CIS 3362 10/5/22 DES

Data Encryption Standard

Standardized symmetric key encryption system

Created by Horst Feistel worked @ IBM.

* NSA made some changes (minor) to Feistel's original submission

Input: 64 bit blocks

Key: 56 bits transmitted in 64 bits with 8 checksum bits

Keyspace is 2^{56}.

↙ Checksum bits each row must have ODD parity

$k_1\ k_2\ k_3\ \ldots$

$k_9\ k_{10}\ k_{11}\ \ldots$

$\vdots$

$k_{57}\ k_{58}\ \ldots$

1011 011 **0**

0000 101 **1**

etc.

Key in 16 hex chars with checksum bits included

Call input P

1. $P_0 = L_0 R_0 = IP(P)$ → init perm

Example of using IP

Input HEX

C F	→	1 1 0 0
A Ø		1 0 1 0
3 7		0 0 1 1
4 9		0 1 0 0
8 2		1 0 0 0
5 5		0 1 0 1
D 1		1 1 0 1
6 B		0 1 1 0

IP = 58, 50, 42, 34, 26, 18, 10, 2, 60, 52, 44, 36, 28, 20, 12, 4

first 16 bits of output: 1110 1001 0110 0100 from IP

Sample IP matrix

3	9	6	2
16	1	15	5
10	11	7	4
8	14	12	13

Input: 1001 0111 1100 0110

Output: 0110 0110 1011 1100

"each # tells you what position bit to grab next"

$P_0 = L_0 R_0$

32 bits left, 32 bits right

A round k takes in $L_{k-1}R_{k-1}$ and produces L_kR_k.

DES 16 rounds

technique called differential cryptanalysis → enough rounds this wasn't effective.

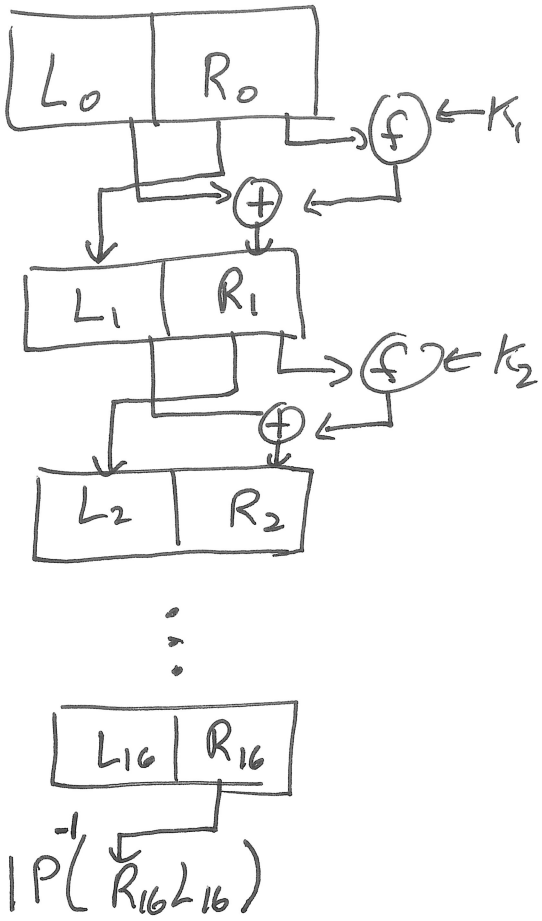
→ Why 16?

(a) long enough secure

(b) not too long slow stuff down.

DES Alg

1. $P_0 = L_0 R_0 = IP(P)$
2. for i from 1 to 16:
   $L_i = R_{i-1}$
   $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
3. $C = IP^{-1}(R_{16} L_{16})$



Note: Make sure that given a random IP matrix, you can calculate the corresponding IP^{-1}.

K is input key.
It's used to generate 16 round keys,
$K_1, K_2, K_3, \ldots K_{16}$
K is 56 unique bits, & checksum bits ($k_1 \ldots k_{64}$)
K_i, each round key is 48 bits

32 bits OUT f(input: 32 bits, input: 48 bits key)

32 bits 32 bits 48 bits

$X \quad f(A, J)$ {

1. $E(A)$ calculate
2. Calculate $B = E(A) \oplus J$, B is 48 bits
3. $B = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$ 6 bits sections
4. for $i = 1$ to 8: → S_i is the i^{th} S-box
   $C_i = S_i(B_i)$ → 6 bit input, → 4 bit output
5. Let $C = C_1 C_2 C_3 \ldots C_8$ (32 bits)
6. Return $P(C)$

S-Boxes

ONLY non-linear component!

$S_i(x_1 x_2 x_3 x_4 x_5 x_6)$

$r = x_1 x_6$

$c = x_2 x_3 x_4 x_5$

$S_1(101110)$

$r = 10 = 2$

$c = 0111 = 7$

$S_1(101110) = S_1(\text{row } 2, \text{col } 7) = \boxed{11}$