DES (Data Encryption Standard)

10/2/19 ①

late 1960's Feistel working for IBM, created a symmetric cipher system (shared secret key) to be used internally.

In early 70s, NSA (National Security Agency) approached Feistel about modifying his cipher to become the gov't. (Open contest to choose a cipher sys.)

NSA changed some portions of Feistel's original design (namely the S-boxes).

WAS THE U.S. Gov't Standard from 1974 – 1998ish (late 1990s early 2000s)

In late 1990s, DES challenge - brute forced a DES key in 3 months (distributed)

Best cryptanalysis progress was using 2^{48} chosen plain/cipher text pairs.

DES is a block cipher.

Plaintext/Cipher - 64 bit blocks

Key - 56 bits (add 8 check bits)

(1) IP - Initial Permutation

(2) 16 rounds of a Feistel Structure (Key used here)

(3) IP^{-1} - Init Perm inverse

Step 1: IP(x)

58 50 42 34 ...

60 52 44 ...

63 55 47 ...

bit 58 = 0
bit 50 = 0
bit 42 = 0
bit 34 = 1

Orig Input: A347 C0E9 DD26 8538

1101 0010 1000 0011

$X_0 = IP(x)$

How to calculate IP^{-1} from IP

In IP, we find $1 \rightarrow 58$, $2 \rightarrow 50$, $3 \rightarrow 42$, ...

$IP^{-1}(58) = 1$
$IP^{-1}(50) = 2$
$IP^{-1}(42) = 3$

Step 2: Rounds

$X_0 = IP(x) = L_0 | R_0$ (32 bits | 32 bits)

```
for (int i=1; i<=16; i++) {
    L[i] = R[i-1];          // 32 bits
    R[i] = L[i-1] ⊕ f(R[i-1], K[i])
}
```

Round i key: ↓ 48 bits

$L[i-1]$: 32 bits

$f(R[i-1], K[i])$ → output = 32 bits

Step 2 Cont: function f

```
f(32 bits A, 48 bits J) {
    A' = E(A) // expands A to be 48 bits
    B = A' ⊕ J = B1 B2 ... B8 // 48 bits subdivided into 8 subblocks of 6 bits each.
    for (int i=1; i<=8; i++)
        Ci = Si(Bi) // for all Si input = 6 bits, output = 4 bits
    C = C1 C2 C3 ... C8 // C is 32 bits
    return P(C)
}
S-boxes (S1, S2, S3 ... , S8)
```

$S_1(b_1 b_2 b_3 b_4 b_5 b_6)$ - row = $b_1 b_6$
col = $b_2 b_3 b_4 b_5$

$S_1(101110)$ - row = 10 = row 2
col = 0111 = col 7

$S_1[2][7] = 11$

$S_1(011011) = 5$ row = 01 = row 1
col = 1101 = col 13

$S_1[1][13] = 5$