Data Encryption Standard (DES) and Simplified DES (SDES)

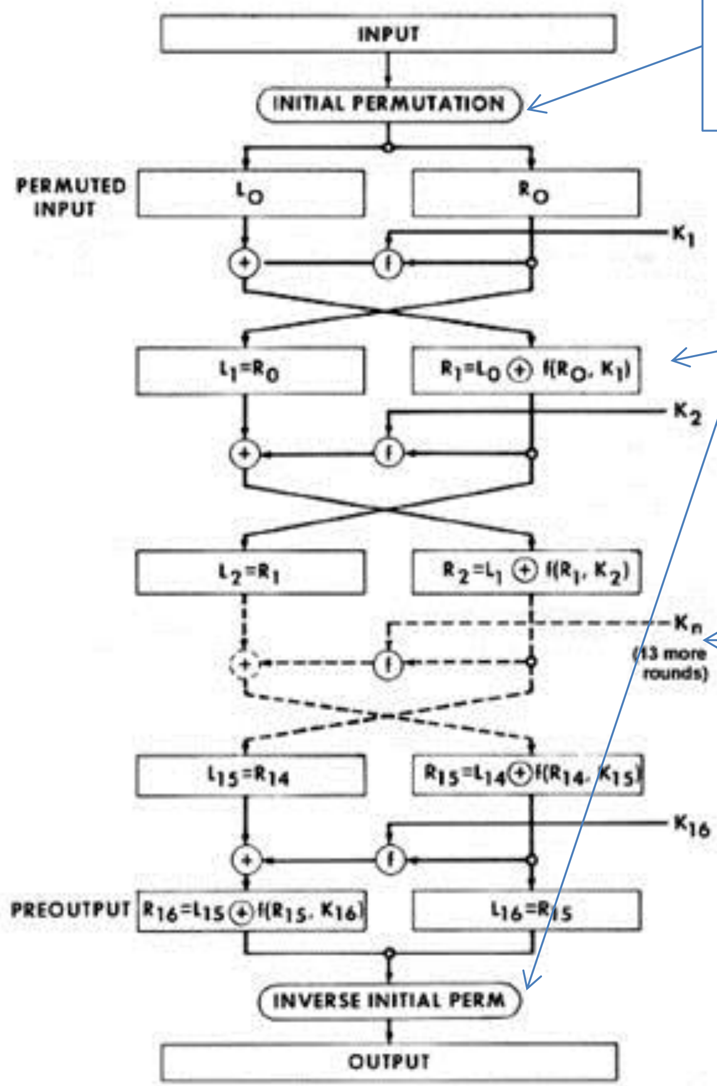
MTH 440

A brief history

- Created by Horst Feistel from IBM
- Named: Dataseal -> Demonstration Cipher -> Demon -> Lucifer
- 1973 NBS (now NIST) held a public competition, Lucifer won, renamed DES (Data Encryption Standard)
- Controversy (collaboration with NSA, key size, secrecy behind design of S-boxes)
- DES became the code provided by 99% of the companies selling equipment using encryption.
- EFF (Electronic Frontier Foundation) in 1998 designed the DES Cracker form $250,000 which broke a DES key in 3 days. Using a network of computers this was reduced to 22 hours 15 minutes in 1999.
- Triple DES: $3DES(x)= E(K_{3,}(D(K_2(E(K_{1,}x)))))$
- New competition announced AES selected in 2002.

DES specifications

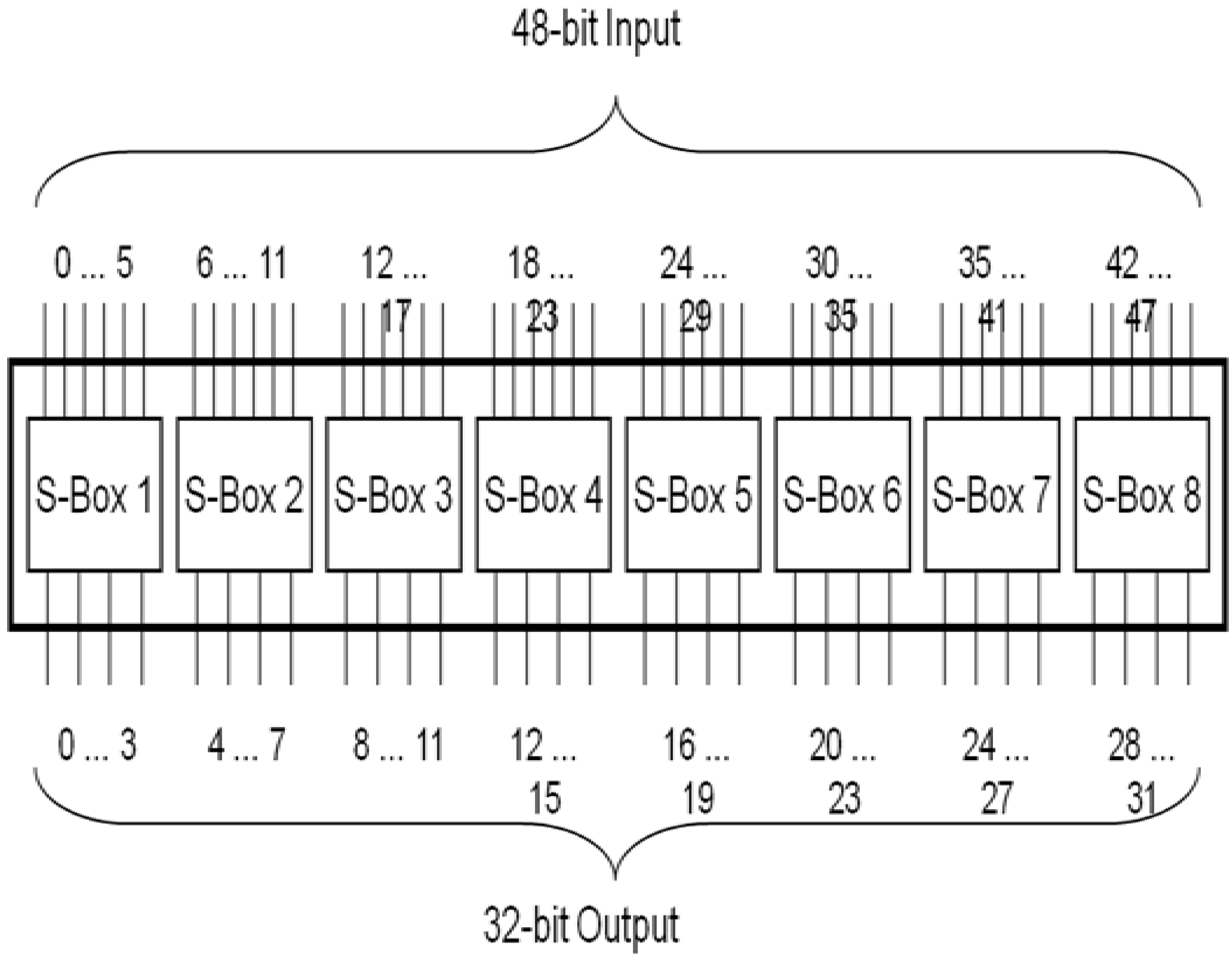
- 64-bit block cipher
- 56-bit key (the key is technically 64 bits but 8 are used as parity bits for error correcting making the effective security equivalent to a 56-bit key)
- 16 round Feistel cipher
- The round function requires 48 bits of input
  - Uses 8 S-boxes of 6-bits each
- Different 48 subkey used for each round



Before beginning an initial permutation of the bits in the plaintext block are applied (IP). The inverse permutation is applied at the end before the ciphertext output.
Note: IP is not secret

The input block is 64 bits so each half is 32 bits. However since the round function requires a 48 bit input an "expansion" function is applied to the 32 bits to expand it to 48 bits. The output of the function is 32 bits.

A different 48 bit key is used in each
Of the 16 rounds. Hence the initial 56 bit
Key is used to generate the 16 "sub keys".



48-bit Input
0 ... 5
6 ... 11
12 ... 17
18 ... 23
24 ... 29
30 ... 35
35 ... 41
42 ... 47
S-Box 1
S-Box 2
S-Box 3
S-Box 4
S-Box 5
S-Box 6
S-Box 7
S-Box 8
0 ... 3
4 ... 7
8 ... 11
12 ... 15
16 ... 19
20 ... 23
24 ... 27
28 ... 31
32-bit Output

10

S1(101100)=**0010**

Input bits 1 and 6 | Input bits 2 thru 5

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

Figure 3-9. Table of 4-bit outputs of S-box 1 (bits 1 thru 4)

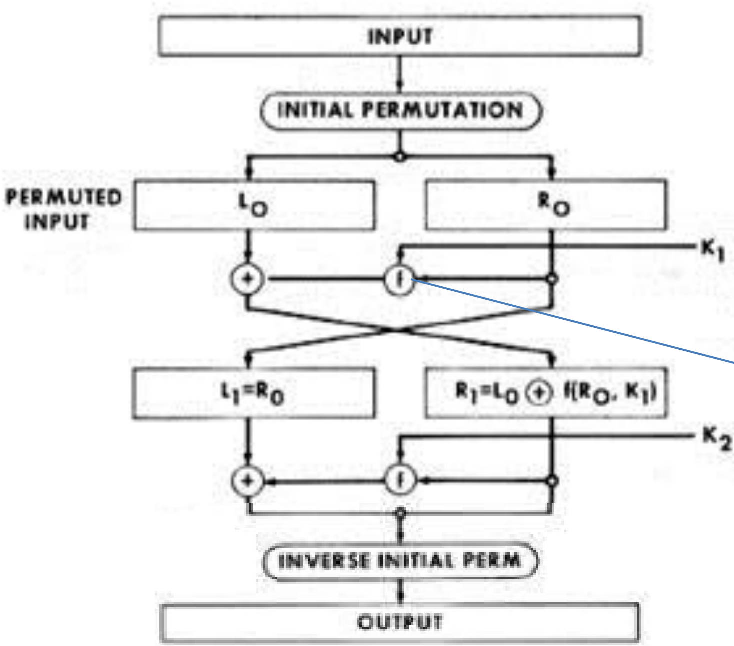
Input bits 7 and 12 | Input bits 8 thru 11

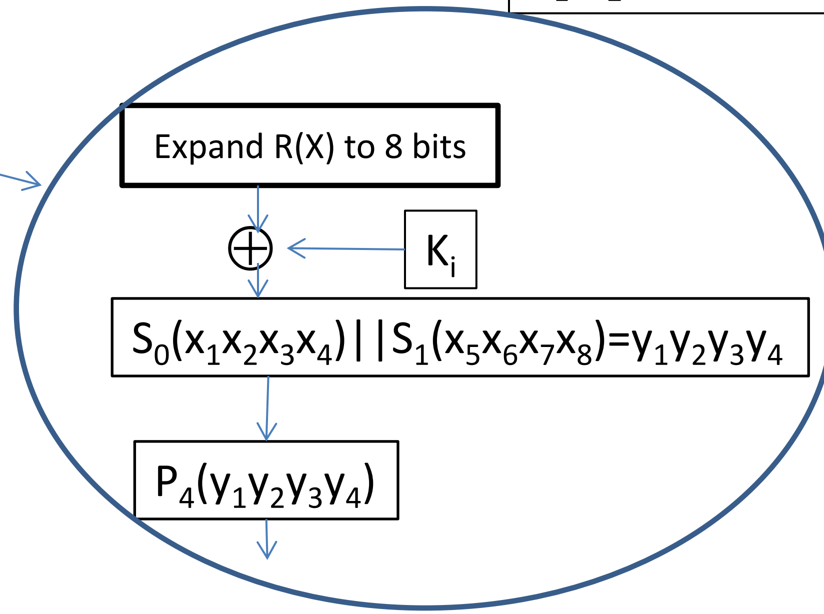
↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1111	0001	1000	1110	0110	1011	0011	0100	1001	0111	0010	1101	1100	0000	0101	1010
01	0011	1101	0100	0111	1111	0010	1000	1110	1100	0000	0001	1010	0110	1001	1011	0101
10	0000	1110	0111	1011	1010	0100	1101	0001	0101	1000	1100	0110	1001	0011	0010	1111
11	1101	1000	1010	0001	0011	1111	0100	0010	1011	0110	0111	1100	0000	0101	1110	1001

Figure 3-10. Table of 4-bit outputs of S-box 2 (bits 5 thru 8)

K: 10 bit key

K used to generate two 8-bit sub-keys K_1,K_2





Key generation:

$K_1(k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10})=k_1k_7k_9k_4k_8k_3k_{10}k_6$

$K_2(k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10})=k_8k_3k_6k_5k_{10}k_2k_9k_1$

Initial Permutation

$IP(x_1x_2x_3x_4x_5x_6x_7x_8) = x_2x_6x_3x_1x_4x_8x_5x_7$

Expansion Function

$EP(x_1x_2x_3x_4)=x_4x_1x_2x_3x_2x_3x_4x_1$

S_0		x_2 x_3	0 0	0 1	1 0	1 1
x_1	x_4					
0	0		01	00	11	10
0	1		11	10	01	00
1	0		00	10	01	11
1	1		11	01	11	10

S_1		x_2 x_3	0 0	0 1	1 0	1 1
x_1	x_4					
0	0		00	01	10	11
0	1		10	00	01	11
1	0		11	00	01	00
1	1		10	01	00	11

$P_4(x_1x_2x_3x_4) = x_2x_4x_3x_1$

$IP^{-1} = x_4x_1x_3x_5x_7x_2x_8x_6$

SDES summary

1. Expand K into K_1,K_2
2. $IP(x)=L(x) \mid\mid R(x)$
3. Find $EP(R(x)) \oplus K_1 = x_1x_2x_3x_4 \ x_5x_6x_7x_8$
4. Apply S-boxes $S_0(x_1x_2x_3x_4) \mid\mid S_1(x_5x_6x_7x_8) = y_1y_2y_3y_4$
5. Compute $L'(x)= L(X) \oplus P_4(y_1y_2y_3y_4)$ (Note $R'(X) = R(X)$)
6. Switch $L'(X)$ and $R'(X)$ to get new input $R'(X) \mid\mid L'(X)$
7. Repeat 3-5 with new input for the 2nd round
8. Apply the inverse permutation to the output of round 2 to get the final answer.

Note: To decipher use the same algorithm, but use K_2 first, then K_1 (still do the IP at the beginning and IP^{-1} at the end)

- Try it – worksheet