생성하고

$$\mathrm{WK}_i = \begin{cases} \mathrm{MK}_{i+12} & , \ 0 \le i \le 3 \\ \mathrm{MK}_{i-4} & , \ 4 \le i \le 7 \end{cases}$$

- 화이트닝키를 사용하여 입력 평문 ***P*** 를 ***X*** 로 변환

$$\begin{aligned} X_{0,i} &= P_i, \quad i = 1, 3, 5, 7 \\ X_{0,0} &= P_0 \boxplus \mathrm{WK}_0 \\ X_{0,2} &= P_2 \oplus \mathrm{WK}_1 \\ X_{0,4} &= P_4 \boxplus \mathrm{WK}_2 \\ X_{0,6} &= P_6 \oplus \mathrm{WK}_3 \end{aligned}$$

→

```
# key
mk0 = eng.allocate_qureg(8)
mk1 = eng.allocate_qureg(8)
mk2 = eng.allocate_qureg(8)
mk3 = eng.allocate_qureg(8)
mk4 = eng.allocate_qureg(8)
mk5 = eng.allocate_qureg(8)
mk6 = eng.allocate_qureg(8)
mk7 = eng.allocate_qureg(8)
mk8 = eng.allocate_qureg(8)
mk9 = eng.allocate_qureg(8)
mk10 = eng.allocate_qureg(8)
mk11 = eng.allocate_qureg(8)
mk12 = eng.allocate_qureg(8)
mk13 = eng.allocate_qureg(8)
mk14 = eng.allocate_qureg(8)
mk15 = eng.allocate_qureg(8)
```

128-bit 마스터키

```
x0 = eng.allocate_qureg(8)  # Plain text
x1 = eng.allocate_qureg(8)
x2 = eng.allocate_qureg(8)
x3 = eng.allocate_qureg(8)
x4 = eng.allocate_qureg(8)
x5 = eng.allocate_qureg(8)
x6 = eng.allocate_qureg(8)
x7 = eng.allocate_qureg(8)
```

64-bit 평문

HIGHT 초기변환

- 마스터키를 활용해서 화이트닝 키 생성하고

$$\mathrm{WK_i} = \begin{cases} \mathrm{MK_{i+12}} & , \ 0 \le \mathrm{i} \le 3 \\ \mathrm{MK_{i-4}} & , \ 4 \le \mathrm{i} \le 7 \end{cases}$$

- 화이트닝키를 사용하여 입력 평문 ***P*** 를 ***X*** 로 변환

$$\begin{aligned} X_{0,i} &= P_i, \quad i = 1, 3, 5, 7 \\ X_{0,0} &= P_0 \boxplus \mathrm{WK}_0 \\ X_{0,2} &= P_2 \oplus \mathrm{WK}_1 \\ X_{0,4} &= P_4 \boxplus \mathrm{WK}_2 \\ X_{0,6} &= P_6 \oplus \mathrm{WK}_3 \end{aligned}$$

→

```
#first whitening
#WK0 = mk12
#WK1 = mk13
#WK2 = mk14
#WK3 = mk15

#WK4 = mk0
#WK5 = mk1
#WK6 = mk2
#Wk7 = mk3

Add(eng, mk12, x0, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk13[i], x2[i])

Add(eng, mk14, x4, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk15[i], x6[i])
```

키 스케쥴

- 서브키 생성

For $i = 0$ to 7
    For $j = 0$ to 7
        $\mathrm{SK}_{16 \cdot i + j} \leftarrow \mathrm{MK}_{j - i \bmod 8} \boxplus \delta_{16 \cdot i + j}$;
    For $j = 0$ to 7
        $\mathrm{SK}_{16 \cdot i + j + 8} \leftarrow \mathrm{MK}_{(j - i \bmod 8) + 8} \boxplus \delta_{16 \cdot i + j + 8}$;

키 스케쥴

- LFSR ***h*** 의 연결 다항식은 $x^7 + x^3 + 1$ 이다. 이 다항식은 $F_2[x]$ 에서 원시다항식이기 때문에

  h 는 128 의 주기를 갖는다

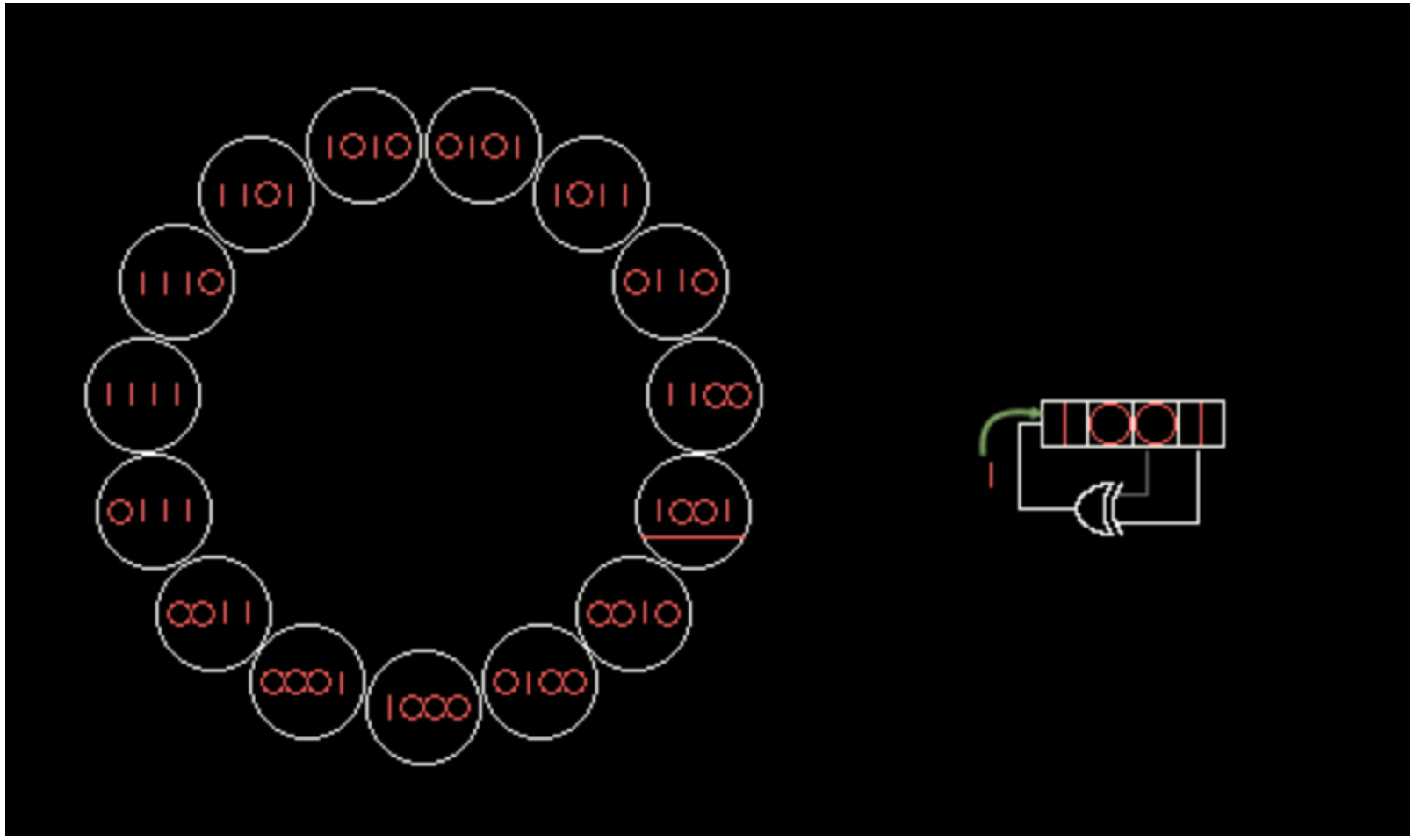
* LFSR : 선형 되먹임 시프트 레지스터

- h의 초기 내부 상태 값은 $\delta_0 = (s_6, s_5, s_4, s_3, s_2, s_1, s_0) = (1, 0, 1, 1, 0, 1, 0)_2$

- $i = 1, \cdots, 127$ 에 대하여 δ_i 는 다음과 같이 생성된다.

$$
\begin{aligned}
s_{i+6} &= s_{i+2} \oplus s_{i-1}, \\
\delta_i &= (s_{i+6},\ s_{i+5},\ s_{i+4},\ s_{i+3},\ s_{i+2},\ s_{i+1},\ s_i), \qquad 1 \le i \le 127
\end{aligned}
$$

LFSR


1010
0101
1011
0110
1100
1001
0010
0100
1000
0001
0011
0111
1111
1110
1101

키 스케쥴

$$\delta_0 = (s_6, s_5, s_4, s_3, s_2, s_1, s_0) = (1, 0, 1, 1, 0, 1, 0)_2$$

$$s_{i+6} = s_{i+2} \oplus s_{i-1},$$
$$\delta_i = (s_{i+6}, s_{i+5}, s_{i+4}, s_{i+3}, s_{i+2}, s_{i+1}, s_i), \qquad 1 \le i \le 127$$

↓

```
def generate_s(eng, s, i):
    i = i + 1

    if (i == 0):
        return i

    # new(0) , 6, 5, 4, 3, 2 , 1 ( i % 7 , (i-1) % 7 )
    CNOT | (s[(i+2)%7], s[(i-1)%7])

    return i
```

라운드 함수

HIGHT는 라운드 함수에서는 다음과 같은 보조 함수를 사용

$$F_0(X) = X^{\lll 1} \oplus X^{\lll 2} \oplus X^{\lll 7}$$
$$F_1(X) = X^{\lll 3} \oplus X^{\lll 4} \oplus X^{\lll 6}$$

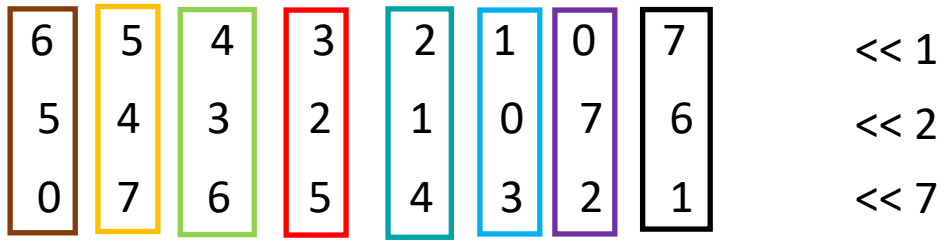
라운드 함수는 Round 1 부터 Round 32 까지 32회 반복

$$X_{i,j} = X_{i-1,j-1}, \quad j = 1, 3, 5, 7$$
$$X_{i,0} = X_{i-1,7} \oplus (F_0(X_{i-1,6}) \boxplus \mathrm{SK}_{4i-1})$$
$$X_{i,2} = X_{i-1,1} \boxplus (F_0(X_{i-1,0}) \oplus \mathrm{SK}_{4i-4})$$
$$X_{i,4} = X_{i-1,3} \oplus (F_0(X_{i-1,2}) \boxplus \mathrm{SK}_{4i-3})$$
$$X_{i,6} = X_{i-1,5} \boxplus (F_0(X_{i-1,4}) \oplus \mathrm{SK}_{4i-2})$$

라운드 함수

$$F_0(X) = X^{\lll 1} \oplus X^{\lll 2} \oplus X^{\lll 7}$$
$$F_1(X) = X^{\lll 3} \oplus X^{\lll 4} \oplus X^{\lll 6}$$

7 6 5 4 3 2 1 0



6 5 4 2 7 3 0 1

```
def F0 (eng, x): #21cnot
    CNOT | (x[4], x[5])    # 5 = 5+4
    CNOT | (x[3], x[4])    # 4 = 4+3

    CNOT | (x[2], x[3])    # 3 = 3+2
    CNOT | (x[2], x[0])    # 0 = 0+2

    CNOT | (x[4], x[2])    # 2 = 2+4+3
    CNOT | (x[5], x[2])    # 2 = 2+5+3

    CNOT | (x[7], x[5])    # 5 = 5+4+7
    CNOT | (x[6], x[4])    # 4 = 4+3+6

    CNOT | (x[0], x[3])    # 3 = 3+0
    CNOT | (x[1], x[3])    # 3 = 3+0+1

    CNOT | (x[6], x[1])    # 1 = 1+6
    CNOT | (x[7], x[1])    # 1 = 1+6+7

    CNOT | (x[7], x[0])    # 0 = 0+2+7

    CNOT | (x[5], x[6])    # 6 = 6+5+4+7
    CNOT | (x[4], x[6])    # 6 = 3+5+7

    CNOT | (x[3], x[6])    # 6 = 0+1+5+7
    CNOT | (x[1], x[6])    # 6 = 0+5+6

    CNOT | (x[6], x[7])    # 7 = 7+0+5+6
    CNOT | (x[5], x[7])    # 7 = 4+0+6

    CNOT | (x[1], x[7])    # 7 = 1+4+0+7
    CNOT | (x[0], x[7])    # 7 = 1+4+2
```

라운드 함수

라운드 함수는 Round 1 부터 Round 32 까지 32회 반복

* i = -1

$$
\begin{aligned}
X_{i,j} &= X_{i-1,j-1}\ ,\ \ j = 1,\ 3,\ 5,\ 7 \\
X_{i,0} &= X_{i-1,7} \oplus (F_0(X_{i-1,6}) \boxplus \mathrm{SK}_{4i-1}) \\
X_{i,2} &= X_{i-1,1} \boxplus (F_0(X_{i-1,0}) \oplus \mathrm{SK}_{4i-4}) \\
X_{i,4} &= X_{i-1,3} \oplus (F_0(X_{i-1,2}) \boxplus \mathrm{SK}_{4i-3}) \\
X_{i,6} &= X_{i-1,5} \boxplus (F_0(X_{i-1,4}) \oplus \mathrm{SK}_{4i-2})
\end{aligned}
$$

```
def Round(eng, s, mk_first, mk_second, mk_third, mk_fourth, x0, x1, x2, x3, x4, x5, x6, x7, i, c0):
    ################
    i = generate_s(eng, s, i)
    Add_s(eng, s, mk_first, i % 7, (i-1) % 7, 0, 7, c0)
    F0(eng, x0)
    F0_CNOT(eng, mk_first, x0)
    X_Add(eng, x0, x1, c0)

    #Reverse
    F0_CNOT(eng, mk_first, x0)
    F0_reverse(eng, x0)
    Add_s_reverse(eng, s, mk_first, i % 7, (i-1) % 7, 0, 7, c0)
    # END : X2 = X1
```

*

$$
\begin{aligned}
&\text{For } \ i = 0 \ \text{ to } \ 7 \\
&\quad \text{For } \ j = 0 \ \text{ to } \ 7 \\
&\qquad \mathrm{SK}_{16 \cdot i + j} \leftarrow \mathrm{MK}_{j - i \bmod 8} \boxplus \delta_{16 \cdot i + j}\ ; \\
&\quad \text{For } \ j = 0 \ \text{ to } \ 7 \\
&\qquad \mathrm{SK}_{16 \cdot i + j + 8} \leftarrow \mathrm{MK}_{(j - i \bmod 8) + 8} \boxplus \delta_{16 \cdot i + j + 8};
\end{aligned}
$$

*

```
def F0_CNOT(eng, a, b): # CNOT normal to F0

    CNOT | (a[0], b[1])
    CNOT | (a[1], b[0])
    CNOT | (a[2], b[3])
    CNOT | (a[3], b[7])
    CNOT | (a[4], b[2])
    CNOT | (a[5], b[4])
    CNOT | (a[6], b[5])
    CNOT | (a[7], b[6])
```

라운드 함수

```
def Round(eng, s, mk_first, mk_second, mk_third, mk_fourth, x0, x1, x2, x3, x4, x5, x6, x7, i, c0):
    ##################
    i = generate_s(eng, s, i)
    Add_s(eng, s, mk_first, i % 7, (i-1) % 7, 0, 7, c0)
    F0(eng, x0)
    F0_CNOT(eng, mk_first, x0)
    X_Add(eng, x0, x1, c0)

    #Reverse
    F0_CNOT(eng, mk_first, x0)
    F0_reverse(eng, x0)
    Add_s_reverse(eng, s, mk_first, i % 7, (i-1) % 7, 0, 7, c0)
    # END : X2 = X1

    ###################

    i = generate_s(eng, s, i)
    Add_s(eng, s, mk_second, i % 7, (i-1) % 7, 0, 7, c0)
    F0(eng, x2)
    F0_Add(eng, mk_second, x2, c0)
    X_CNOT(eng, x2, x3)

    #Reverse
    F0_Add_Reverse(eng, mk_second, x2, c0)
    F0_reverse(eng, x2)
    Add_s_reverse(eng, s, mk_second, i % 7, (i-1) % 7, 0, 7, c0)
    # END : X4 = X3

    ###################
```

```
    ###################

    i = generate_s(eng, s, i)
    Add_s(eng, s, mk_third, i % 7, (i-1) % 7, 0, 7, c0)
    F0(eng, x4)
    F0_CNOT(eng, mk_third, x4)
    X_Add(eng, x4, x5, c0)

    #Reverse
    F0_CNOT(eng, mk_third, x4)
    F0_reverse(eng, x4)
    Add_s_reverse(eng, s, mk_third, i % 7, (i - 1) % 7, 0, 7, c0)
    #END : X6 = X5

    ####################

    i = generate_s(eng, s, i)
    Add_s(eng, s, mk_fourth, i % 7, (i - 1) % 7, 0, 7, c0)
    F0(eng, x6)
    F0_Add(eng, mk_fourth, x6, c0)
    X_CNOT(eng, x6, x7)

    #Reverse
    F0_Add_Reverse(eng, mk_fourth, x6, c0)
    F0_reverse(eng, x6)
    Add_s_reverse(eng, s, mk_fourth, i % 7, (i - 1) % 7, 0, 7, c0)
    #End : x0 = x7

    return i
```

라운드 함수

```
Add(eng, mk12, x0, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk13[i], x2[i])

Add(eng, mk14, x4, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk15[i], x6[i])

i=-1

# Round 1
i = Round(eng, s, mk0, mk1, mk2, mk3, x0, x1, x2, x3, x4, x5, x6, x7, i, c0)
# Round2
i = Round(eng, s, mk4, mk5, mk6, mk7, x7, x0, x1, x2, x3, x4, x5, x6, i, c0)
#Round 3
i = Round(eng, s, mk8, mk9, mk10, mk11, x6, x7, x0, x1, x2, x3, x4, x5, i, c0)
# Round 4
i = Round(eng, s, mk12, mk13, mk14, mk15, x5, x6, x7, x0, x1, x2, x3, x4, i, c0)
```

```
# Round 5
i = Round(eng, s, mk7, mk0, mk1, mk2, x4, x5, x6, x7, x0, x1, x2, x3, i, c0)
# Round 6
i = Round(eng, s, mk3, mk4, mk5, mk6, x3, x4, x5, x6, x7, x0, x1, x2, i, c0)
# Round 7
i = Round(eng, s, mk15, mk8, mk9, mk10, x2, x3, x4, x5, x6, x7, x0, x1, i, c0)
# Round 8
i = Round(eng, s, mk11, mk12, mk13, mk14, x1, x2, x3, x4, x5, x6, x7, x0, i, c0)


# Round 9
i = Round(eng, s, mk6, mk7, mk0, mk1, x0, x1, x2, x3, x4, x5, x6, x7, i, c0)
# Round 10
i = Round(eng, s, mk2, mk3, mk4, mk5, x7, x0, x1, x2, x3, x4, x5, x6, i, c0)
# Round 11
i = Round(eng, s, mk14, mk15, mk8, mk9, x6, x7, x0, x1, x2, x3, x4, x5, i, c0)
# Round 12
i = Round(eng, s, mk10, mk11, mk12, mk13, x5, x6, x7, x0, x1, x2, x3, x4, i, c0)
```

.
.
.

```
# Round 32
i = Last_Round(eng, s, mk13, mk14, mk15, mk8, x1, x2, x3, x4, x5, x6, x7, x0, i, c0)
```

라운드 함수

HIGHT 라스트 라운드, 최종변환

$$
\begin{aligned}
&X_{32,i} = X_{31,i}\,, \quad i = 0,\ 2,\ 4,\ 6 \\
&X_{32,1} = X_{31,1} \boxplus (F_1(X_{31,0}) \oplus \mathrm{SK}_{124}) \\
&X_{32,3} = X_{31,3} \oplus (F_0(X_{31,2}) \boxplus \mathrm{SK}_{125}) \\
&X_{32,5} = X_{31,5} \boxplus (F_1(X_{31,4}) \oplus \mathrm{SK}_{126}) \\
&X_{32,7} = X_{31,7} \oplus (F_0(X_{31,6}) \boxplus \mathrm{SK}_{127})
\end{aligned}
$$

$$
\begin{aligned}
&C_i = X_{32,i}\,, \quad i = 1,\ 3,\ 5,\ 7 \\
&C_0 = X_{32,0} \boxplus \mathrm{WK}_4 \\
&C_2 = X_{32,2} \oplus \mathrm{WK}_5 \\
&C_4 = X_{32,4} \boxplus \mathrm{WK}_6 \\
&C_6 = X_{32,6} \oplus \mathrm{WK}_7
\end{aligned}
$$

```
# Round 32
i = Last_Round(eng, s, mk13, mk14, mk15, mk8, x1, x2, x3, x4, x5, x6, x7, x0, i, c0)

#Last Whitening
Add(eng, mk0, x1, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk1[i], x3[i])

Add(eng, mk2, x5, 0, 7, 0, 7, c0)

for i in range(8):
    CNOT | (mk3[i], x7[i])

#END cipher : 0, 7, 6, 5, 4, 3, 2, 1
```

감사합니다