Systems and Network Security (NETW-1002)

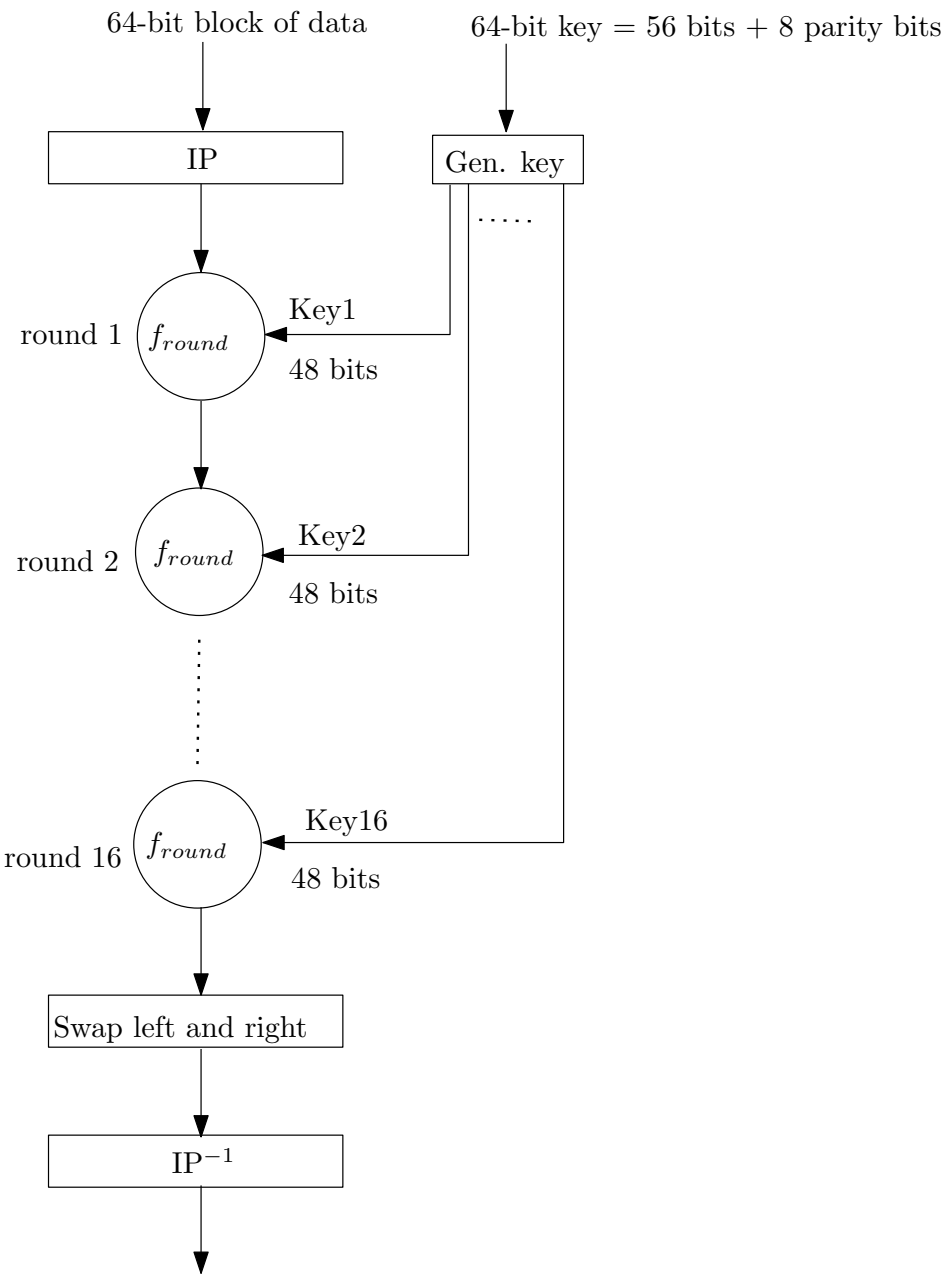
Dr. Mohamed Abdelwahab Saleh

IET-Networks, GUC

Spring 2017

TOC

DES Function Block



DES–IP and IP^{-1}

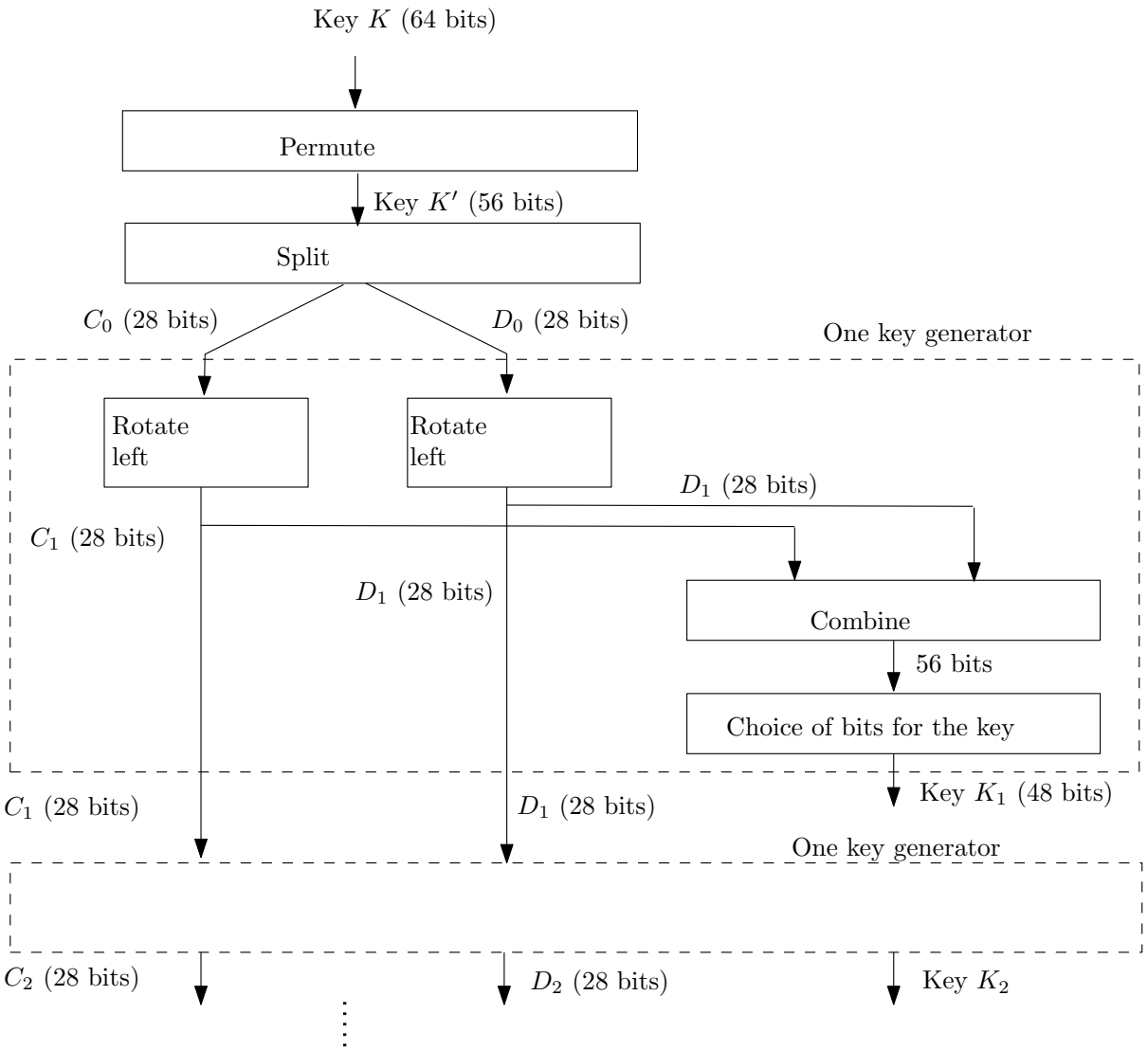
IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Genaration of Keys



Repeat to get the rest of the 16 keys

Genaration of Keys–Tables (Permutation and Key Choice)

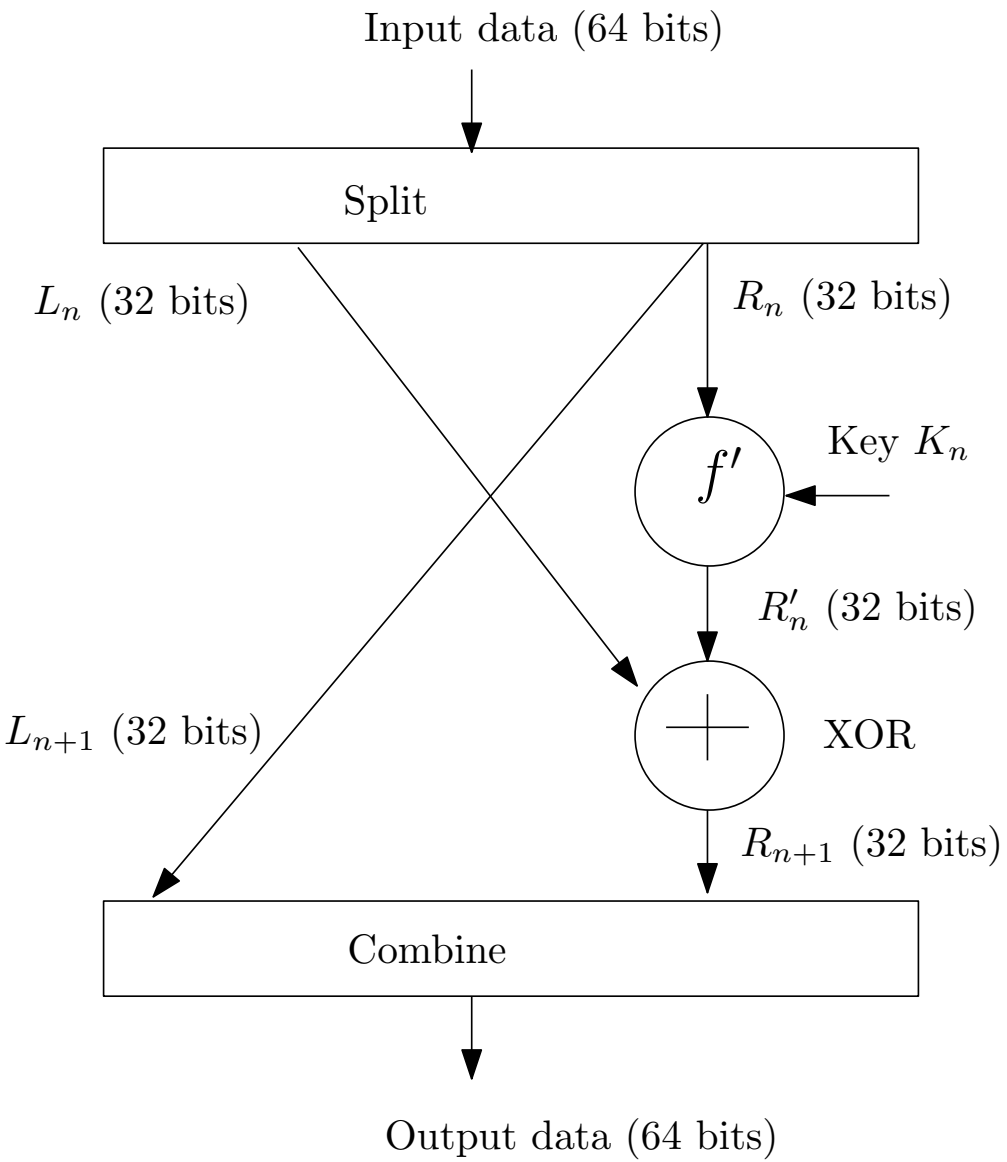
P_K

57	49	41	33	25	17	9	C_0
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	D_0
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

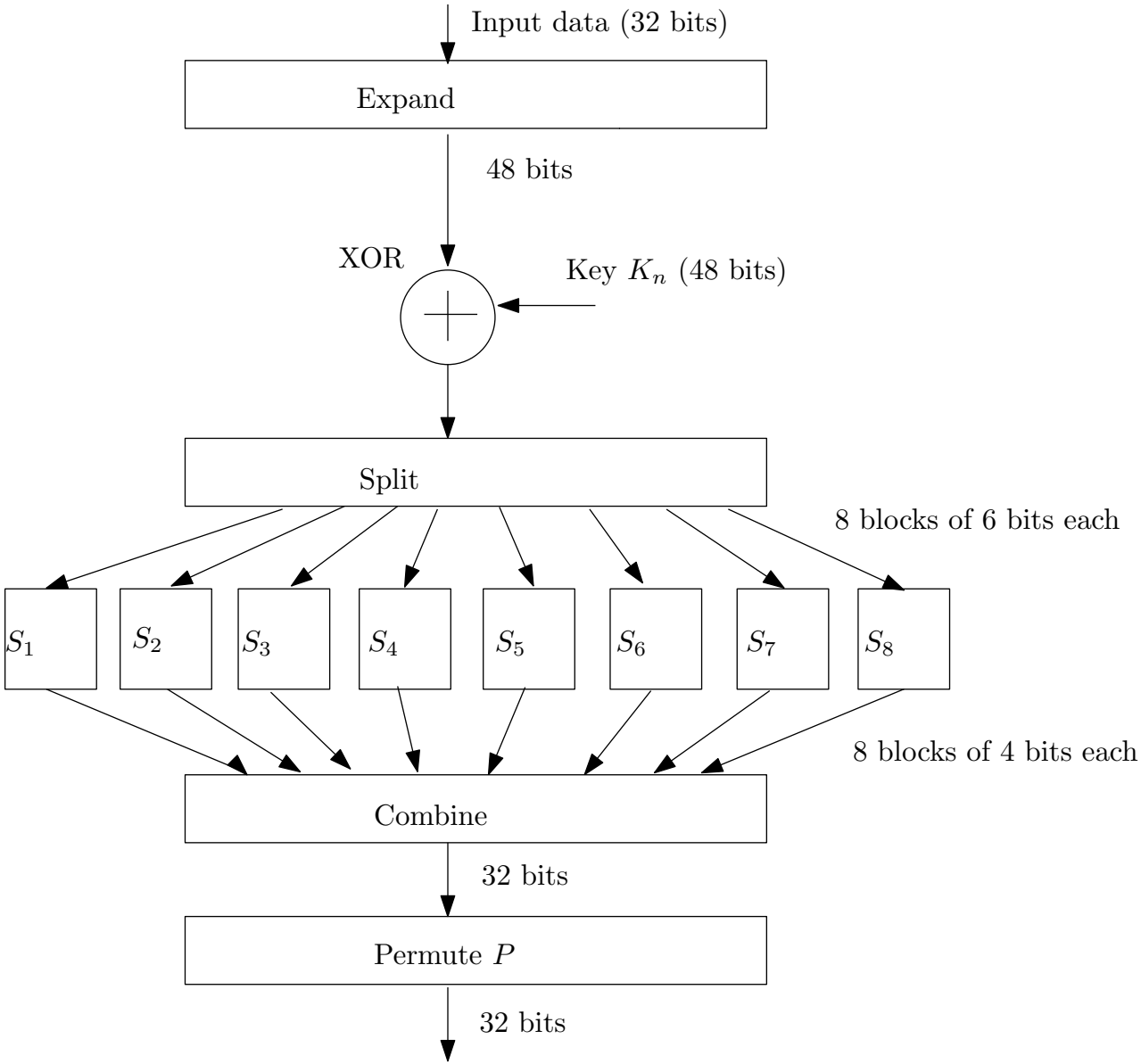
Key choice

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Encryption Function



f'


Input data (32 bits)
Expand
48 bits
XOR
Key K_n (48 bits)
Split
8 blocks of 6 bits each
S_1
S_2
S_3
S_4
S_5
S_6
S_7
S_8
8 blocks of 4 bits each
Combine
32 bits
Permute P
32 bits

f'–Expansion Table

Expansion table

32	1	2	3	4	5
4	5	6	7	8	9
8	9	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	1

f'–S-boxes

How to use the tables?

– We start by a block of six bits: $b_1.b_2.b_3.b_4.b_5.b_6$

– The value of $b_1.b_6$, in decimal, determine a row r in the table.

– The value of $b_2.b_3.b_4.b_5$, in decimal, determine a column c in the table.

– The output of the S-box is the binary value at row r and column c.

S_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

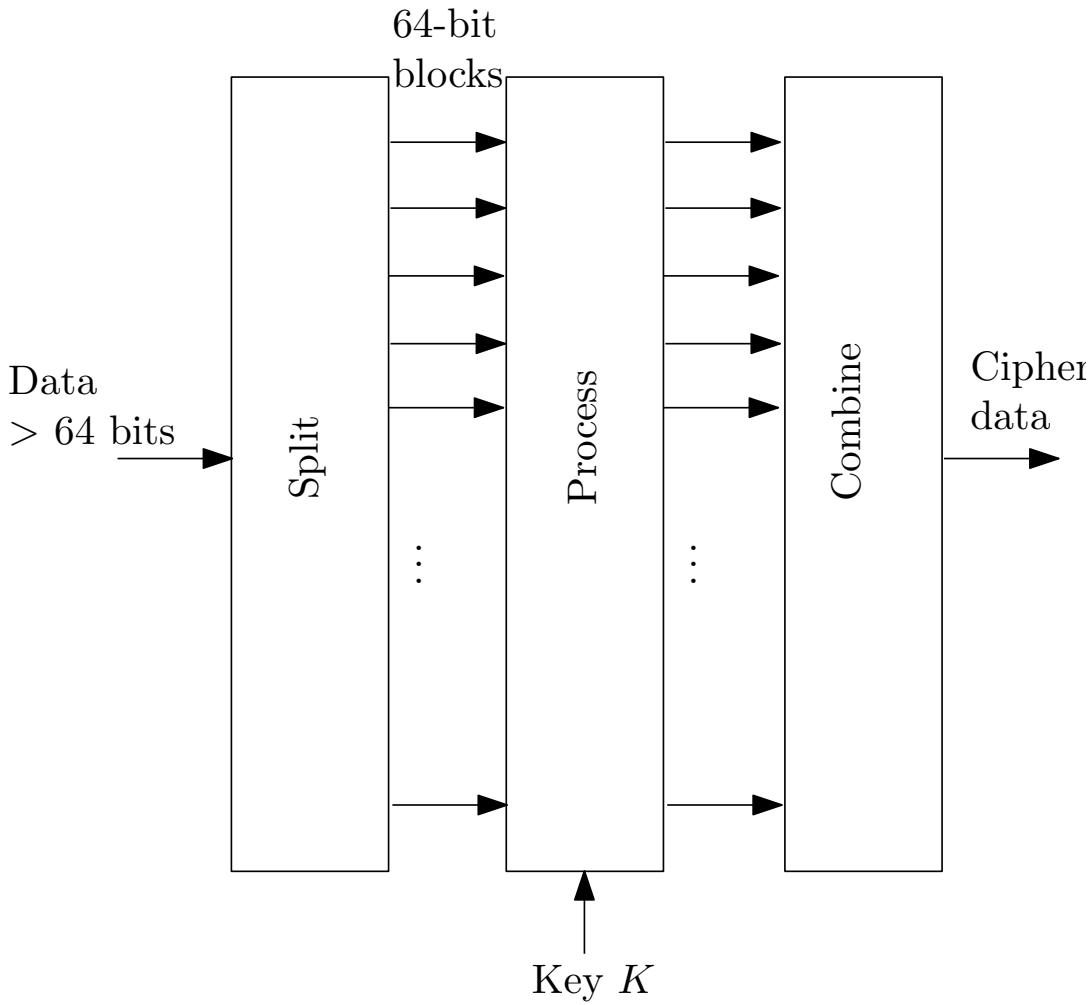
f'–Permutation Table

The permutation table used to permute the combined output bits from the S-boxes is given below:

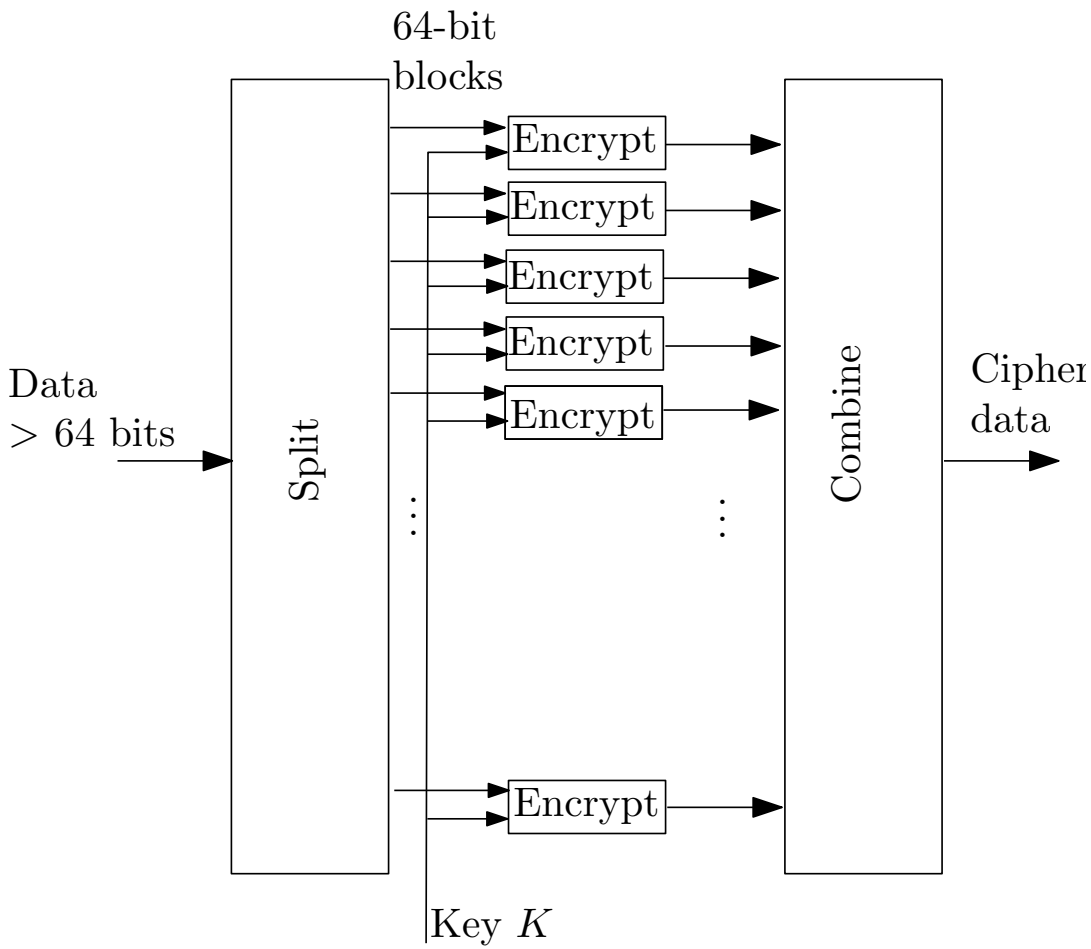
Permutation table P

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

DES for Input Larger than 64 bits



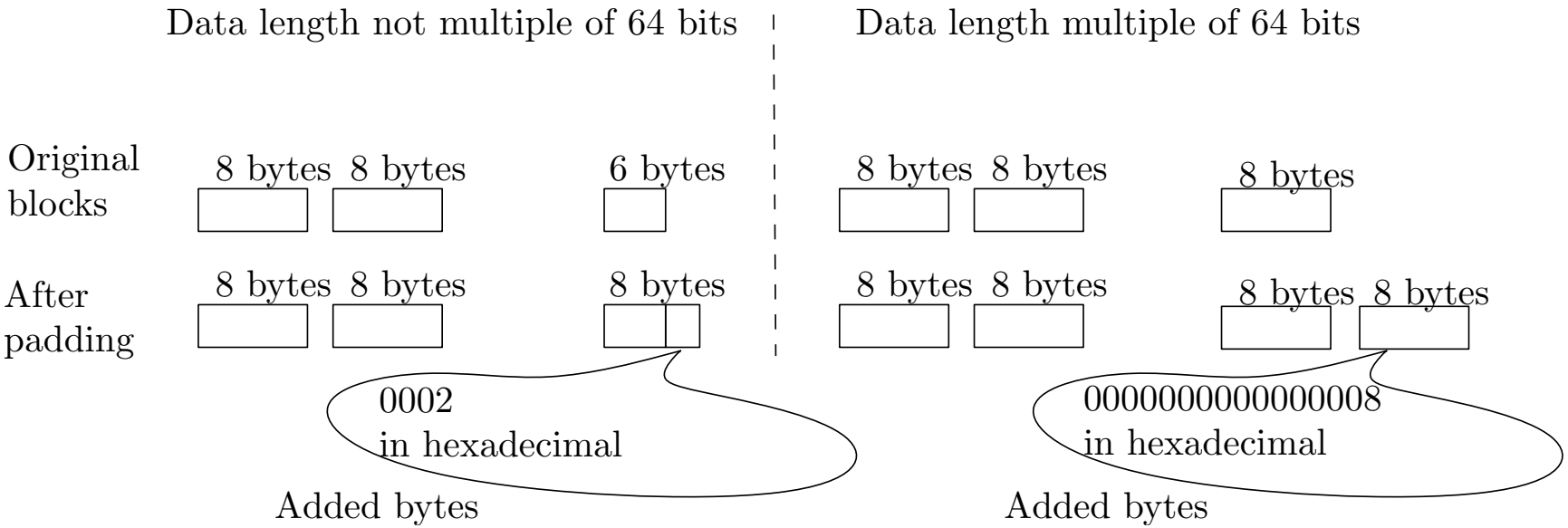
Electronic Code Book Mode



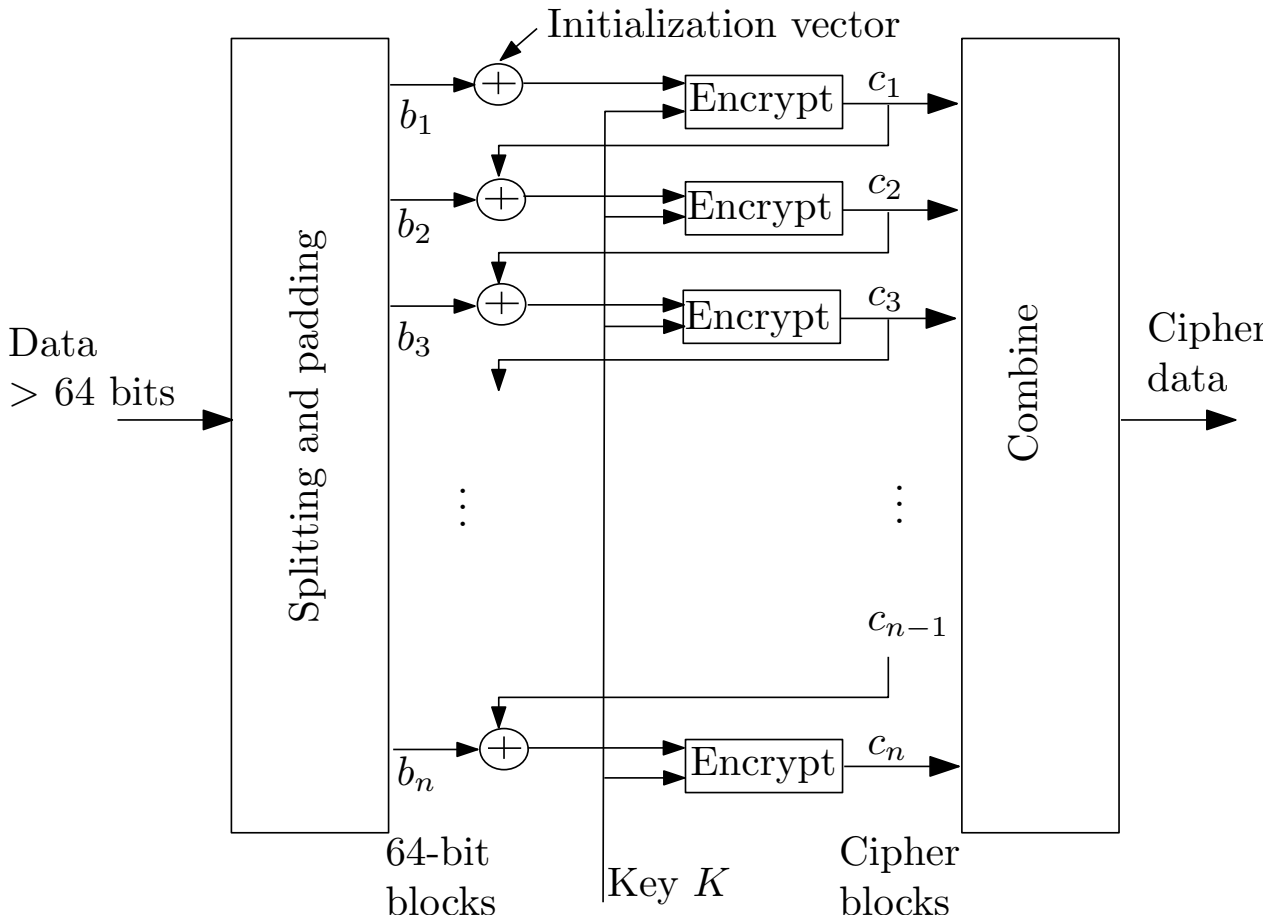
Padding

- There exists several methods for message padding.
- The most common of which is to add zeros at the end of the last block and store the number of added bytes as the last byte in the new 64-bit block.
- At the receiver side, reading the last byte of the decrypted data, one would now how many bytes to remove from the data in order to obtain the original message.
- Here, a confusion may arise in case no padding bytes were added, since, in this case, the last byte of the decrypted data will be an original byte of the message.
- The solution is to *always* add padding bytes, even if the message length is a multiple of 64 bits.
- Therefore, in the case of DES, the number of padding bytes will range from one to eight.

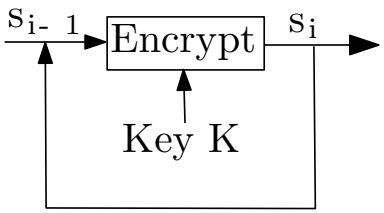
Padding–Example



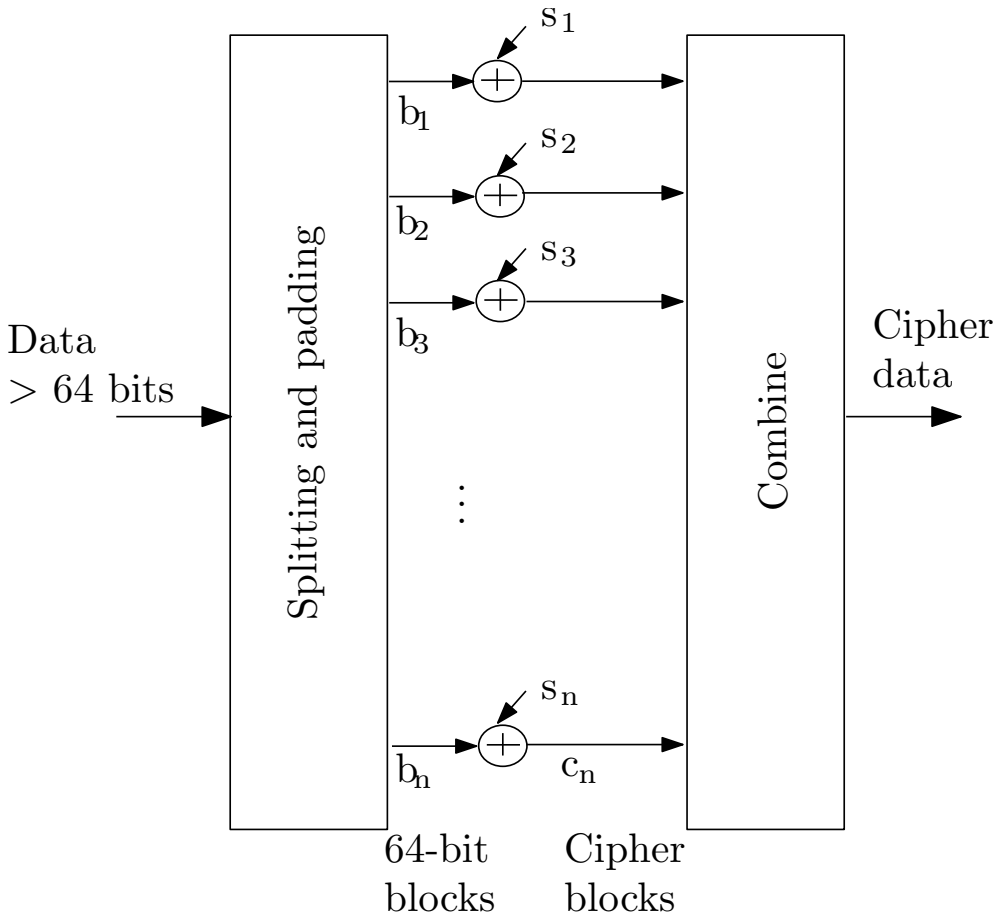
Cipher Block Chaining Mode



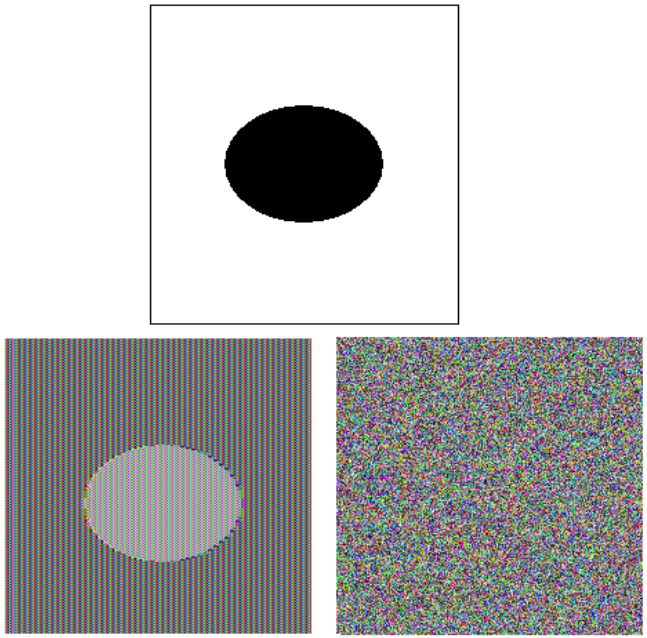
Output Feedback Mode



s_0 = intialization vector (64 bits)
s_1 = DES encryption of s_0 by K
s_2 = DES encryption of s_1 by K
$\vdots$



Difference between ECB and CBC



The original picture is at the top. At the bottom, the picture at the left is its ECB encryption, while the one at the right is the CBC encryption.