Department of Electronics

Cryptography

Fall 2019

Hasan Mahmood

hasan@qau.edu.pk

Week 5 (02 & 03 October 2019)

Basic Number Theory

- Messages are represented as numbers
- Encryption process: Mathematical operations ->output from input
- Number Theory and Theory of Congruences
- Divisibility, Prime numbers, Greatest Common Divisor
- Solving $ax+by=d$
- Congruences
- Division, Fraction, Exponentiation
- Three Pass Protocol
- Primitive Roots
- Inverting Matrix Mod n
- Square Root Mod n
- Finite Fields

The Data Encryption Standard

- In 1973, National Bureau of Standards (NBS) (NIST) issued a request
- IBM-> LUCIFER->NSA->Data Encryption Standard (DES)
- Released in 1975, became official standard in 1977 (mysterious)
- Block cipher, consists of 64 bits blocks, called Feistel system
- In 1990 Ele Biham and Adi Shamir-> applied differential cryptanalysis
- Key size small, not it is possible to break DES by brute force algorithm
- Replaced in 2000 by NIST by a new standard

A simplified DES-Type Algorithm

- 12-bits message L_0R_0, where L_0 consists of 6 bits and R_0 consists of 6 bits
- The key is 9 bits
- In ith round, $L_{i-1}R_{i-1}$ is transformed to L_iR_i using 8-bit key Ki (from K)
- Function $f(R_{i-1}, K_i)$ that takes 6-bit input and produces 6-bit output
- R to L through XOR

The output for the ith round is defined as follows:

$$L_i = R_{i-1} \text{ and } R_i = L_{i-1} \oplus f(R_{i-1}, K_i),$$

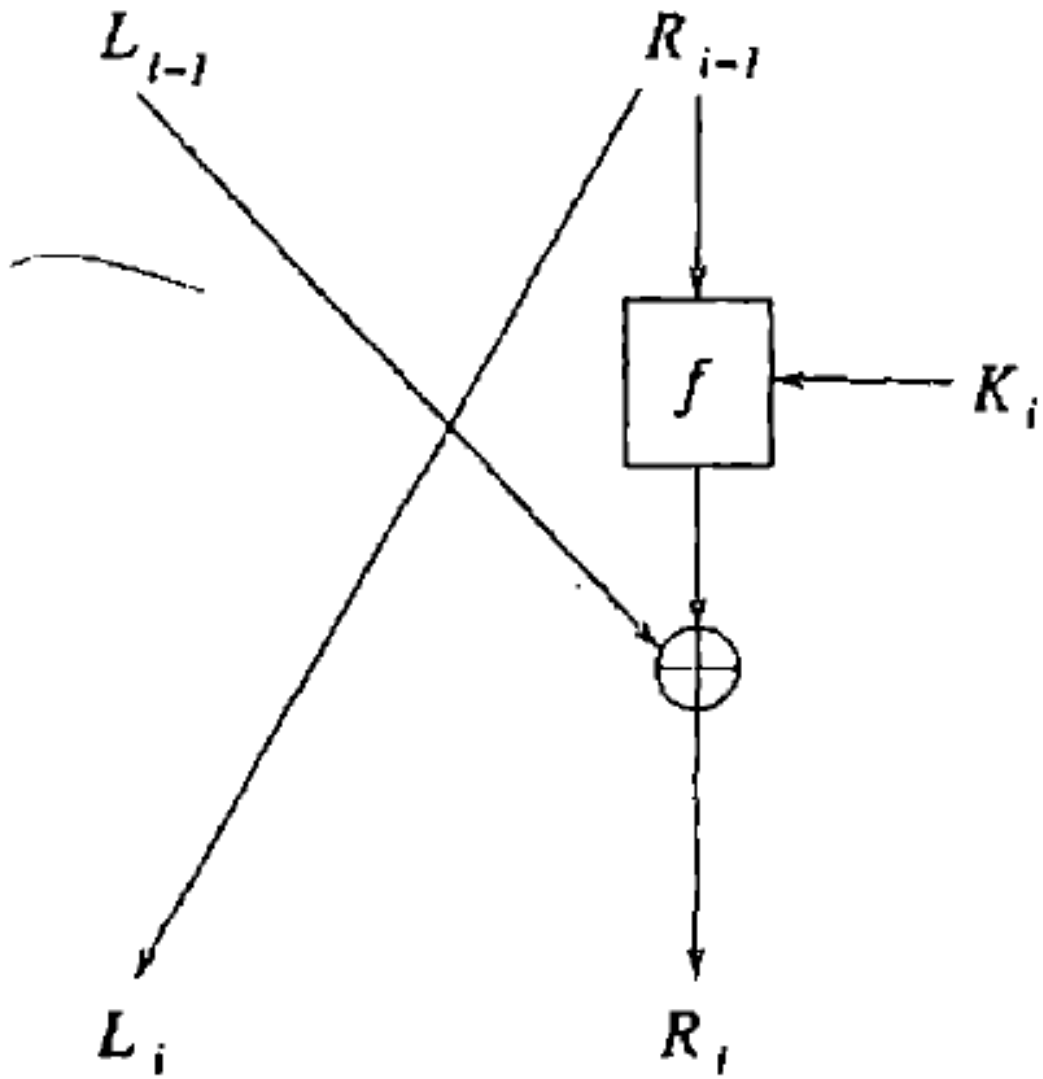


Figure 4.1: One Round of a Feistel System.

Decryption

- Performed for certain number of rounds, say *n,* produce *LnRn*
- Start with LnRn
- Use the same procedure as before, but with the keys Ki used in reverse order

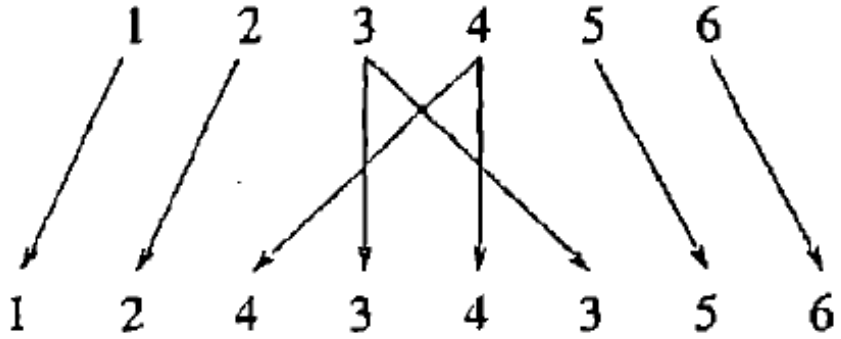
$$[L_n] \quad [R_n \oplus f(L_n, K_n)].$$

Decryption

$$[L_n] \quad [R_n \oplus f(L_n, K_n)] \quad = [R_{n-1}] \quad [L_{n-1} \oplus f(R_{n-1}, K_n) \oplus f(L_n, K_n)]$$
$$= [R_{n-1}] \quad [L_{n-1}].$$

Function *f*

- Any f would work in the above procedures
- Some choices of f yield much better security than others
- Expander function, input 6-bits, output 8-bits
- 011001 is expanded to 01010101



S-box

- The main components are called S-boxes
- Two are used. The input is 4-bits (1st bit is row, other 3 bits represents the column)

$$S_1 \quad \begin{bmatrix} 101 & 010 & 001 & 110 & 011 & 100 & 111 & 000 \\ 001 & 100 & 110 & 010 & 000 & 111 & 101 & 011 \end{bmatrix}$$

$$S_2 \quad \begin{bmatrix} 100 & 000 & 110 & 101 & 111 & 001 & 011 & 010 \\ 101 & 011 & 000 & 111 & 110 & 010 & 001 & 100 \end{bmatrix}.$$

- Input 1010, output 110

Key

- The key is 9 bits
- The key *Ki* for the *i*th round of encryption is obtained by using 8 bits of K starting with the *i*th bit
- If *K*=010011001, then K_5=01100101
- The key is selected by the user
- It is used to determine the location of substitution in a S-box

$f(R_{i-1}, K_i)$

- The input R_{i-1} consists of 6-bits
- The expander function is used to expand it to 8-bits
- The result is XORed with K_i to produce another 8-bit number
- The first 4 bits are sent to S_1, and last 4 bits are sent to S_2
- Each S-box outputs 3-bits, which are concatenated to form a 6-bit number to get $f(R_{i-1}, K_i)$

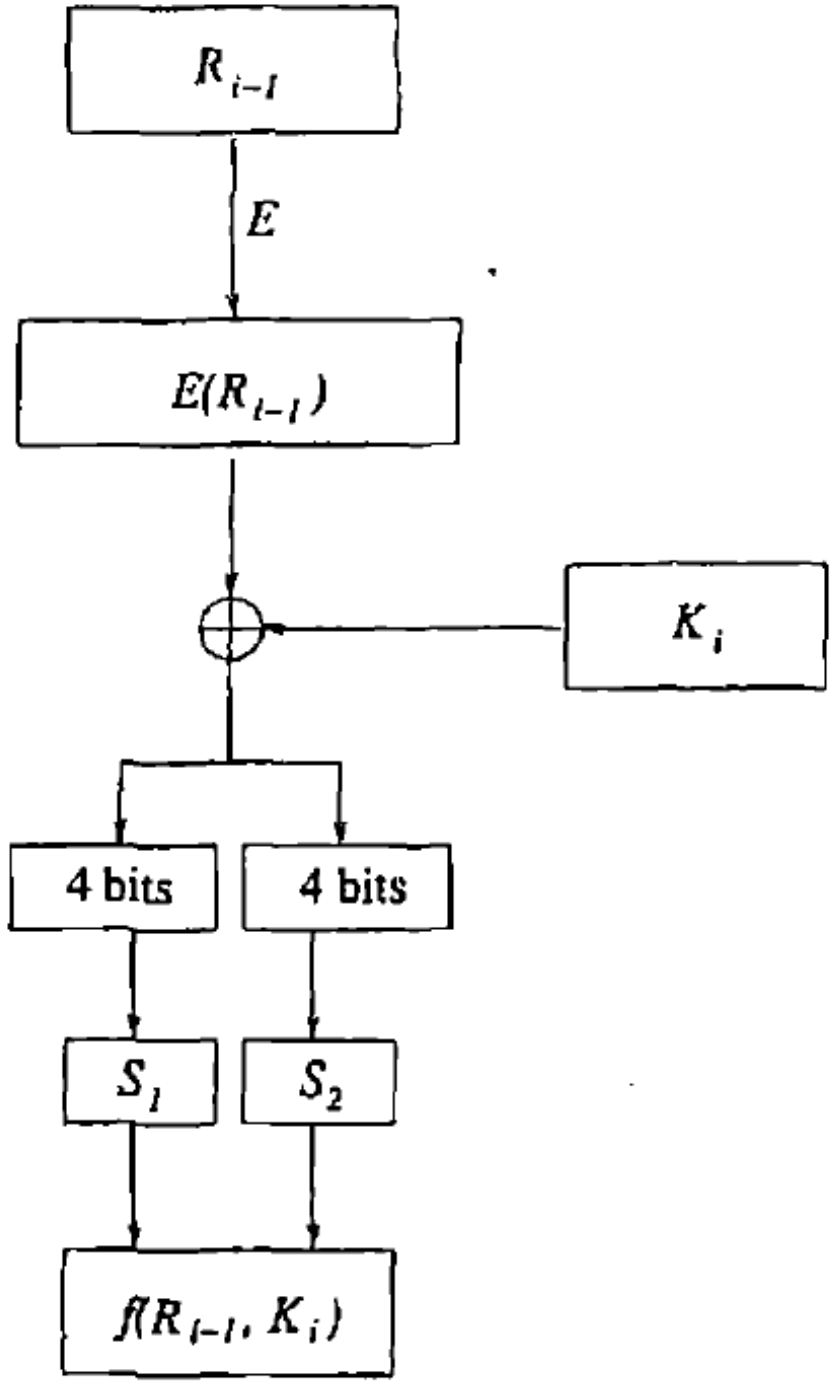
For example, suppose $R_{i-1} = 100110$ and $K_i = 01100101$. We have

$$E(100110) \oplus K_i = 10101010 \oplus 01100101 = 11001111.$$

Substitution

- The first four bits are sent to S_1 and the last 4 bits are sent to S_2
- $F(R_{i-1}, K_i)$=000100 (after lookup in the S-boxes)
- Suppose the input is $L_{i-1}R_{i-1}$=011100100110
- K_i=01100101
- R_{i-1}=100110
- Therefore, $f(R_{i-1}, K_i)$=000100
- This is XORed with L_{i-1}=011100 to yield R_i=011000
- Since $L_i=R_{i-1}$, we obtain L_iR_i=100110011000

DES Round Simplified



DES

- Block cipher, consists of 64 bits
- The key has 56 bits, expressed as 64 bit string
- 8^{th}, 16^{th}, 24^{th}, …. Bits are parity bits (odd parity), only error detection
- Output is 64 bit ciphertext
- The algorithm consists of 3 stages.
- The algorithm begins with the plaintext *m* of 64 bits

DES three stages

1. The bits of m are permuted by a fixed initial permutation to obtain $m_0 = IP(m)$. Write $m_0 = L_0 R_0$, where L_0 is the first 32 bits of m_0 and R_0 is the last 32 bits.

DES three stages
Stage 2

2. For $1 \leq i \leq 16$, perform the following:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i), \end{aligned}$$

where K_i is a string of 48 bits obtained from the key K and f is a function to be described later.

DES three stages
Stage 3

3. Switch left and right to obtain $R_{16}L_{16}$, then apply the inverse of the initial permutation to get the ciphertext $c = IP^{-1}(R_{16}L_{16})$.

DES decription

- Performed by exactly the same procedure
- Except the keys $K_1, \ldots, K_{16}$ are used in reverse order

DES Details

- Initial permutation
- For example, the 1st bit becomes the 58th bit

Initial Permutation

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

Function $f(R, K_i)$

1. First, R is expanded to $E(R)$ by the following table.

Expansion Permutation											
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

Function $f(R, K_i)$ computation

2. Compute $E(R) \oplus K_i$, which has 48 bits, and write it as $B_1 B_2 \cdots B_8$, where each B_j has 6 bits.

Mapping to S-boxes Step 3

- There are 8 S-boxes $S_1, \ldots, S_8$
- B_j is the input for S_j
- Row b_1, b_6
- Column $b_2b_3b_4b_5$
- B_3=001001
- Output 0100

4. The string $C_1 C_2 \cdots C_8$ is permuted according to the following table.

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

The resulting 32-bit string is $f(R, K_j)$.

Key processing

Finally, we describe how to obtain $K_1, \ldots K_{16}$. Recall that we start with a 64-bit K.

1. The parity bits are discarded and the remaining bits are permuted by the following table.

Key Permutation													
57	49	41	33	25	17	9	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	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Write the result as C_0D_0, where C_0 and D_0 have 28 bits.

Key processing, step 2, key bits shifting

2. For $1 \leq i \leq 16$, let $C_i = LS_i(C_{i-1})$ and $D_i = LS_i(D_{i-1})$. Here LS_i means shift the input one or two places to the left, according to the following table.

Number of Key Bits Shifted per Round																
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Key processing, step 3, key selection

3. 48 bits are chosen from the 56-bit string C_iD_i according to the following table. The output is K_i.

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

It turns out that each bit of the key is used in approximately 14 of the 16 rounds.

Remarks

- In a good cipher system, each bit of the cyphertext should depend on all bits of the plaintext
- The expansion E(R) is designed so that this will happen in only few rounds
- The purpose of the initial purpose is not clear
- S-boxes are the heart of the algorithm and provide security
- Design is a mystery….

S-box criteria

1. Each S-box has 6 input bits and 4 output bits. This was the largest that could be put on one chip in 1974.

2. The outputs of the S-boxes should not be close to being linear functions of the inputs (linearity would have made the system much easier to analyze).

3. Each row of an S-box contains all numbers from 0 to 15.

4. If two inputs to an S-box differ by 1 bit, the outputs must differ by 2 bits.

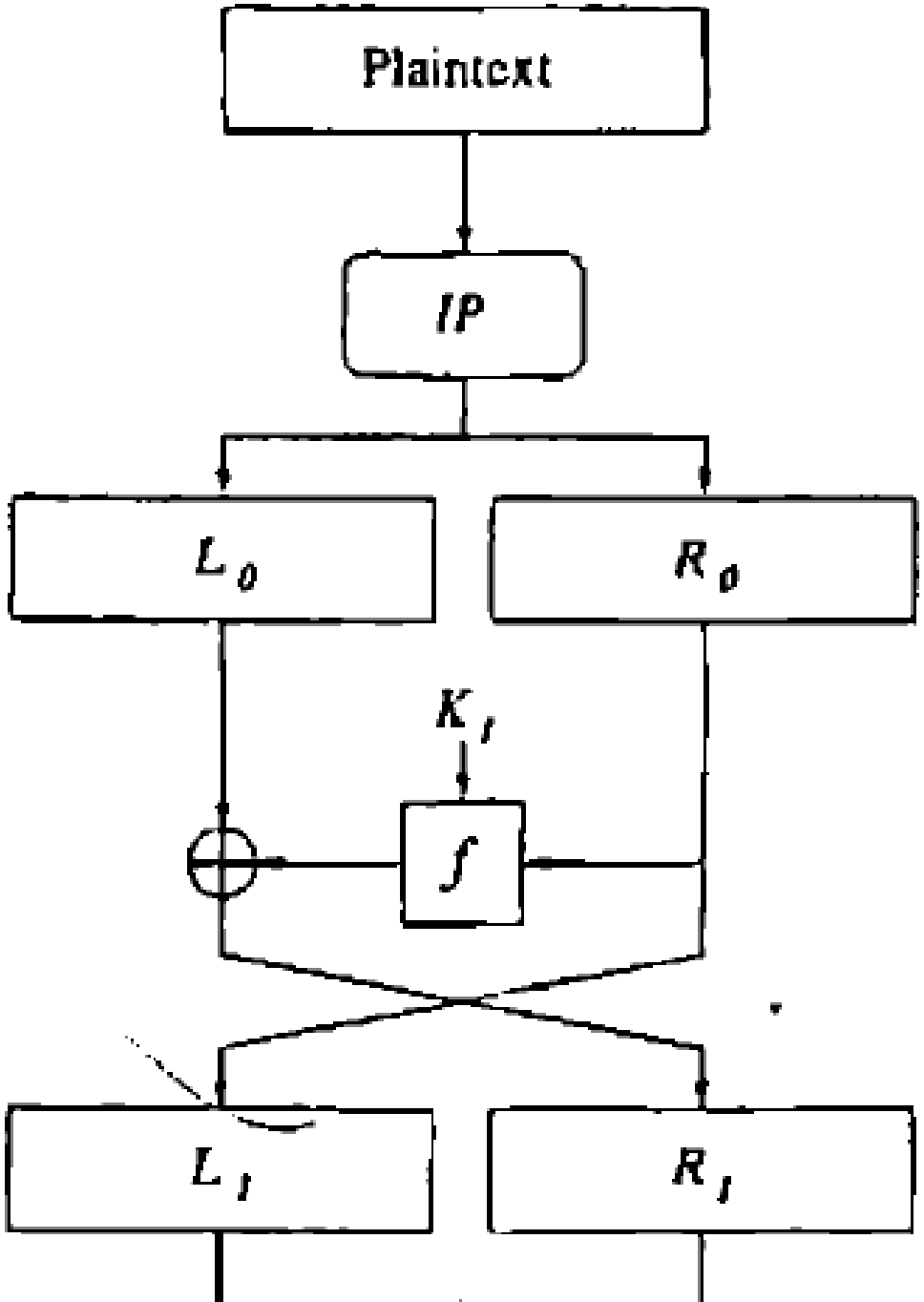
5. If two inputs to an S-box differ in their first 2 bits but have the same last 2 bits, the outputs must be unequal.

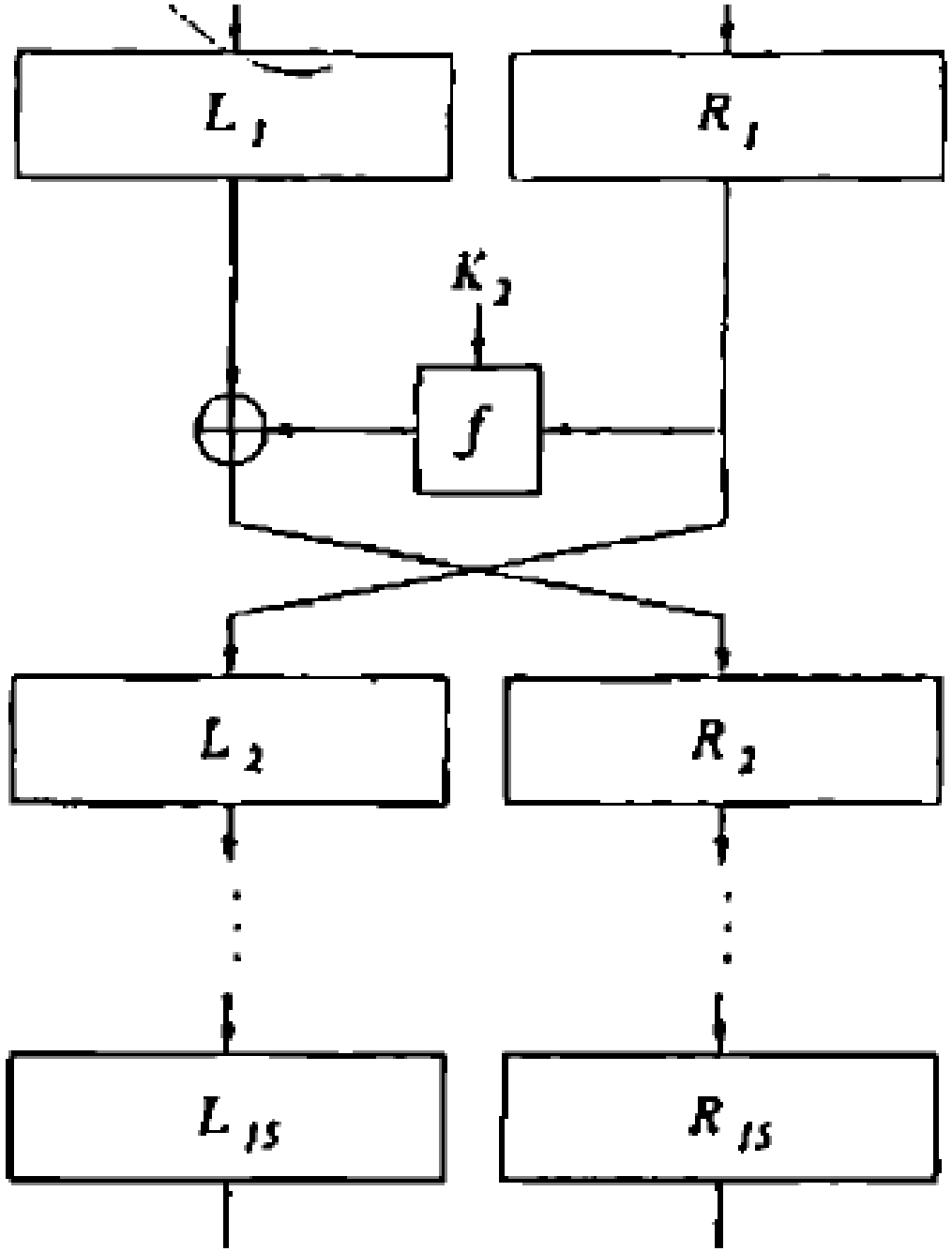
S-box criteria, continued

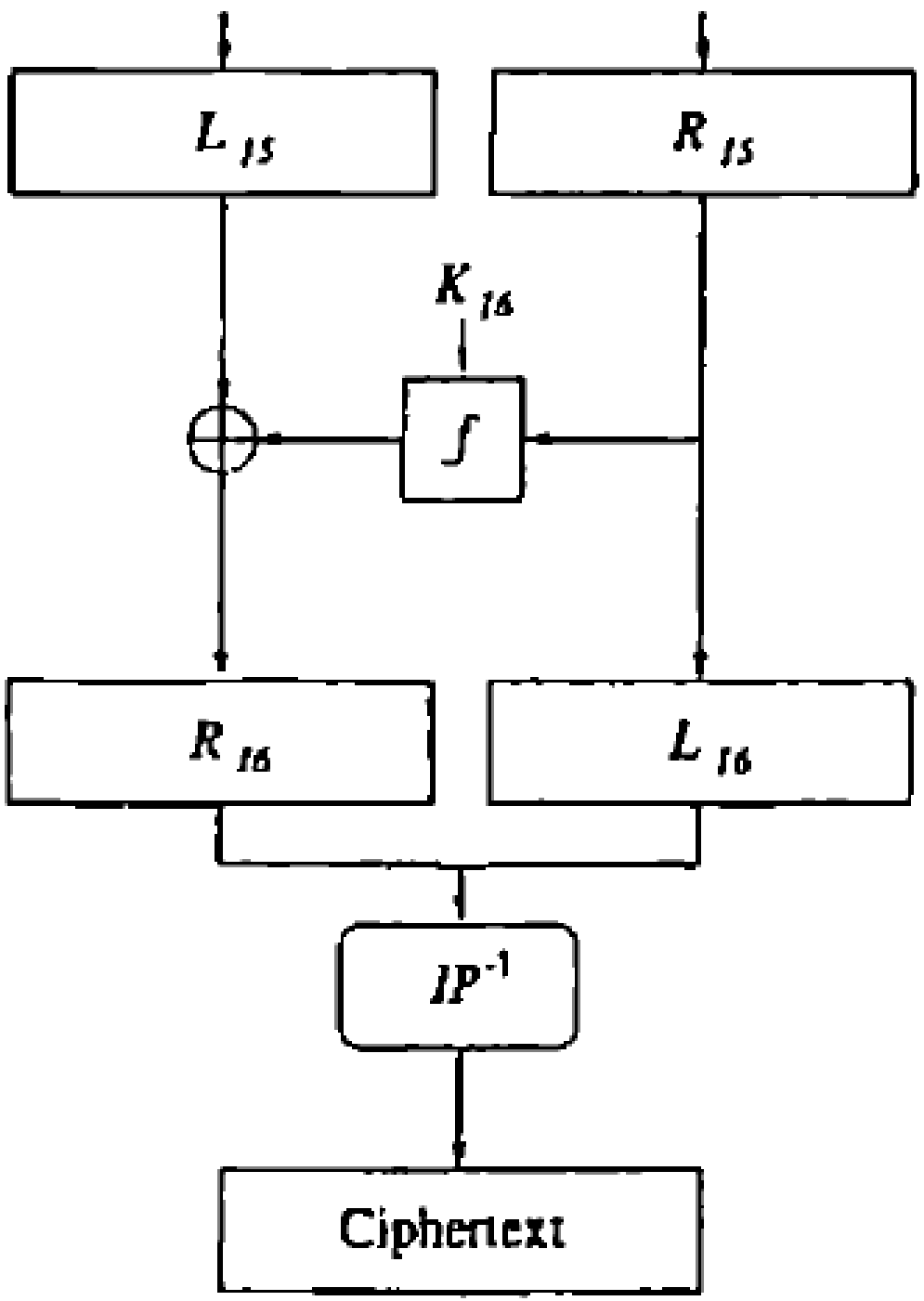
6. There are 32 pairs of inputs having a given XOR. For each of these pairs, compute the XOR of the outputs. No more than eight of these output XORs should be the same. This is clearly to avoid an attack via differential cryptanalysis.

7. A criterion similar to (6), but involving three S-boxes.

DES algorithm




L_1
R_1
K_2
f
L_2
R_2
L_{15}
R_{15}


L_{15}
R_{15}
K_{16}
f
R_{16}
L_{16}
IP^{-1}
Ciphertext

Modes of operation

- DES is a block cipher, 64-bits blocks, longer or shorter messages
- Character by character transmission (messages shorter than 64-bits)
- Many modes of operation, allowing users to choose appropriate modes to meet the requirements of their applications

1. Electronics codebook (ECB)
2. Cipher block chaining (CBC)
3. Cipher feedback (CFB)
4. Output feedback (OFB)
5. Counter (CTR)

Electronics Codebook (RCB)

- Break plaintext into appropriate sized blocks, and process separately
- Encryption function E_k is used
- This is know as the electronics codebook (ECB) mode of operation
- Plaintext: $P=[P_1, P_2, P_3, \ldots, P_L]$
- Cyphertext : $C=[C_1, C_2, \ldots, C_L]$
- Where $C_j=E_K(P_j)$ *is the encryption of* P_j *using key K*
- Apparent weakness when plaintext is long

ECB weakness

- Eve has been observing communication between Alice and Bob for long enough period of time
- If Eve has managed to acquire some plaintext pieces corresponding to the ciphertext pieces (that was observed)
- Eve can start to build up a codebook with which Eve can decipher future communication between Alice an Bob
- Eve never needs to calculate the Key
- Codebook is used to decipher the communication
- Real problem if the fragments are repeated in the plaintext
- Email header example, it repeats on specific dates
- False ciphertext message corrupt the original message

Date: Tue, 29 Feb 2000 13:44:38 -0500 (EST)

Cipher Block Chaining (CBC)

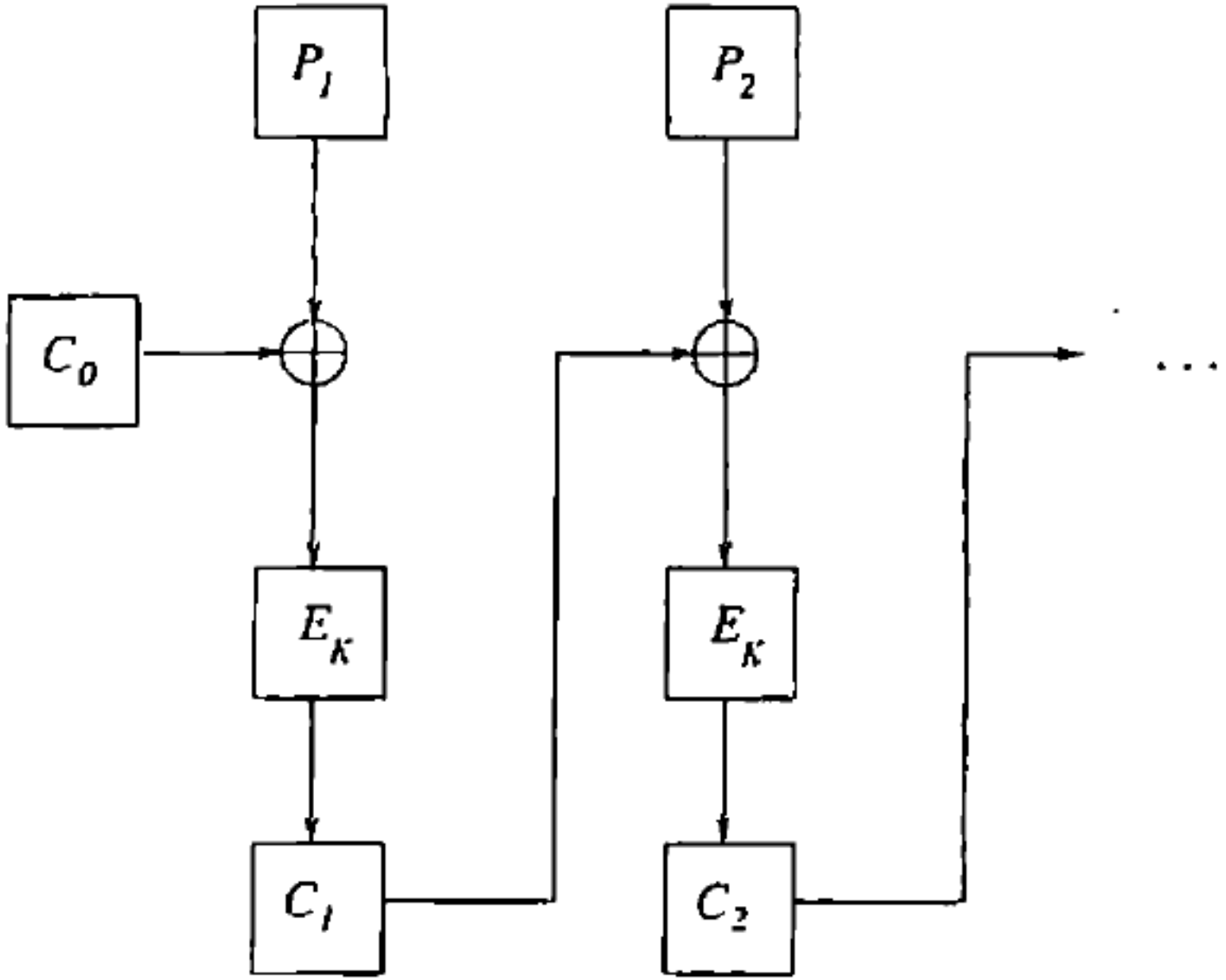
- Reduce problem in ECB mode is to use chaining
- Chaining is a feedback mechanism where the encryption of block depends on the encryption of the previous blocks
- In general, encryption proceeds as

$$C_j = E_K(P_j \oplus C_{j-1}),$$

- With decryption as
-

$$P_j = D_K(C_j) \oplus C_{j-1}$$

CBC



Cipher Feedback (CFB)

- CBC and ECB work when complete block of 64-bit of plaintext is available
- Based on Linear Feedback Shift Register (LFSR)
- Cipher feedback mode is a stream mode of operation that produces pseudorandom bits using the block cipher E_K
- In general, it operates in a *k*-bit mode, where each application produces *k* random bits XORing with the plaintext (8-bit version)
- Useful for interactive computer applications
- Plaintext is broken into 8-bit pieces $P=[P_1, P_2, \ldots]$

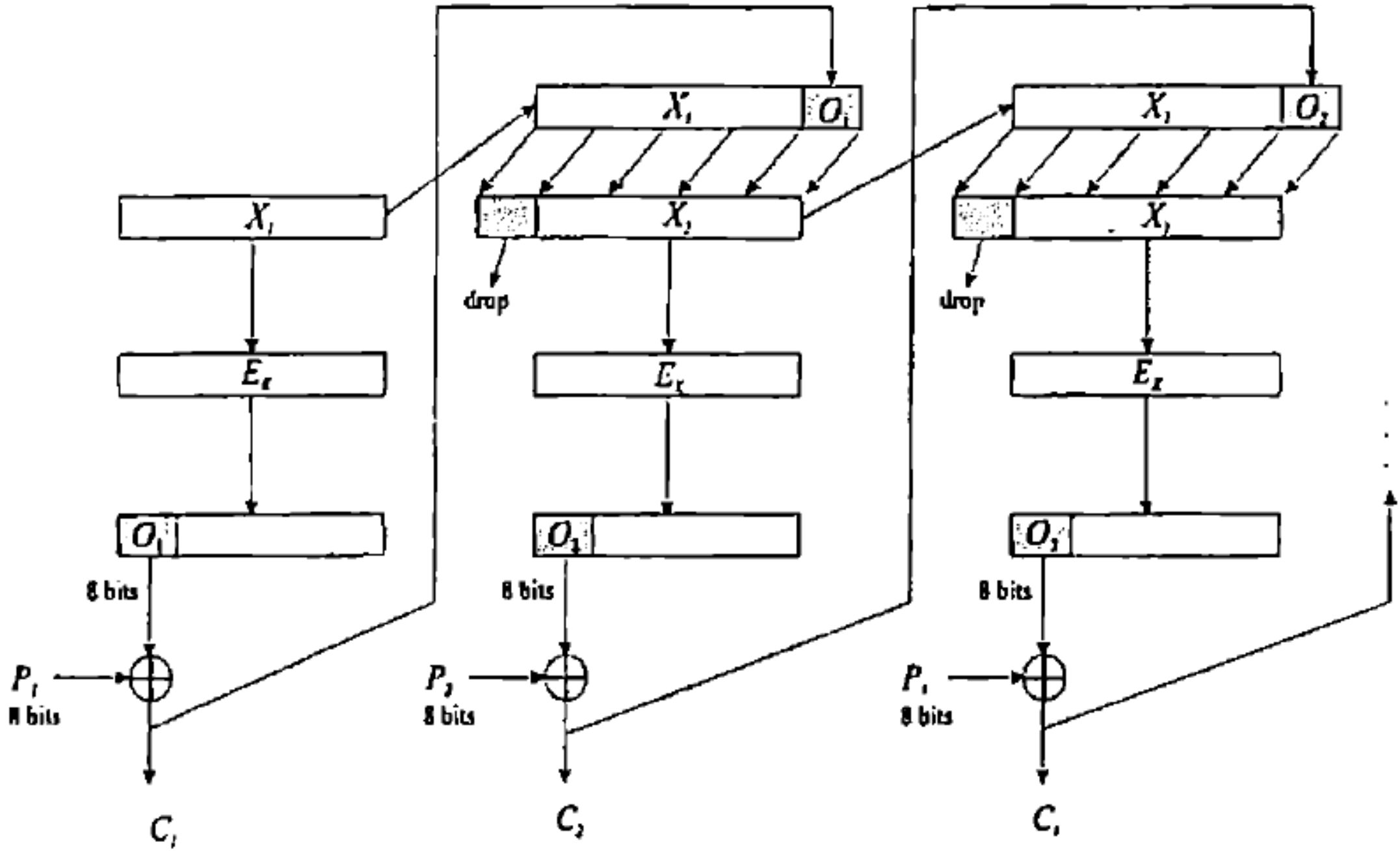
Cipher Feedback (CFB) Encryption

- An initial 64-bit X_1 is chosen, then for j=1,2,3,…. the following is performed:

$$\begin{aligned} O_j &= L_8(E_K(X_j)) \\ C_j &= P_j \oplus O_j \\ X_{j+1} &= R_{56}(X_j) \| C_j, \end{aligned}$$

- $L_8(X)$ denotes the 8 leftmost bits of X
- $R_{56}(X)$ denotes the rightmost 56 bits of X
- $X||Y$ denotes the string obtained by wiring X followed by Y

CFM



CFB decryption

Decryption is done with the following steps:

$$P_j = C_j \oplus L_8(E_K(X_j))$$

$$X_{j+1} = R_{56}(X_j) \parallel C_j.$$