SEED 암호

https://www.youtube.com/watch?v=c84tCXPwgRI

SEED 개요

- SEED 암호

  1999년 KISA(한국인터넷진흥원)에서 개발한 블록암호 알고리즘

  전자상거래 등에서 주로 사용됨 (ex. Active X를 통해 SEED 사용)

  128비트의 평문을 입력으로 받아 128비트의 키로 암호화(현재 256비트 키로 암호화 하는 방식도 개발됨)

  Feistel 구조로 이루어짐

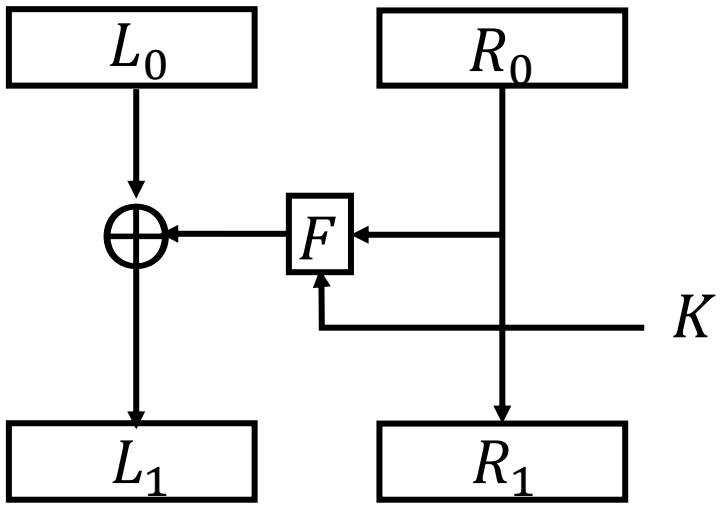
➢ Feistel 구조?

  입력을 좌우 블록으로 분할하여

  한블록에 라운드 함수를 적용시킨 출력 값을

  다른 블록에 적용하는 과정을 반복적으로 시행하는 구조

  Ex)DES

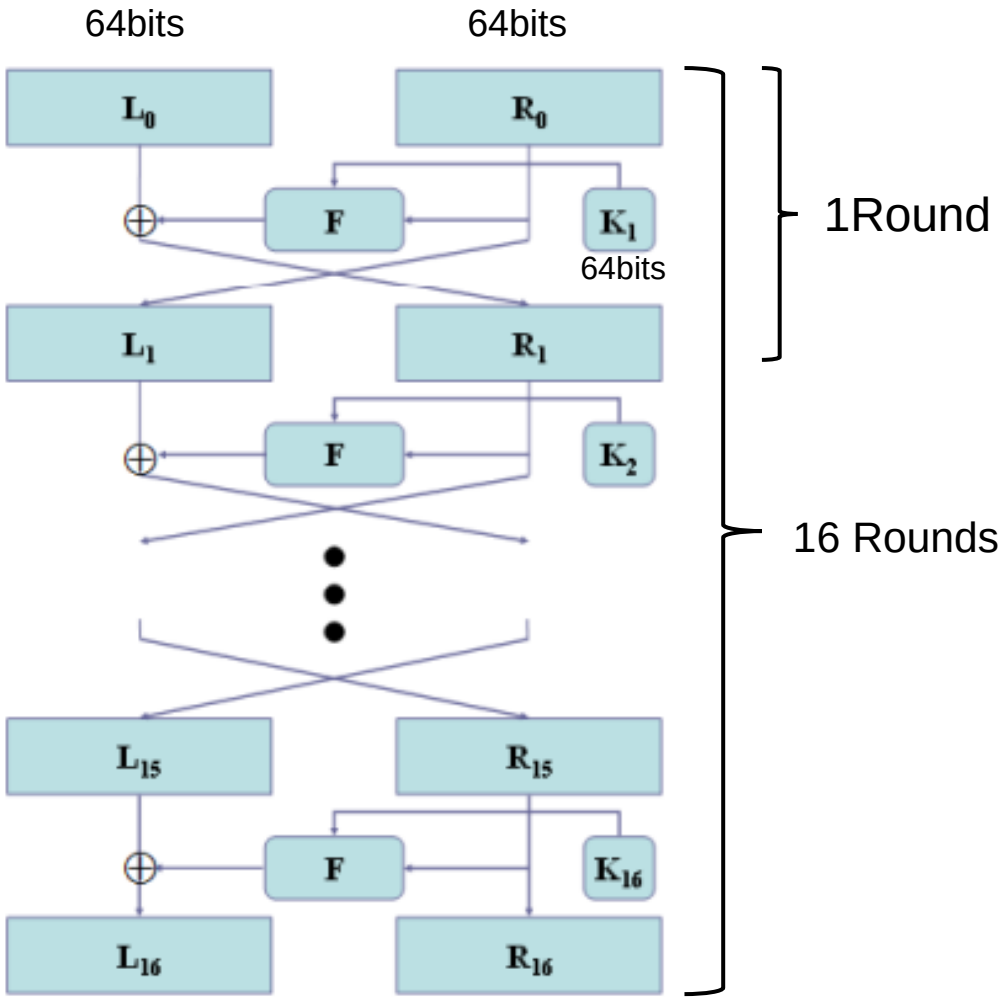


SEED 구조

- SEED 구조

전체 구조는 Feistel 구조로 이루어짐.
128 비트 평문 블록과 128비트 키를 사용
128비트 평문을 각각 64비트씩 좌우로 나누어 연산
총 16라운드 진행

* 128비트 키를 사용한다는 것은 K_i가 128비트가 아닌 128비트키를 이용하여 K_i를 생성.
K_i 는 64비트



SEED 구조 – F Function

- F Function

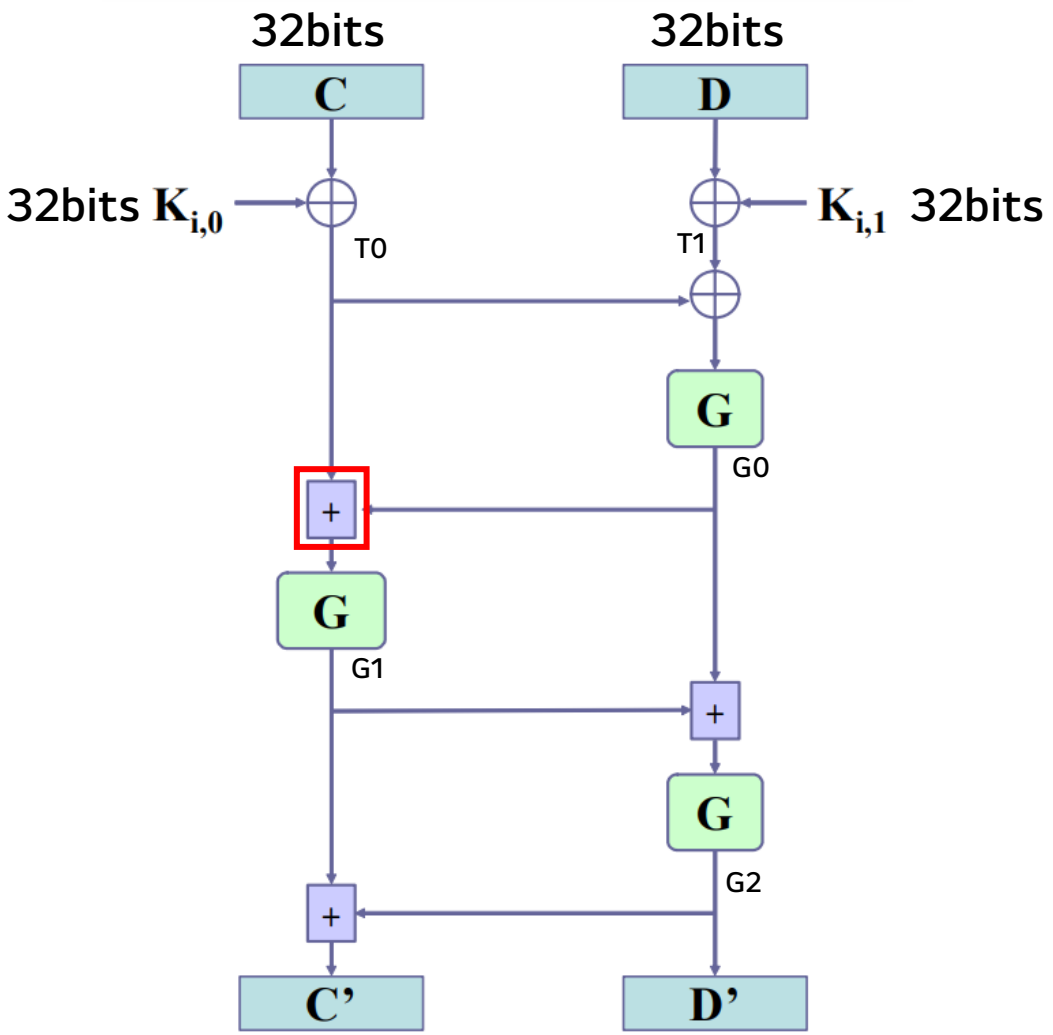
  R_i 과 K(= $K_{i,0}$, $K_{i,1}$)를 입력으로 받음

  R_i 는 다시 32비트 C,D로 나뉨.

  XOR, Addition, G Function으로 이루어짐.

  C,D는 각각 $K_{i,0}$ 과 $K_{i,1}$ 과 XOR 연산

양자컴퓨터에서 XOR이나 다른 연산은 쉽게 구현되지만
Addition은 여러 양자 게이트들의 조합으로 구현 해야함.

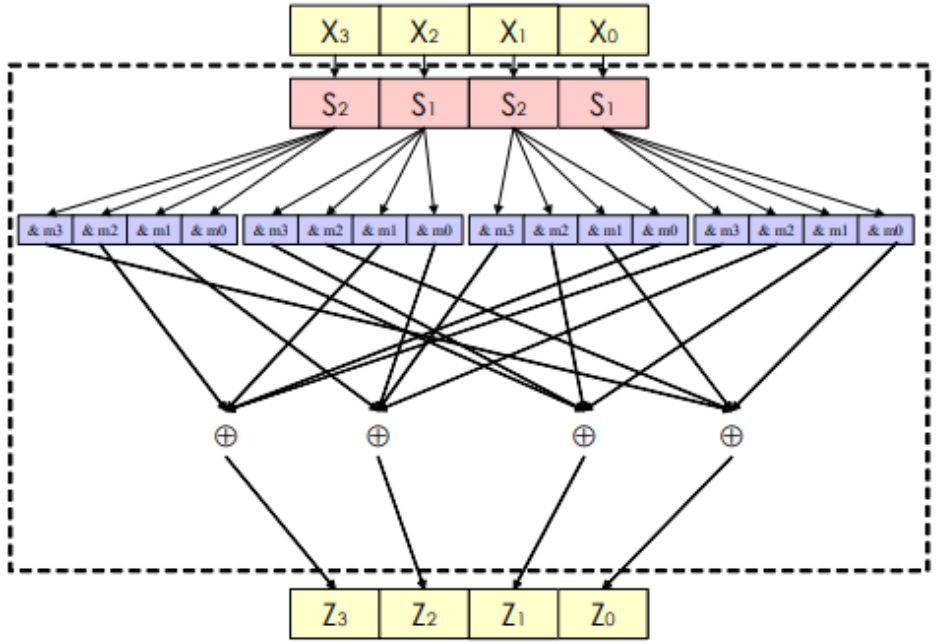


SEED 구조 – G Function

- G Function

  2개의 Sbox 사용

  - Sbox는 입력으로 8비트값을 받음
  - 상수 m0,m1,m2,m3와 &연산



$$Z = SS_3(X_3) \oplus SS_2(X_2) \oplus SS_1(X_1) \oplus SS_0(X_0)$$

$$Y_3 = S_2(X_3),\quad Y_2 = S_1(X_2),\quad Y_1 = S_2(X_1),\quad Y_0 = S_1(X_0),$$

$$\begin{aligned}
Z_3 &= (Y_0 \& m_3) \oplus (Y_1 \& m_0) \oplus (Y_2 \& m_1) \oplus (Y_3 \& m_2)\\
Z_2 &= (Y_0 \& m_2) \oplus (Y_1 \& m_3) \oplus (Y_2 \& m_0) \oplus (Y_3 \& m_1)\\
Z_1 &= (Y_0 \& m_1) \oplus (Y_1 \& m_2) \oplus (Y_2 \& m_3) \oplus (Y_3 \& m_0)\\
Z_0 &= (Y_0 \& m_0) \oplus (Y_1 \& m_1) \oplus (Y_2 \& m_2) \oplus (Y_3 \& m_3)
\end{aligned}$$

$$(m_0 = 0\text{xfc},\quad m_1 = 0\text{xf3},\quad m_2 = 0\text{xcf},\quad m_3 = 0\text{x3f})$$

$$\begin{aligned}
SS_3 &= S_2(X_3)\&m_2 \;\|\; S_2(X_3)\&m_1 \;\|\; S_2(X_3)\&m_0 \;\|\; S_2(X_3)\&m_3,\\
SS_2 &= S_1(X_2)\&m_1 \;\|\; S_1(X_2)\&m_0 \;\|\; S_1(X_2)\&m_3 \;\|\; S_1(X_2)\&m_2,\\
SS_1 &= S_2(X_1)\&m_0 \;\|\; S_2(X_1)\&m_3 \;\|\; S_2(X_1)\&m_2 \;\|\; S_2(X_1)\&m_1,\\
SS_0 &= S_1(X_0)\&m_3 \;\|\; S_1(X_0)\&m_2 \;\|\; S_1(X_0)\&m_1 \;\|\; S_1(X_0)\&m_0,
\end{aligned}$$

SEED 구조 - Sbox

- **Sbox 구현**

  양자 컴퓨터에서는 Look-up table사용 불가 -> 특정 상태로 결정 지을 수 없기 때문

$$S_i : Z_{2^8} \rightarrow Z_{2^8},$$
$$S_1(x) = A^{(1)} \cdot x^{247} \oplus 169 \quad S_2(x) = A^{(2)} \cdot x^{251} \oplus 56$$

GF(2^8)에서 계산

SEED GF(2^8)의 기약다항식 $p(x) = x^8 + x^6 + x^5 + x + 1$

$x^{-1} \equiv x^{254} mod\ p(x)$

$(x^{-1})^8 \equiv x^{247} mod\ p(x)$

$(x^{-1})^4 \equiv x^{251} mod\ p(x)$

$x^{-1} = x^{254} = \left((a \cdot a^2) \cdot (a \cdot a^2)^4 \cdot (a \cdot a^2)^{16} \cdot a^{64}\right)^2$

$$A^{(1)} = \begin{array}{c} \\ x_7 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_0 \end{array} \begin{array}{c} \begin{array}{cccccccc} x_7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x_0 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}, \quad A^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

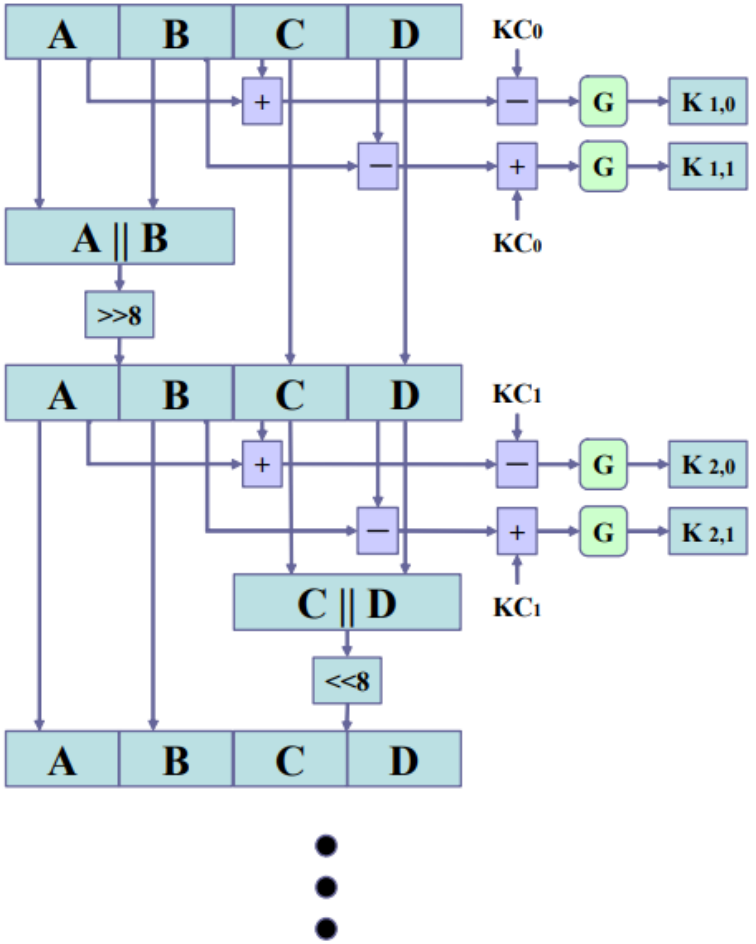
SEED 구조 - 키스케줄

- 라운드 키 생성

128비트 키 $K = A||B||C||D$ $(A, B, C, D$ 32비트)
$K_{i,0}, K_{i,1}$ = 32비트

홀수 라운드일 경우 : $A||B$ Right shift 8
짝수 라운드일 경우 : $C||D$ Left shift 8

Sub 연산
-> 1의 보수와 +1을 사용하여 계산
Ex) KC_0(빼지는 수)을 1의 보수를 취한 후 $A + C$와 add
그 후 +1



Q & A