DATA ENCRYPTION STANDARD: DECRYPTION

Complete worked example in the exact algorithm order

DES | DECRYPTION

CIPHERTEXT (given)

85E813540F0AB405

SECRET KEY

133457799BBCDFF1

Decryption uses the SAME DES round function...

...but uses the round keys in REVERSE order.

TARGET: RECOVER THE ORIGINAL PLAINTEXT

0123456789ABCDEF

64-bit block • 56-bit effective key • 16 Feistel rounds

THE CORRECT DES DECRYPTION SEQUENCE

Follow these operations in order; the keys run backward

DES | DECRYPTION

01 IP(ciphertext) → 02 Split L and R → 03 Select K16 first → 04 Expand R (E)

05 XOR with key → 06 Eight S-boxes → 07 P-permutation → 08 Update L and R

09 Complete 16 rounds → 10 Swap halves → 11 Inverse IP → 12 Original plaintext

CRITICAL DIFFERENCE

Encryption: K1 → K16 | Decryption: K16 → K1

INITIAL PERMUTATION: SIMPLE BIT ARRANGEMENT

IP always rearranges bit positions; it does not alter bit values

DES | DECRYPTION

IDEA: Select the input bits in the order stated by the IP table.

1 Begin with an 8-bit teaching example

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

2 Apply this miniature IP rule

[2, 4, 6, 8, 1, 3, 5, 7]

RESULT

10110010 → 01001101

Remember: IP moves positions. It does NOT use XOR or the secret key.

ACTUAL DES: APPLY IP TO THE CIPHERTEXT

The ciphertext is permuted first - before beginning Round 1

1 Start with the 64-bit encrypted block

CIPHERTEXT / C

85E813540F0AB405

2 Apply the fixed standard DES IP table

IP positions: 58, 50, 42, 34, 26, 18, 10, 2, ...



64-BIT RESULT

IP(C) = 0A4CD99543423234

This output is the starting block for the 16 decryption rounds.

SPLIT INTO LEFT AND RIGHT HALVES

After the initial permutation, create two 32-bit pieces

IP(C) = 0A4CD99543423234

FIRST 32 BITS: L0

0A4CD995

LAST 32 BITS: R0

43423234

DECRYPTION ROUND 1

Send R0 = 43423234 into F(R0, K16).

Each half has 8 hexadecimal digits = 32 bits.

WHY DES DECRYPTION USES REVERSED KEYS

Same Feistel round structure; only the subkey order changes

DES | DECRYPTION

Original 64-bit key: 133457799BBCDFF1

ENCRYPTION

K1	K2	K3	...	K15	K16

DECRYPTION

K16	K15	K14	...	K2	K1

! **The subkeys are used in reverse order**

Important: Do not reverse the bits of a key; reverse the ORDER of the 16 subkeys.

K16 is used in decryption Round 1. K1 is used in decryption Round 16.

ROUND KEY: FIRST DECRYPTION SUBKEY K16

The same PC-1, rotations and PC-2 produce all sixteen round subkeys

DES | DECRYPTION

1 64-bit key → 2 PC-1: 56 bits → 3 C and D halves → 4 Circular shifts → 5 PC-2: 48 bits

FIRST SUBKEY USED IN DECRYPTION

K16 = CB3D8B0E17F5

Next keys:

Round 2: K15 = BF918D3D3F0A

Round 16: K1 = 1B02EFFC7072

Each subkey contains 48 bits. The complete sequence is shown later.

EXPANSION BOX: THE 4-BIT NEIGHBOR RULE

The standard DES expansion repeats boundary bits to convert 32 bits into 48 bits

RULE: Previous last bit + current 4 bits + next first bit = 6 bits

R0 = 43423234 (8 hexadecimal digits = 32 bits)

Start with the FIRST 4-bit group (hex 4 = 0100)

Last bit of previous nibble	Current four bits	First bit of next nibble
0	0100	0

0 + 0100 + 0 = 001000

FIRST EXPANDED GROUP

001000 (6 bits)

EXPANSION BOX: EXPAND ALL EIGHT GROUPS

Compute each 6-bit group using both neighboring 4-bit groups

DES | DECRYPTION

GROUP	PREVIOUS	CURRENT FOUR	NEXT	SIX-BIT OUTPUT
1	0	0100	0	001000
2	0	0011	0	000110
3	1	0100	0	101000
4	0	0010	0	000100
5	0	0011	0	000110
6	1	0010	0	100100
7	0	0011	0	000110
8	1	0100	0	101000

FINAL 48-BIT RESULT

206A041A41A8

EXPANSION BOX: STANDARD DES E TABLE

Official E selection positions confirm the same 48-bit output

DES | DECRYPTION

Each entry gives the original bit position to copy.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

R0 = 43423234

32 bits

E(R0) = 206A041A41A8

48 bits

XOR OPERATION: SIMPLE RULE

XOR returns 1 when two bits differ; 0 when the bits match

0 XOR 0	0 XOR 1	1 XOR 0	1 XOR 1
= 0	= 1	= 1	= 0

1 Calculate an 8-bit XOR manually

```
  1 0 1 1 0 0 1 0
  0 1 1 0 1 0 0 1
-----------------
  1 1 0 1 1 0 1 1
```

ANSWER

10110010 XOR 01101001 = 11011011

DES ROUND 1: XOR WITH THE SUBKEY

Combine the 48-bit expanded right half with the 48-bit K16

EXPANDED RIGHT HALF 206A041A41A8

ROUND 1 SUBKEY K16 CB3D8B0E17F5

1 XOR the six corresponding hexadecimal bytes

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
20 XOR CB = EB	6A XOR 3D = 57	04 XOR 8B = 8F	1A XOR 0E = 14	41 XOR 17 = 56	**A8 XOR F5 = 5D**

FINAL XOR OUTPUT (48 BITS)

206A041A41A8 XOR CB3D8B0E17F5

= EB578F14565D

S-BOX: ONE LOOKUP IN DES DECRYPTION

Six bits select one row and one column; the output has four bits

FIRST SIX BITS: 111010

bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
1	1	1	0	1	0

OUTER BITS: ROW

10 = 2

INNER FOUR: COLUMN

1101 = 13

Standard DES S1[2, 13] = 10 (decimal).

S1 OUTPUT

10 decimal = 1010 binary

ALL EIGHT S-BOXES: COMPLETE ROUND 1

48 bits become 32 bits after eight independent S-box lookups

48-BIT INPUT **EB578F14565D**

S-BOX	6-BIT INPUT	ROW	COLUMN	DECIMAL	4-BIT RESULT
S1	111010	2	13	10	**1010**
S2	110101	3	10	7	**0111**
S3	011110	0	15	8	**1000**
S4	001111	1	7	3	**0011**
S5	000101	1	2	2	**0010**
S6	000101	1	2	4	**0100**
S7	011001	1	12	2	**0010**
S8	011101	1	14	9	**1001**

JOIN ALL 4-BIT RESULTS **A7832429 (32 bits)**

P-PERMUTATION: REARRANGE THE 32 S-BOX BITS

P changes bit positions but does not increase the bit count

DES | DECRYPTION

32-BIT S-BOX RESULT

A7832429

STANDARD P TABLE (read left to right, top to bottom)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

F(R0, K16) = P(A7832429)

P-PERMUTATION OUTPUT

C8C04F98

FEISTEL ROUND: SIMPLE LEFT/RIGHT UPDATE

Each DES round applies the SAME two half-update equations

RULE: New left = old right. New right = old left XOR F(old right, key).

OLD LEFT L0

1011

OLD RIGHT R0

0010

Given this miniature example: F(R0, key) = 0110

NEW LEFT L1 = R0

0010

NEW RIGHT R1 = 1011 XOR 0110

1101

ACTUAL DECRYPTION ROUND 1: UPDATE THE HALVES

The first decryption round uses K16, not K1

OLD LEFT L0

0A4CD995

OLD RIGHT R0

43423234

F(R0, K16) = C8C04F98

1 **Now apply the two update equations**

L1 = R0

43423234

R1 = L0 XOR F

C28C960D

0A4CD995 XOR C8C04F98 = C28C960D

DECRYPTION ROUND 2: ONE MORE EXAMPLE

Use the output from Round 1 as the input to Round 2

ROUND 2 OPERATION	ACTUAL HEXADECIMAL VALUE
Key K15	BF918D3D3F0A
Expand R1	E054594AC05B
XOR with K15	5FC5D477FF51
Apply S1 ... S8	B2E88D3C
P permutation	5B81276E

L2 = R1

C28C960D

R2 = L1 XOR F

18C3155A

COMPLETE DECRYPTION: ROUNDS 1 THROUGH 8

Use the same Feistel equations for every successive round

ROUND	48-BIT SUBKEY	NEW LEFT (32 BITS)	NEW RIGHT (32 BITS)
01	CB3D8B0E17F5	43423234	**C28C960D**
02	BF918D3D3F0A	C28C960D	**18C3155A**
03	5F43B7F2E73A	18C3155A	**75BD1858**
04	97C5D1FABA41	75BD1858	**C5783C78**
05	7571F59467E9	C5783C78	**B7D5D7B2**
06	215FD3DED386	B7D5D7B2	**247CC67A**
07	B1F347BA464F	247CC67A	**D5694B90**
08	E0DBEBEDE781	D5694B90	**064ABA10**

The output halves of one round become the inputs of the next.

COMPLETE DECRYPTION: ROUNDS 9 THROUGH 16

The final decryption round uses the original encryption subkey K1

ROUND	48-BIT SUBKEY	NEW LEFT (32 BITS)	NEW RIGHT (32 BITS)
09	F78A3AC13BFB	064ABA10	**E967CD69**
10	EC84B7F618BC	E967CD69	**8A4FA637**
11	63A53E507B2F	8A4FA637	**77220045**
12	7CEC07EB53A8	77220045	**A25C0BF4**
13	72ADD6DB351D	A25C0BF4	**CC017709**
14	55FC8A42CF99	CC017709	**EF4A6544**
15	79AED9DBC9E5	EF4A6544	**F0AAF0AA**
16	1B02EFFC7072	F0AAF0AA	**CC00CCFF**

Final decryption subkey: K1 = 1B02EFFC7072.

AFTER ROUND 16: SWAP THE TWO HALVES

DES performs the final half-swap before applying inverse IP

DES | DECRYPTION

FINAL LEFT HALF L16

F0AAF0AA

FINAL RIGHT HALF R16

CC00CCFF

Swap the order of the two halves

↓

R16 || L16

CC00CCFFF0AAF0AA

This 64-bit block is the INPUT to the final IP inverse.

INVERSE INITIAL PERMUTATION: SIMPLE EXAMPLE

IP inverse restores the bit positions that an IP rearranged

REMEMBER: Inverse IP is a rearrangement - not XOR and not a secret-key operation.

AFTER IP

10110010 → 01001101

1 **Use the matching inverse ordering**

[5, 1, 6, 2, 7, 3, 8, 4]

2 **Select those positions from the permuted bits**

INVERSE IP RESTORES

01001101 → 10110010

FINAL INVERSE IP: RECOVER THE ORIGINAL TEXT

Apply DES IP inverse to the swapped 64-bit round result

DES | DECRYPTION

64-BIT INPUT TO INVERSE IP

CC00CCFFF0AAF0AA

Apply the standard 64-position IP inverse table



RECOVERED PLAINTEXT

0123456789ABCDEF

DES-DEC(85E813540F0AB405, 133457799BBCDFF1) = 0123456789ABCDEF

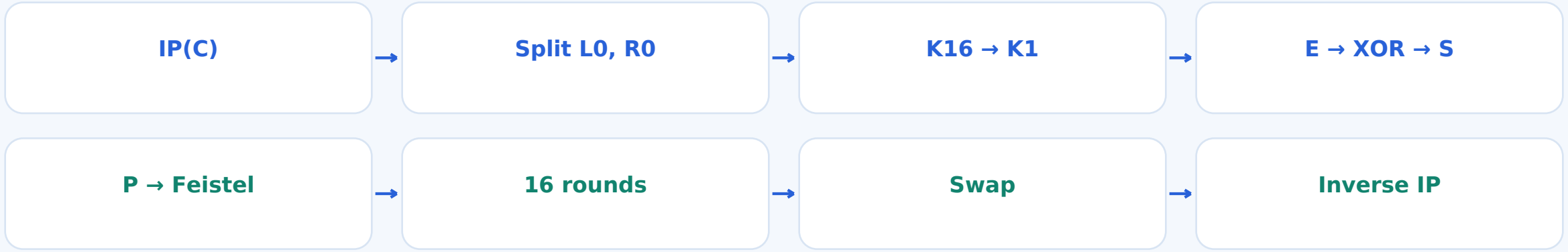
DES DECRYPTION: FINAL REVIEW

One coherent example from ciphertext back to plaintext

ENCRYPTED INPUT

85E813540F0AB405

The key sequence is reversed; the round function stays the same.



ORIGINAL PLAINTEXT

0123456789ABCDEF

DES is historically important but insecure for modern deployment (56-bit effective key).