# SEED

Lee, Hoon-Jae

## History

- SEED
  - Need of Korean standard encryption algorithm
  - Need of more-bits encryption algorithm
    - DES: 56 bits, RC4: 40 bits
  - SEED is not an abbreviation, just a name
- HAS-160
  - One of the Korean Hash cryptographic algorithms
  - Longer bits than MD5

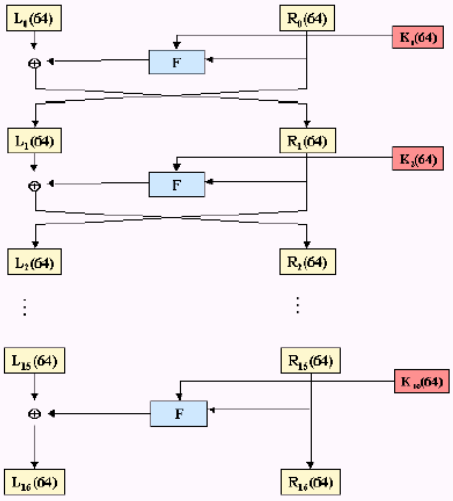
## Motivation

- Korean government enforces the use of SEED
  - Mandatory cipher on internet banking applications
- Proprietary usage of SEED
  - Possible security holes
  - Lack of compatibility
- Private TLS ID

## Features of SEED and HAS-160

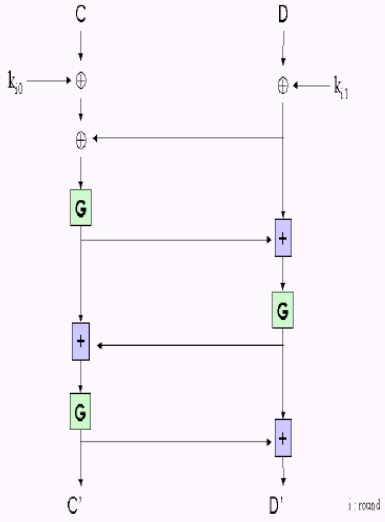
- SEED
  - 128-bit block cipher
  - Feistel structure with 2 s-boxes
- HAS-160
  - 160-bit output secure hash function
  - Looks like SHA1
- Specifications are available at TTA
  - http://www.tta.or.kr
  - Written in Korean

# SEED - Structure




- DES-like structure
- The size of input/output bit is fixed 128-bit
- The size of key bit is fixed 128-bit
- Four 8*8 S-boxes
- Fixed xor & modular addition operations
- The number of rounds is fixed 16

# SEED -round function F

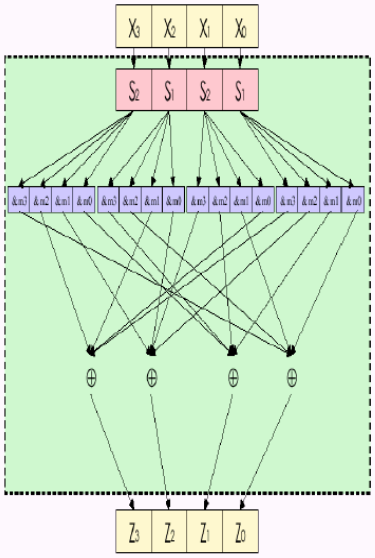



- 64-bit input/output block
- Divided into 32-bit block
- Wrapped with 4 phases
  - mixing phase of two 32-bit subkey block
  - 3 layers of G functions with addition operation

$C' = G[G[G[(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})] + (D \oplus k_{i,1})] + G[(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})]]$

$D' = C' + G[G[(C \oplus k_{i,0}) \oplus (D \oplus k_{i,1})] + (D \oplus k_{i,1})]$

# SEED - G_function (S -Box)



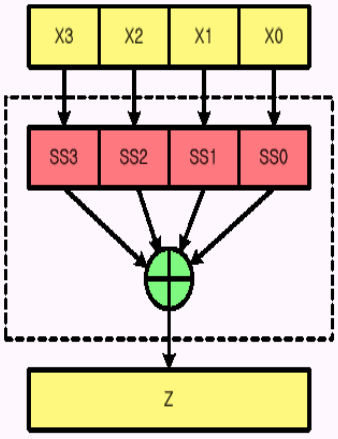
- Four 8*8 S-boxes
- Block permutation with sixteen 4-bit sub-blocks

Y3=S2(X3), Y2=S1(X2), Y1=S2(X1), Y0=S1(X0)

Z0 = (Y0&m0) ⊕ (Y1&m1) ⊕ (Y2&m2) ⊕ (Y3&m3)
Z1 = (Y0&m1) ⊕ (Y1&m2) ⊕ (Y2&m3) ⊕ (Y3&m0)
Z2 = (Y0&m2) ⊕ (Y1&m3) ⊕ (Y2&m0) ⊕ (Y3&m1)
Z3 = (Y0&m3) ⊕ (Y1&m0) ⊕ (Y2&m1) ⊕ (Y3&m2)

( m0=0xfc, m1=0xf3, m2=0xcf, m3=0x3f )

# SEED - G_function (SS -Box)




- Four 8*32 SS-boxes & xor operation

SS3 = S2(X3)&m2||S2(X3)&m1||S2(X3)&m0 ||S2(X3)&m3
SS2 = S1(X2)&m1||S1(X2)&m0||S1(X2)&m3 ||S1(X2)&m2
SS1 = S2(X1)&m0||S2(X1)&m3||S2(X1)&m2 ||S2(X1)&m1
SS0 = S1(X0)&m3||S1(X0)&m2||S1(X0)&m1 ||S1(X0)&m0

Z = SS3(X3) ⊕ SS2(X2) ⊕ SS1(X1) ⊕ SS0(X0)

# SEED - Key Schedule