LEFT AND RIGHT HALVES: SIMPLE EXAMPLE

See how each half moves during one DES-style Feistel round

DES | CLASS LECTURE

IDEA: In each round: new left = old right; new right = old left XOR the round-function output.

1 Start with 8 bits (a miniature example to learn the rule)

1	2	3	4	5	6	7	8
1	0	1	1	0	0	1	0

LEFT (L_0) = 1011 | **RIGHT (R_0) = 0010**

2 Use the two Feistel round rules

RULE A Copy the old right half

$$L_1 = R_0 = 0010$$

RULE B XOR the old left half with F

$$R_1 = L_0 \text{ XOR } F(R_0, K_1)$$

GIVEN (to keep this example easy):

$F(R_0, K_1) = 0110$

$R_1 = 1011 \text{ XOR } 0110 = 1101$

AFTER ROUND 1

$L_1 = 0010$ | $R_1 = 1101$

ACTUAL DES: MANAGING THE TWO 32-BIT HALVES

Use the same DES values from the IP, E-box, S-box and XOR lectures

START: After the Initial Permutation, split the 64-bit result into two 32-bit halves.

1 Split the permuted 64-bit input

IP(plaintext) = CC00CCFFF0AAF0AA

L_0 = CC00CCFF

R_0 = F0AAF0AA

2 Calculate the round function using the right half

R_0 = F0AAF0AA → $F(R_0, K_1)$ → 234AA9BB

3 Apply the two half-update rules

NEW LEFT = OLD RIGHT

L_1 = F0AAF0AA

NEW RIGHT = OLD LEFT XOR F

CC00CCFF XOR 234AA9BB

R_1 = EF4A6544

ROUND 1 OUTPUT: $L_1 \| R_1$ = F0AAF0AAEF4A6544