Cryptography

Symmetric Encryption – DES

Malay Bhattacharyya

Assistant Professor

Machine Intelligence Unit
and
Centre for Artificial Intelligence and Machine Learning (CAIML)
Indian Statistical Institute, Kolkata

April 17, 2021

A brief history of encryption standards

1960s: Setting up a research project (led by H. Feistel) in computer cryptography by IBM to standardize encryption methods
1971: Development of LUCIFER algorithm by IBM
1972: Attempt to develop a marketable commercial encryption product on a single chip in the Tuchman-Meyer project of IBM
1973: National Bureau of Standards (NBS) asked for the proposals of a national cipher standard
1973: The results of Tuchman-Meyer project submitted to NBS
1977: The algorithm of Tuchman-Meyer project is adopted as the Data Encryption Standard (DES)
1994: National Institute of Standards and Technology (NIST) reaffirmed DES for federal use for another five years
1999: NIST issued a new version of its standard (FIPS PUB 46-3) and proposed triple DES

Basics

DES is a Feistel-type block cipher having substitution-permutation network pattern.

The basic features of DES are as follows:

- **Block size:** 64 bits
- **Key size:** 56 bits (+ 8 bits for parity check)
- **Rounds:** 16
- **Structure:** Balanced Feistel network

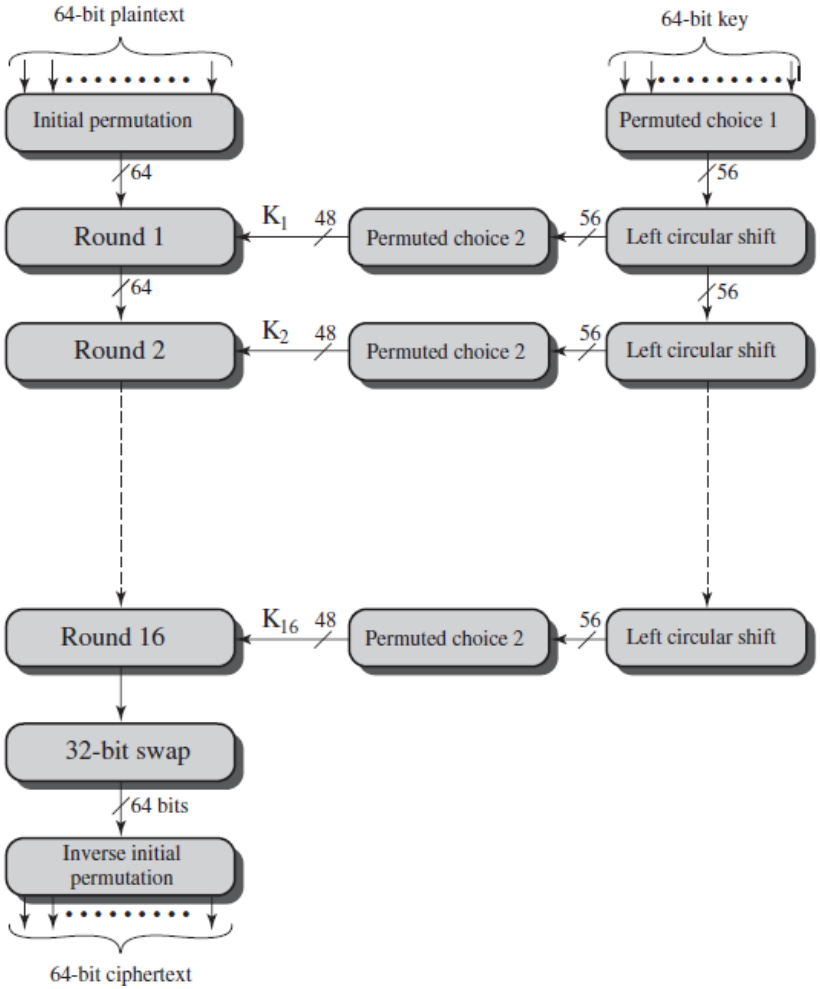
The working principle of DES

The DES encryption algorithm works in four phases as follows:

1. Initial permutation
2. The substitution-permutation rounds
3. Final swapping
4. Inverse initial permutation

Note: The decryption is just the reverse of encryption in DES.

The working principle of DES



Visualization of the DES encryption scheme

The initial permutation operation

The initial permutation is defined by a table that corresponds to 64 bits (numbered from 1 to 64). The entries in this table contain a permutation of the numbers from 1 to 64. Each entry indicates the position of a numbered input bit to form the output of 64 bits.

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

An example of initial permutation table

The inverse initial permutation operation

An example of the inverse initial permutation table is shown below.

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

An example of inverse initial permutation table

The substitution-permutation round of DES

The substitution-permutation rounds are further separated into four subphases as follows:

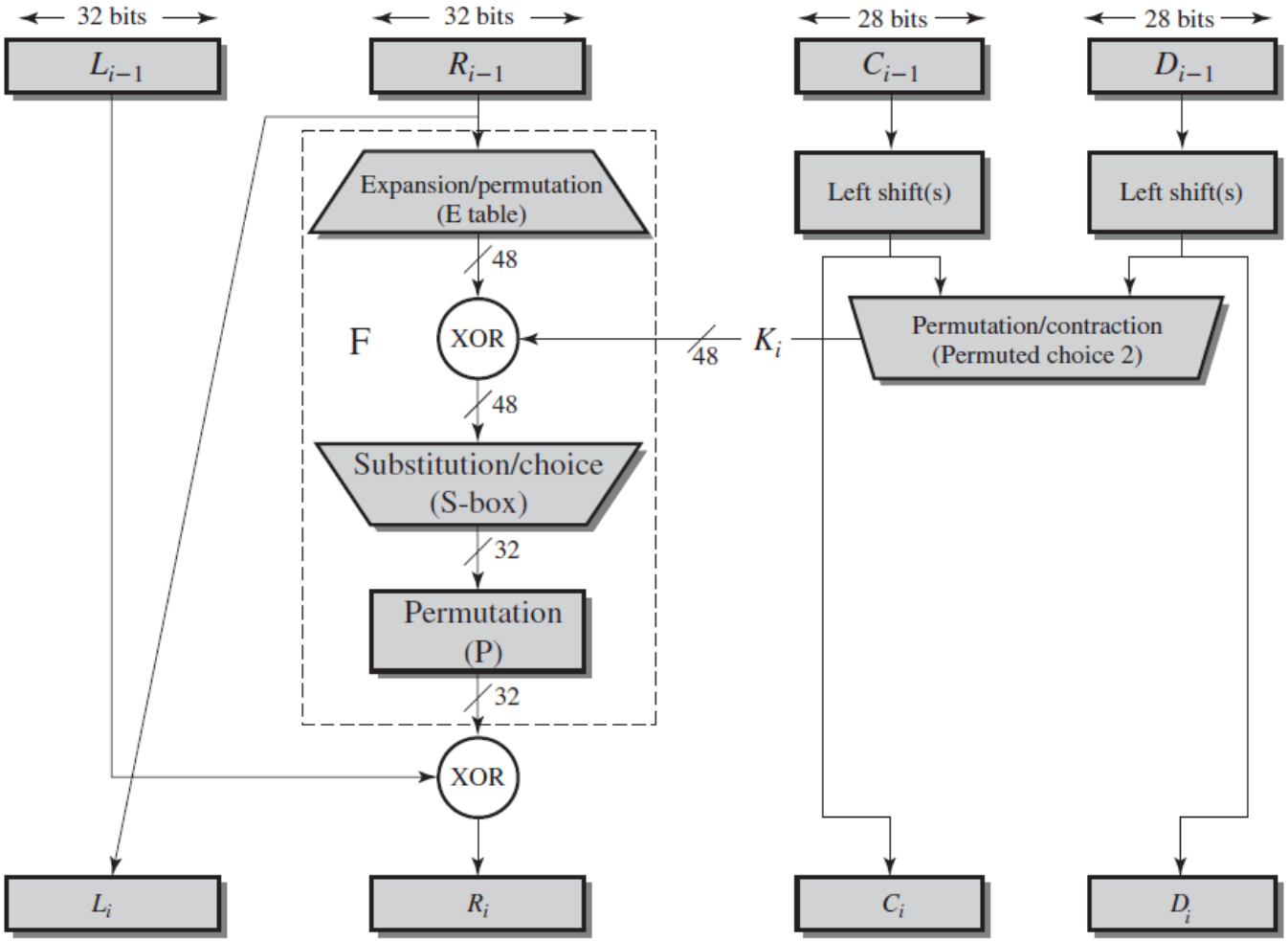
1. Expansion (E table)
2. XOR operation
3. Substitution (S-box)
4. Permutation (P-box)

As in any classic Feistel cipher, the overall processing at each round can be summarized using the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

The substitution-permutation round of DES



A single round of the DES algorithm explained

The E table operation

In E table, the 32 bits of input are split into groups of 4 bits and then converted to groups of 6 bits by taking the outer bits from the two adjacent groups.

For example, consider that the input bits to an E table are given as shown below

1011 0100 0000 0101 0010 1101 1001 0110

Then, this will become the following

010110 101000 000000 001010 100101 011011 110010 101101

The E table operation

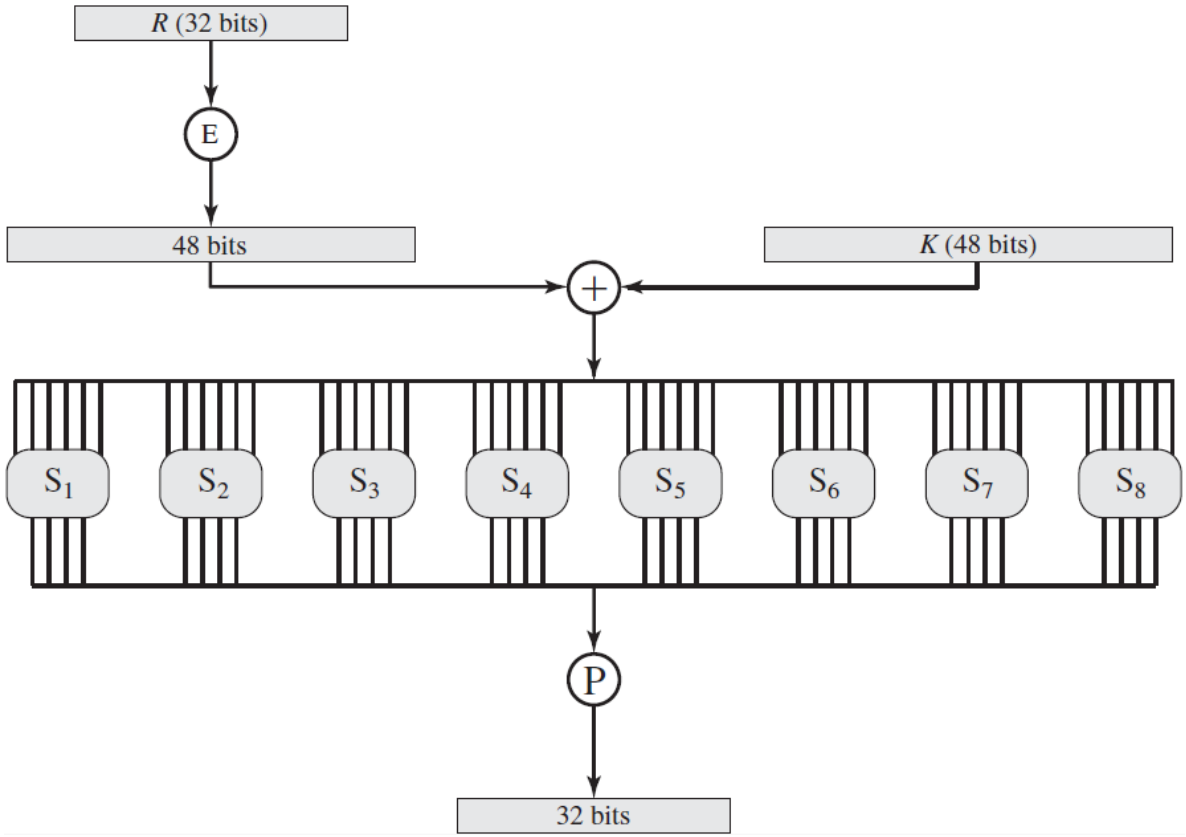
Each row of the E table (showing the bit position) includes two more elements (corresponding positions) from the previous and next row. The last bit of previous row and first bit of next row come to the first and last position of the current row, respectively.

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

An example of expansion permutation table

The S-box operation

All the eight S-boxes (performing general reversible substitution) accept 6 bits as input and produces 4 bits as output.



Calculating the value of $F(R_{i-1}, K_i)$

The S-box operation

Given a 6-bit input, the 4-bit output is generated by selecting the element in the following table that has the row and column indices denoted by the outer (the first and last) two bits and inner four bits, respectively.

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

Hence, the 6-bit input 110110 will be converted to the 4-bit element (output) present in the cell at row 10 and column 1011, i.e. 0101.

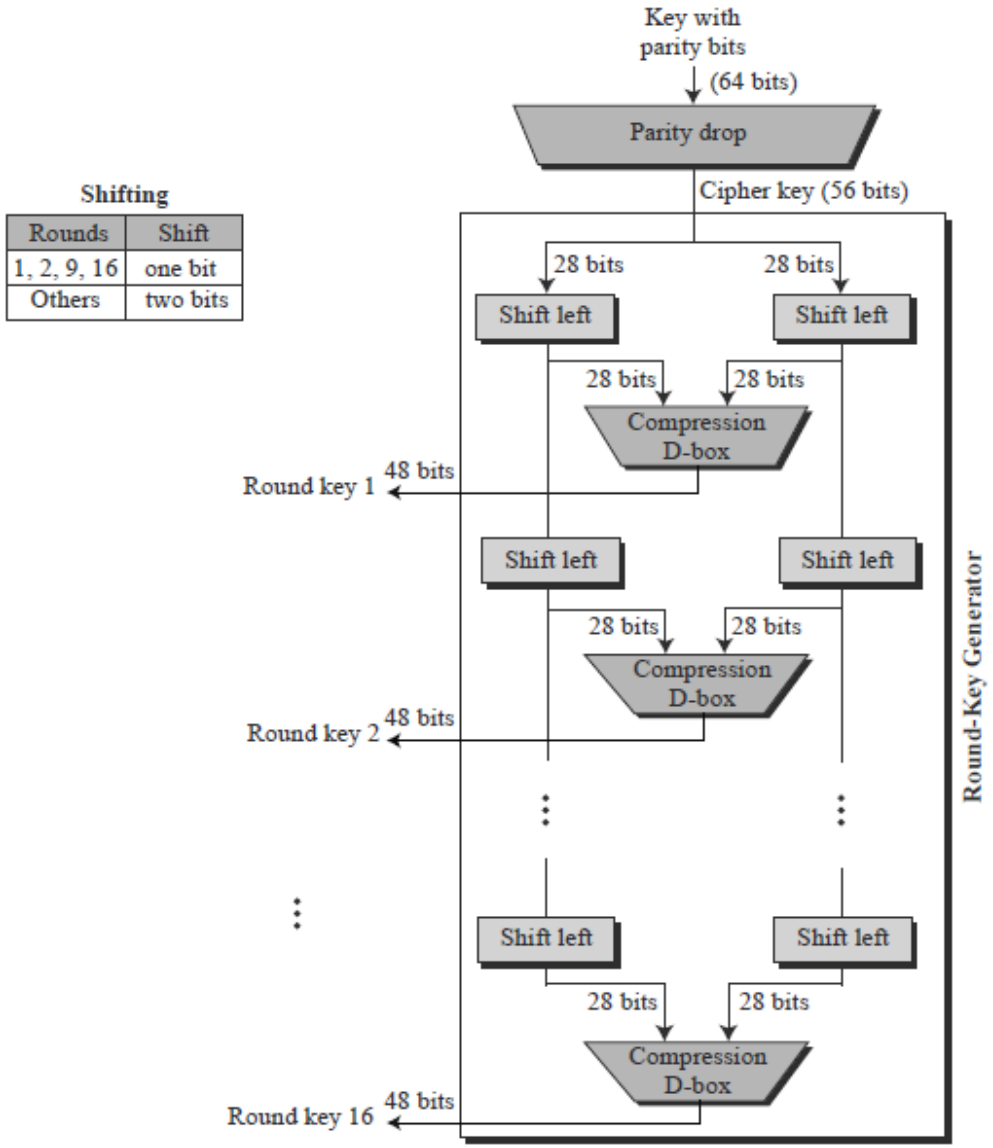
The permutation function

The permutation function is applied with the help of a permutation table (P-box) as shown below.

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

An example of permutation function

The key generation in DES



Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

The key generation in DES

The key constitutes of a total 64 bits but only 56 bits are taken by ignoring every eighth bit. The key is used as follows:

1. The key is first subjected to a permutation governed by a permutation table. The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0.
2. At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift or (rotation) of 1 or 2 bits, as governed by a schedule.
3. These shifted values serve as input to the next round. They also serve as input to another permutation table, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

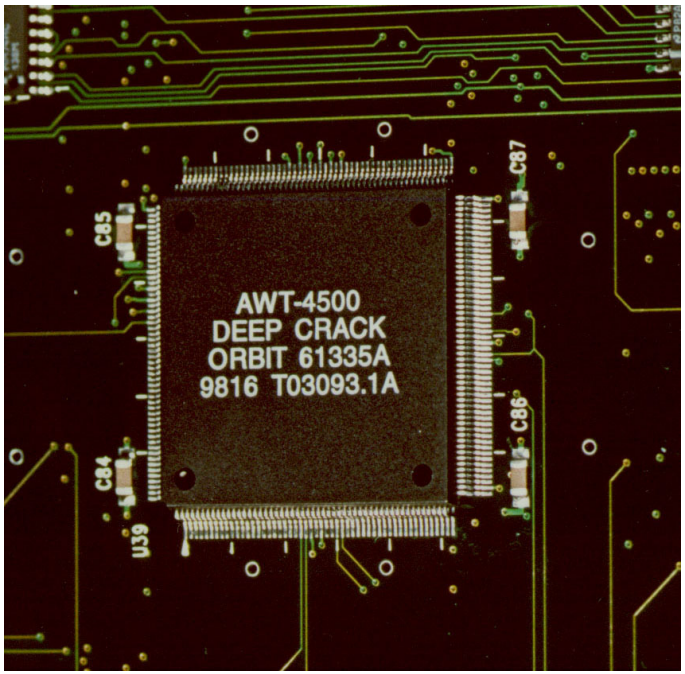
Breaking the DES

The strength of DES depends on two major issues:

1. **Key size:** With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16} keys.
2. **Nature of the algorithm:** The characteristics of the DES algorithm can be guessed by breaking the eight substitution tables, or S-boxes, that are used in each iteration. Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public.

Breaking the DES

The Electronic Frontier Foundation (EFF) first built a machine in 1998 to break DES by brute-force attack of the key space. The machine costs less than $250,000 and the attack took less than three days.



The EFF's DES cracker "Deep Crack" custom microchip

Basics

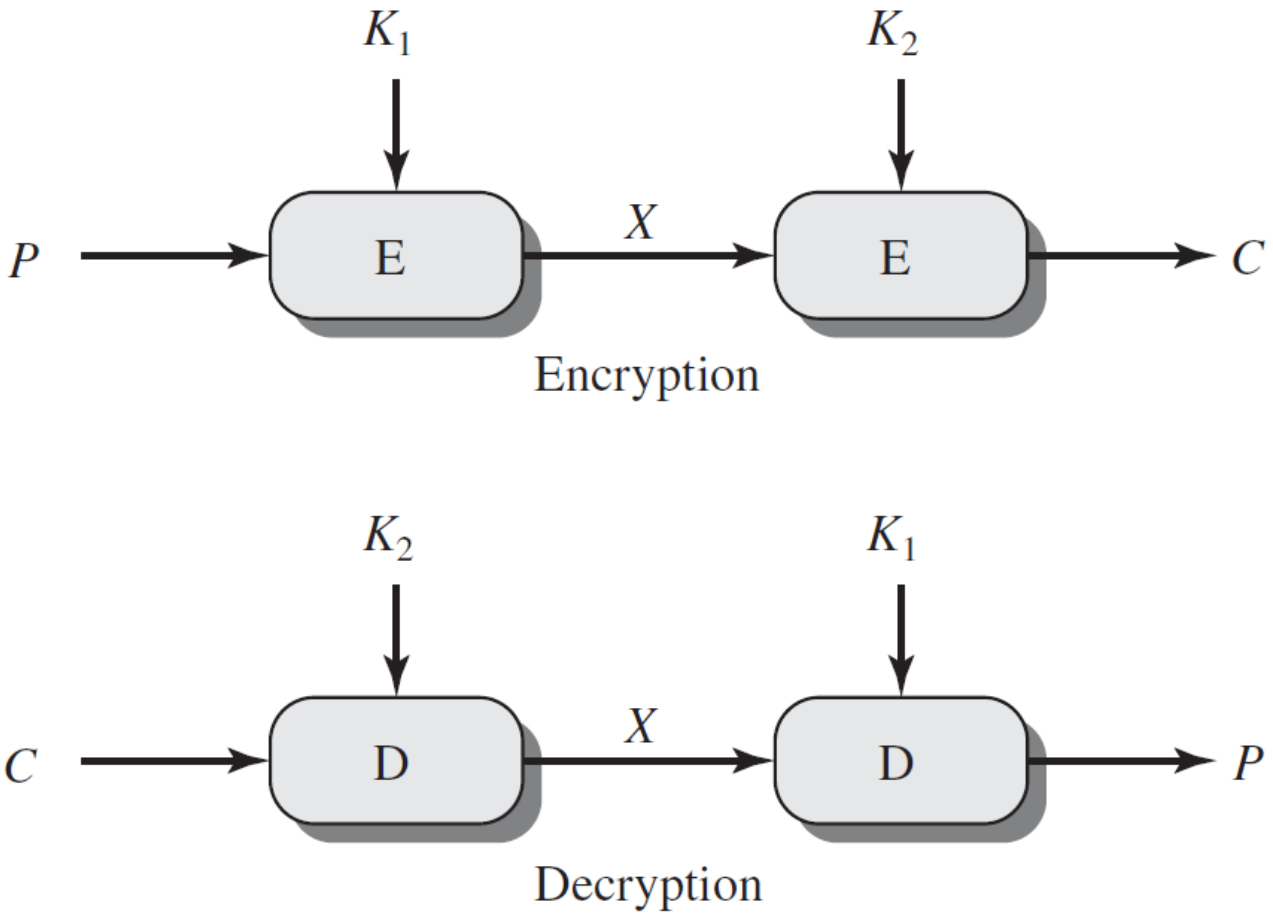
As soon as the potential vulnerability of DES to a brute-force attack become clear, researchers tried to develop more complex forms of DES.

The use of multiple encryption along with multiple keys will increase the strength of DES rapidly.

Multiple encryption and decryption can be formalized as follows:

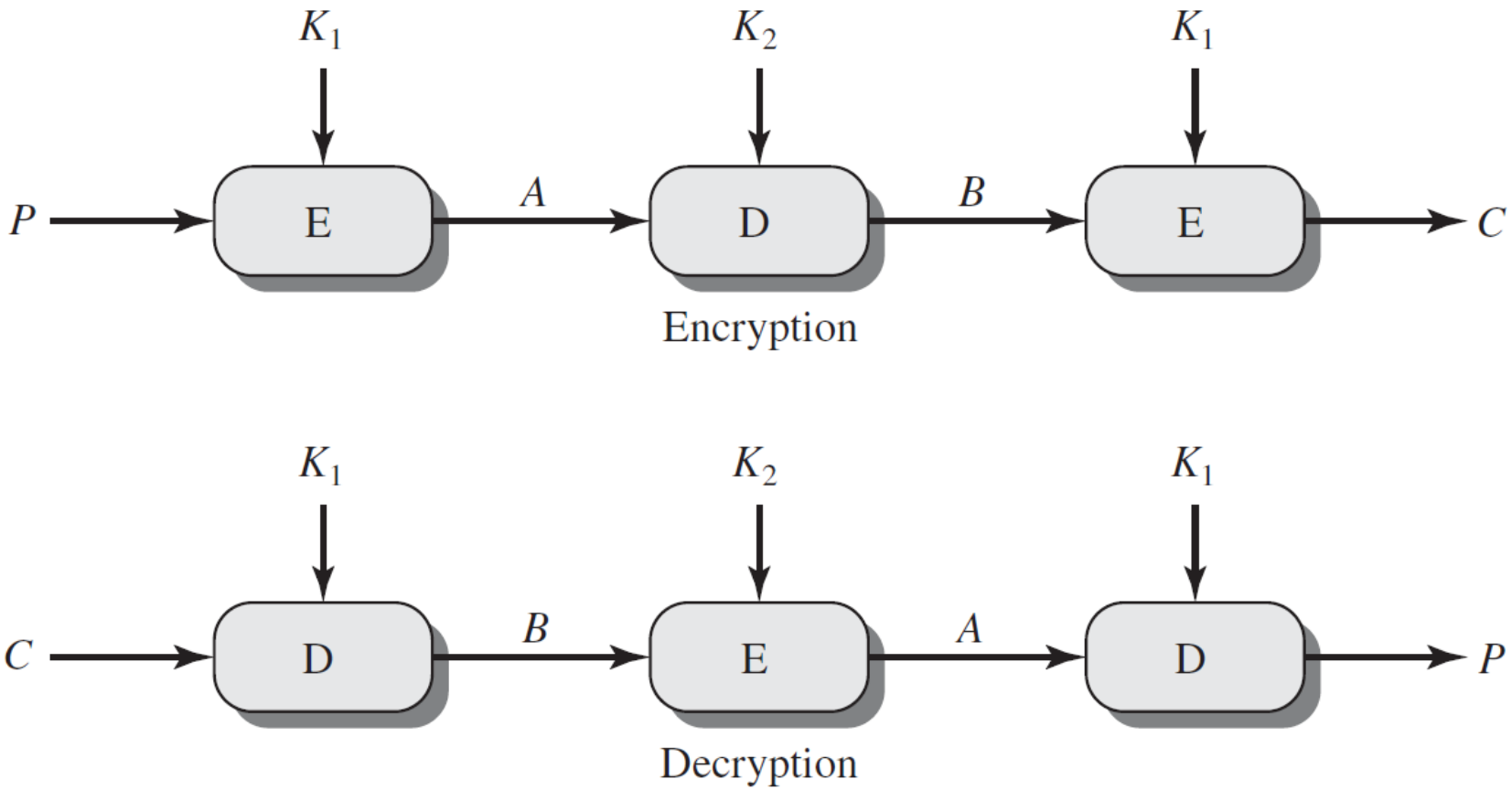
- Encryption: Ciphertext $C = e_n(...e_2(e_1(P, K_1), K_2)..., K_n)$.
- Decryption: Plaintext $P = d_n(...d_n(d_n(C, K_1), K_2)..., K_n)$.

Basics



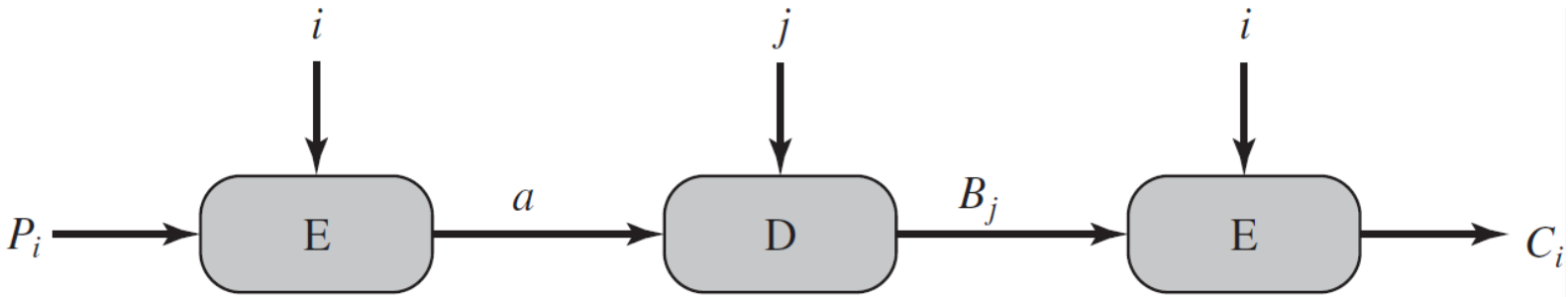
The double encryption and decryption scheme

Basics



The triple encryption and decryption scheme

The working principle of TDES with 2 keys



The TDES 2-key encryption with candidate pair of keys

Disadvantages of TDES

The disadvantages of TDES are given below:

- The processing of TDES is very complex.
- TDES is very slow, especially in software.
- TDES is hard to recover when the file or hard drive crash.