Course: Cryptography and Network Security
Code: CS-34310
Branch: M.C.A - 4th Semester

Lecture – 15 : Modern Symmetric-Key Ciphers

Faculty & Coordinator : Dr. J Sathish Kumar (JSK)

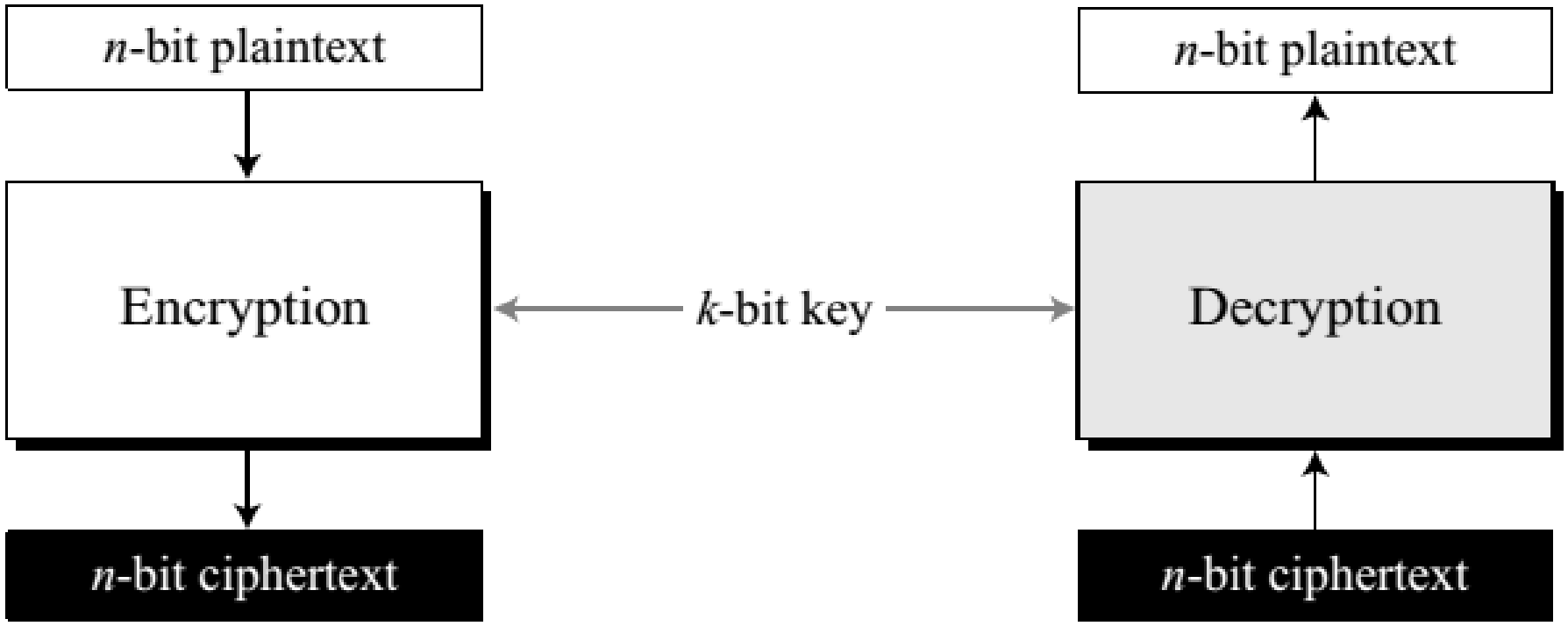
Department of Computer Science and Engineering

Motilal Nehru National Institute of Technology Allahabad,
Prayagraj-211004

Introduction

- The traditional symmetric-key ciphers that we have studied so far are character-oriented ciphers.
- With the advent of the computer, we need bit-oriented ciphers.
- This is because the information to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data.
- MODERN BLOCK CIPHERS
- MODERN STREAM CIPHERS

A modern block cipher



How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

- Encoding 100 characters using 8-bit ASCII results in an 800-bit message.
- The plaintext must be divisible by 64.
- If |M| and |Pad| are the length of the message and the length of the padding,

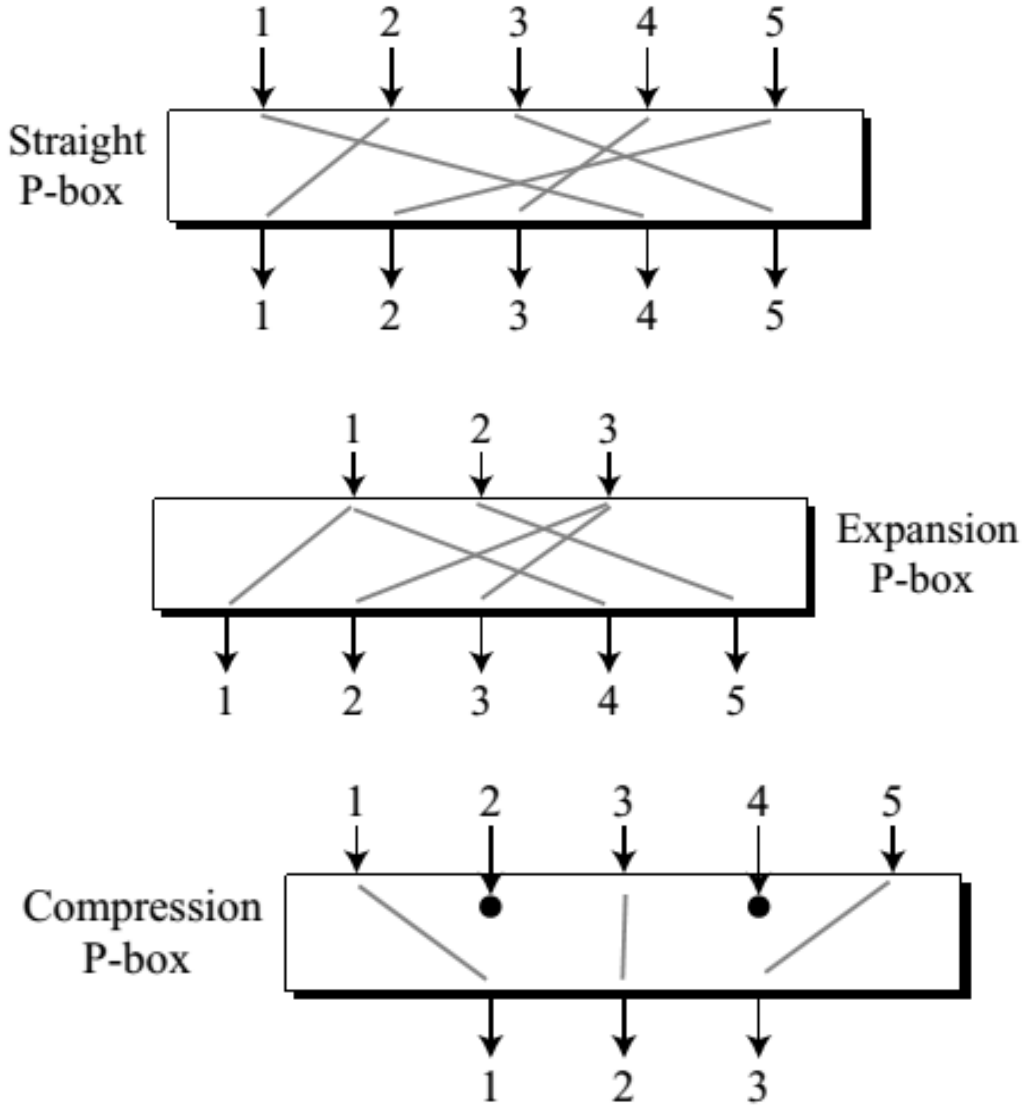
  $|M| + |Pad| = 0 \bmod 64 \rightarrow |Pad| = -800 \bmod 64 \rightarrow 32 \bmod 64$
- This means that 32 bits of padding (for example, 0's) need to be added to the message.
- The plaintext then consists of 832 bits or thirteen 64-bit blocks.
- Note that only the last block contains padding.
- The cipher uses the encryption algorithm thirteen times to create thirteen ciphertext blocks.

Components of a Modern Block Cipher

- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.
- However, modern block ciphers normally are not designed as a single unit.
- To provide the required properties of a modern block cipher, such as diffusion and confusion, a modern block cipher is made of a combination of transposition units (called P-boxes), substitution units (called S-boxes), and some other units.

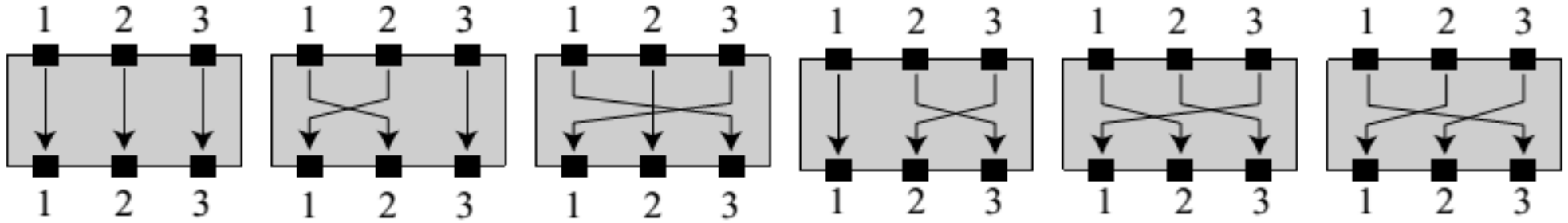
P-Boxes

- A P-box (permutation box) parallels the traditional transposition cipher for characters.
- It transposes bits.
- We can find three types of P-boxes in modern block ciphers:
  - Straight P-boxes,
    - Ex: (5 × 5 straight P-box)
  - Expansion P-boxes,
    - Ex: (3 × 5 expansion P-box) and
  - Compression P-boxes
    - (5 × 3 compression P-box)



Straight P-Boxes

- A straight P-Box with n inputs and n outputs is a permutation. There are n! possible mappings.



All 6 possible mappings of a 3 × 3 P-box.

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

An example of a straight permutation table when n is 64.

Design an 8 × 8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

- We need a straight P-box with the table [4 1 2 3 6 7 8 5]. The relative positions of input bits 1, 2, 3, 6, 7, and 8 have not been changed, but the first output takes the fourth input and the eighth output takes the fifth input.

Compression P-Boxes

- A compression P-box is a P-box with n inputs and m outputs where m < n.
- Some of the inputs are blocked and do not reach the output
- We need to know that a permutation table for a compression P-box has m entries, but the content of each entry is from 1 to n with some missing values (those inputs that are blocked).
- An example of a permutation table for a 32 × 24 compression P-box.
- Note that inputs 7, 8, 9, 15, 16, 23, 24, and 25 are blocked.

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

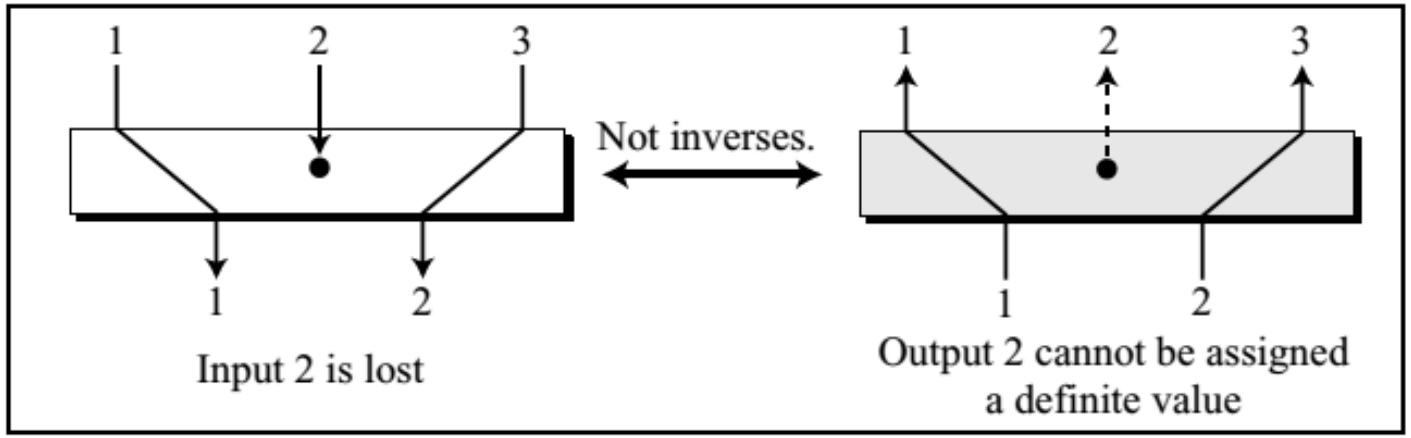
Expansion P-Boxes

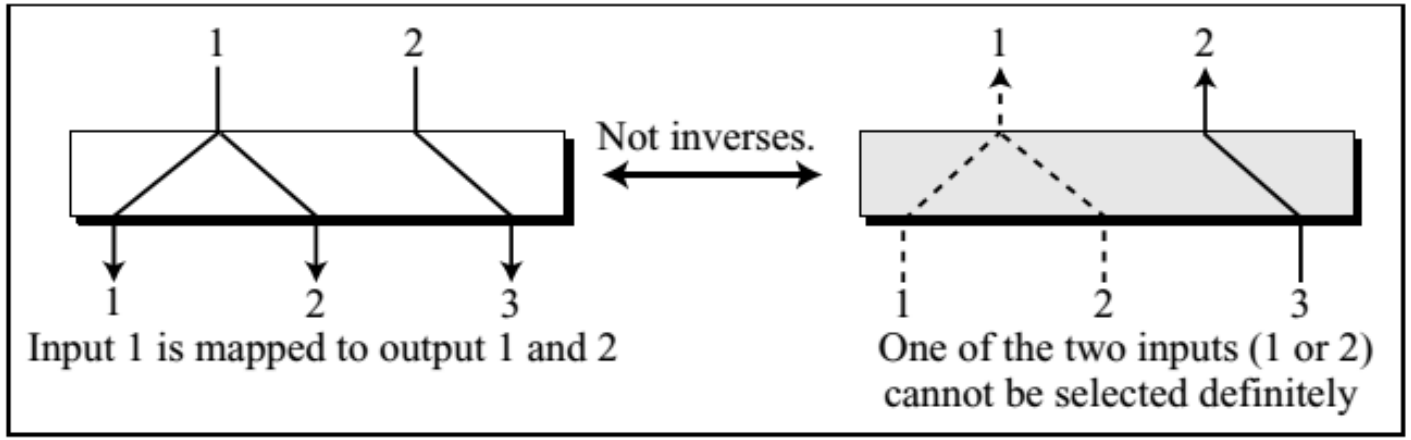
- An expansion P-box is a P-box with n inputs and m outputs where m > n.
- Some of the inputs are connected to more than one input
- The expansion P-boxes used in modern block ciphers normally are keyless, where a permutation table shows the rule for transposing bits.
- We need to know that a permutation table for an expansion P-box has m entries, but m – n of the entries are repeated (those inputs mapped to more than one output).
- An example of a permutation table for a 12 × 16 expansion P-box.
- Note that each of the inputs 1, 3, 9, and 12 is mapped to two outputs.

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12

Invertibility

Compression P-box

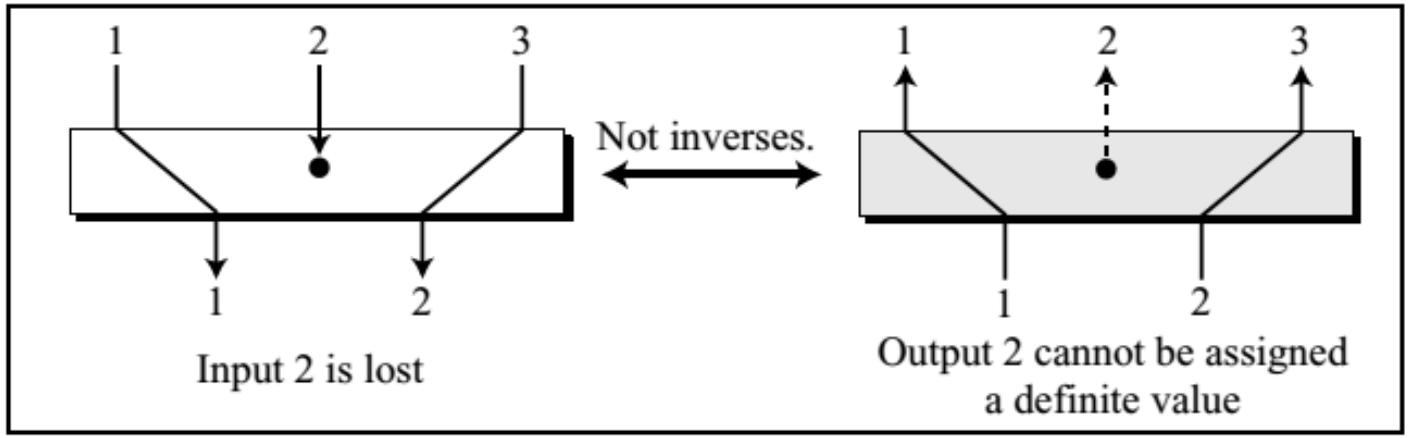


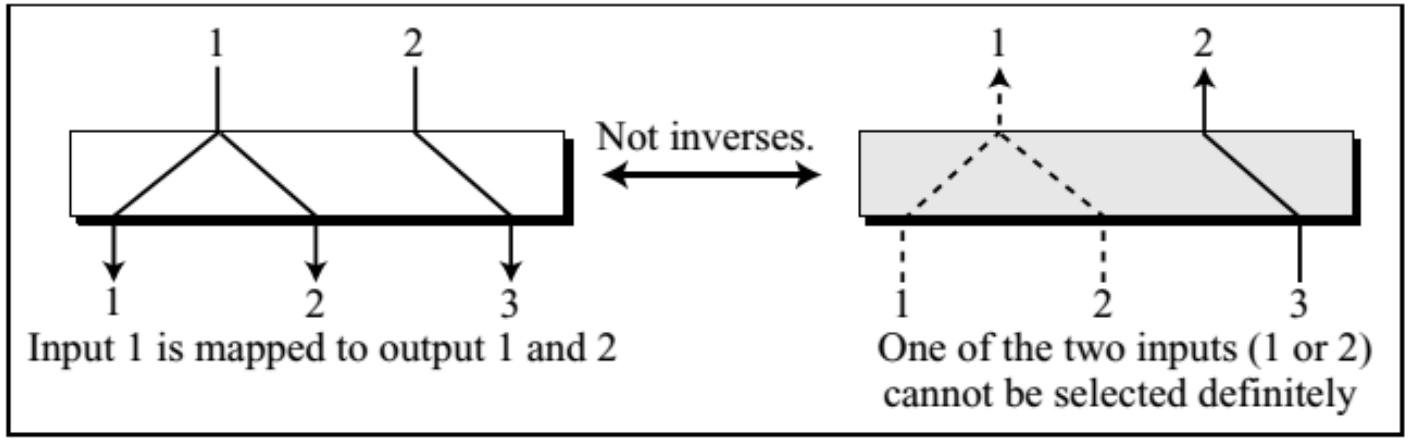


Expansion P-box

Invertibility

Compression P-box





Expansion P-box

A straight P-box is invertible, but compression and expansion P-boxes are not.

S-Boxes

- An S-box (substitution box) can be thought of as a miniature substitution cipher.
- However, an S-box can have a different number of inputs and outputs.
- In other words, the input to an S-box could be an n-bit word, but the output can be an m-bit word, where m and n are not necessarily the same.

Linear Versus Nonlinear S-Boxes

- In an S-box with n inputs and m outputs, we call the inputs x_0, x_1, …, x_n and the outputs y_1, …, y_m.
- The relationship between the inputs and the outputs can be represented as a set of equations

$$
\begin{aligned}
y_1 &= f_1(x_1, x_2, \ldots, x_n) \\
y_2 &= f_2(x_1, x_2, \ldots, x_n) \\
&\ldots \\
y_m &= f_m(x_1, x_2, \ldots, x_n)
\end{aligned}
$$

Linear Versus Nonlinear S-Boxes

- In a linear S-box, the above relations can be expressed as

$$
\begin{aligned}
y_1 &= a_{1,1}x_1 \oplus a_{1,2}x_1 \oplus \cdots \oplus a_{1,n}x_n \\
y_2 &= a_{2,1}x_1 \oplus a_{2,2}x_1 \oplus \cdots \oplus a_{2,n}x_n \\
&\cdots \\
y_m &= a_{m,1}x_1 \oplus a_{m,2}x_1 \oplus \cdots \oplus a_{m,n}x_n
\end{aligned}
$$

- In a nonlinear S-box we cannot have the above relations for every output.

Linear Versus Nonlinear S-Boxes

- In an S-box with three inputs and two outputs, we have $y_1 = x_1 \oplus x_2 \oplus x_3$ $y_2 = x_1$. Is it linear or not?
- The S-box is linear because $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$ and $a_{2,2} = a_{2,3} = 0$.
- The relationship can be represented by matrices, as shown below

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Linear Versus Nonlinear S-Boxes

- In an S-box with three inputs and two outputs, we have

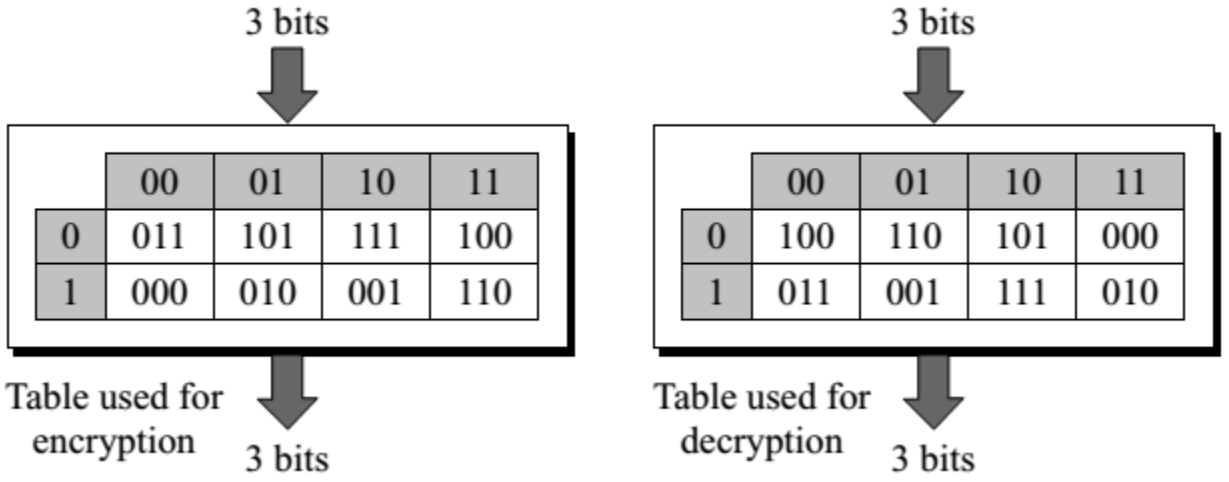
  $y_1 = (x_1)^3 + x_2$, $y_2 = (x_1)^2 + x_1x_2 + x_3$

  where multiplication and addition is in GF(2). The S-box is nonlinear because there is no linear relationship between the inputs and the outputs.

Invertibility

- S-boxes are substitution ciphers in which the relationship between input and output is defined by a table or mathematical relation.
- An S-box may or may not be invertible.
- In an invertible S-box, the number of input bits should be the same as the number of output bits.

For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.



Exclusive-Or

- The five properties of the exclusive-or operation in the GF(2^n) field makes this operation a very interesting component for use in a block cipher.
- Closure: This property guarantees that the result of exclusive-oring two n-bit words is another n-bit word.
- Associativity: This property allows us to use more than one exclusive-or operator in any order. { $x \oplus (y \oplus z) \leftrightarrow (x \oplus y) \oplus z$ }
- Commutativity: This property allows us to swap the inputs without affecting the output. { $x \oplus y \leftrightarrow y \oplus x$ }
- Existence of identity: The identity element for the exclusive-or operation is an n-bit word that consists of all 0's, or (00… 0). This implies that exclusive-oring of a word with the identity element does not change that word. { $x \oplus (00...0) = x$ }
- Existence of inverse: In the GF(2^n) field, each word is the additive inverse of itself. This implies that exclusive-oring of a word with itself yields the identity element. {$x \oplus x = (00…0)$}

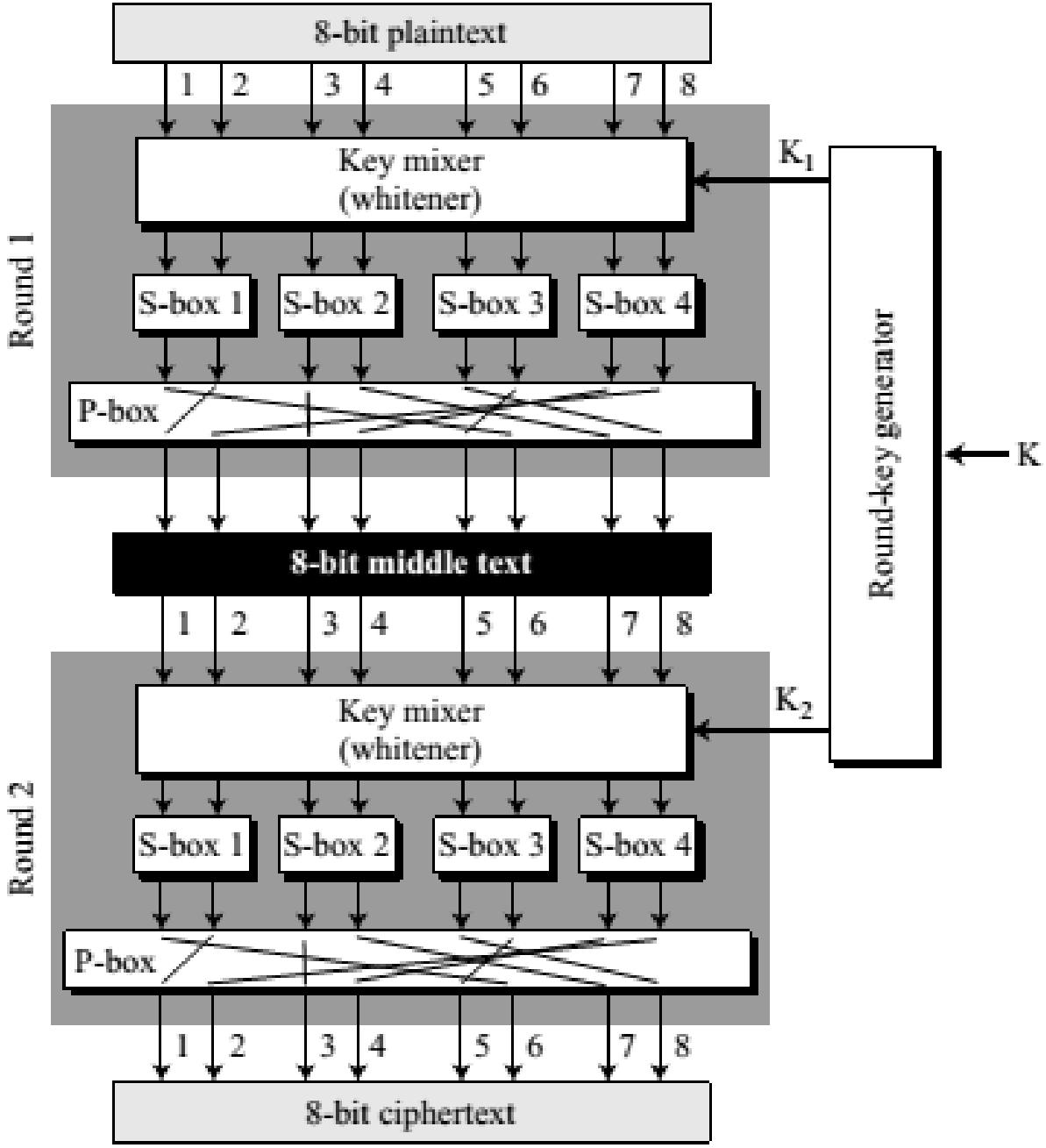
Diffusion and Confusion

- The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.
- This will frustrate the adversary who uses ciphertext statistics to find the plaintext.
- Diffusion implies that each symbol (character or bit) in the ciphertext is dependent on some or all symbols in the plaintext.
- In other words, if a single symbol in the plaintext is changed, several or all symbols in the ciphertext will also be changed.

- The idea of confusion is to hide the relationship between the ciphertext and the key.
- This will frustrate the adversary who tries to use the ciphertext to find the key.
- In other words, if a single bit in the key is changed, most or all bits in the ciphertext will also be changed.

Rounds

- Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.
- Each iteration is referred to as a round.
- In an N-round cipher, the plaintext is encrypted N times to create the ciphertext.
- The ciphertext is decrypted N times to create the plaintext.
- We refer to the text created at the intermediate levels (between two rounds) as the middle text.



Rounds

A simple product cipher with two rounds.

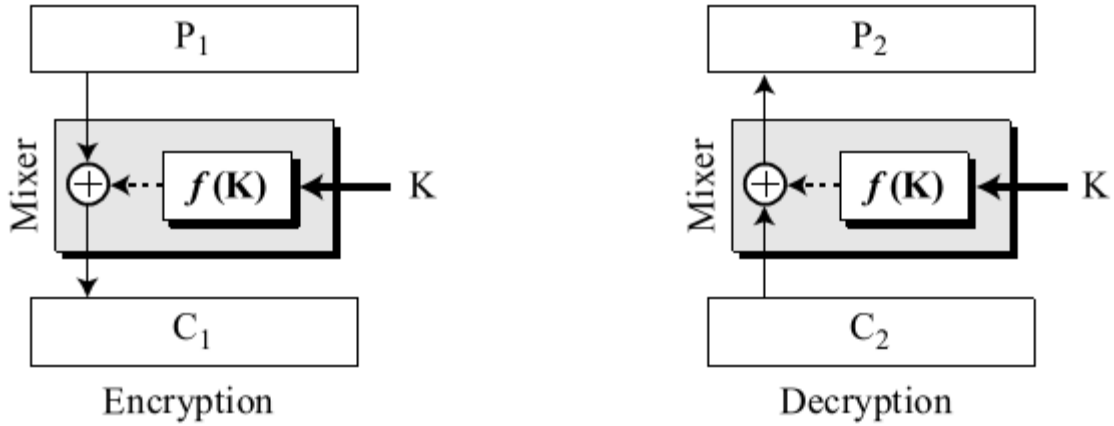
The 8-bit text is mixed with the key to whiten the text (hide the bits using the key).
This is normally done by exclusive-oring the 8-bit word with the 8-bit key.

Two Classes of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
- The ciphers in the first class use both invertible and noninvertible components.
  - The ciphers in this class are normally referred to as Feistel ciphers.
  - The block cipher DES is a good example of a Feistel cipher.
- The ciphers in the second class use only invertible components.
  - We refer to ciphers in this class as non-Feistel ciphers (for the lack of another name).
  - The block cipher AES is a good example of a non-Feistel cipher.

Feistel Ciphers

- A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible.
- A Feistel cipher combines all noninvertible elements in a unit and uses the same unit in the encryption and decryption algorithms.
- The effects of a noninvertible component in the encryption algorithm can be canceled in the decryption algorithm if we use an exclusive-or operation



Feistel Ciphers

- In other words, if $C_2 = C_1$ (no change in the ciphertext during transmission), then $P_2 = P_1$

Encryption: $C_1 = P_1 \oplus f(K)$

Decryption: $P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\ldots0) = P_1$

The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern.
Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

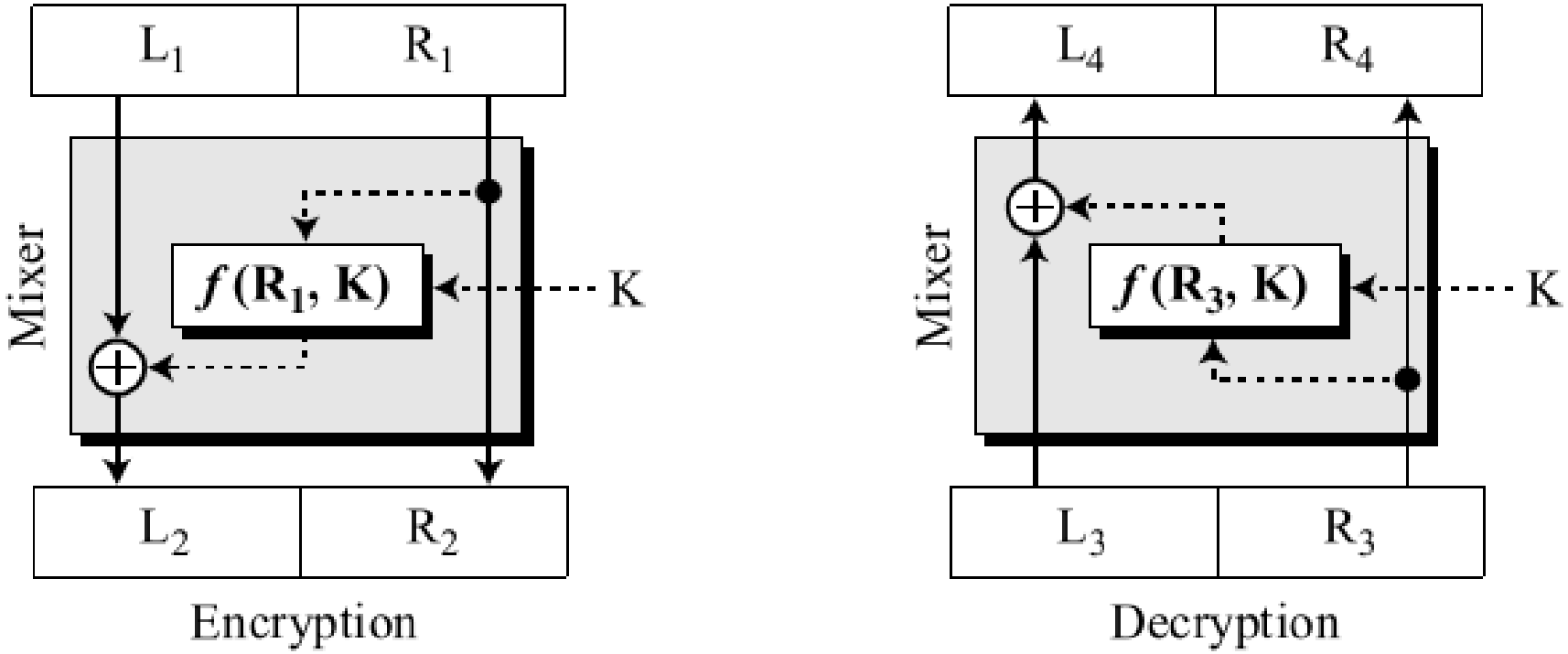
Decryption: $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$ Same as the original P

The function ƒ(101) = 1001 is non-invertible, but the exclusive-or operation allows us to use the function in both encryption and decryption algorithms.

Improvement of the previous Feistel design

- The input to the function to also be part of the plaintext in the encryption and part of the ciphertext in the decryption
- The key can be used as the second input to the function.
- In this way, our function can be a complex element with some keyless elements and some keyed elements.
- To achieve this goal, divide the plaintext and the ciphertext into two equal-length blocks, left and right.
- We call the left block L and the right block R.
- Let the right block be the input to the function, and let the left block be exclusive-ored with the function output.

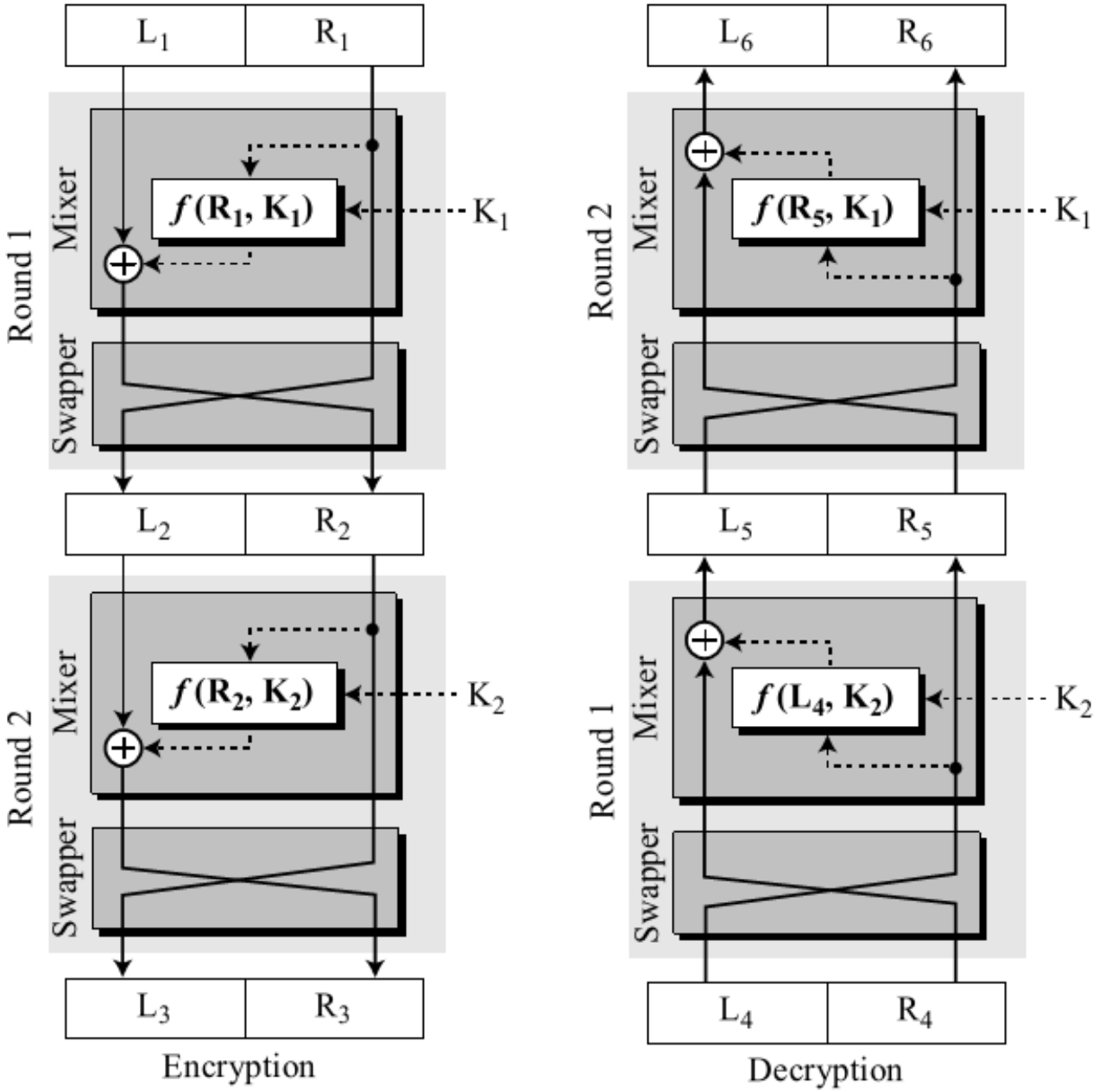
Improvement of the previous Feistel design



$$R_4 = R_3 = R_2 = R_1$$
$$L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus f(R_1, K) = L_1$$

Final Design of the Feistel design

- The preceding improvement has one flaw.
  - The right half of the plain-text never changes.
  - Eve can immediately find the right half of the plaintext by inter-cepting the ciphertext and extracting the right half of it.
- Needs more improvement.
  - First, increase the number of rounds.
  - Second,add a new element to each round: a swapper.



Final design of a Feistel cipher with two rounds

Final design of a Feistel cipher with two rounds

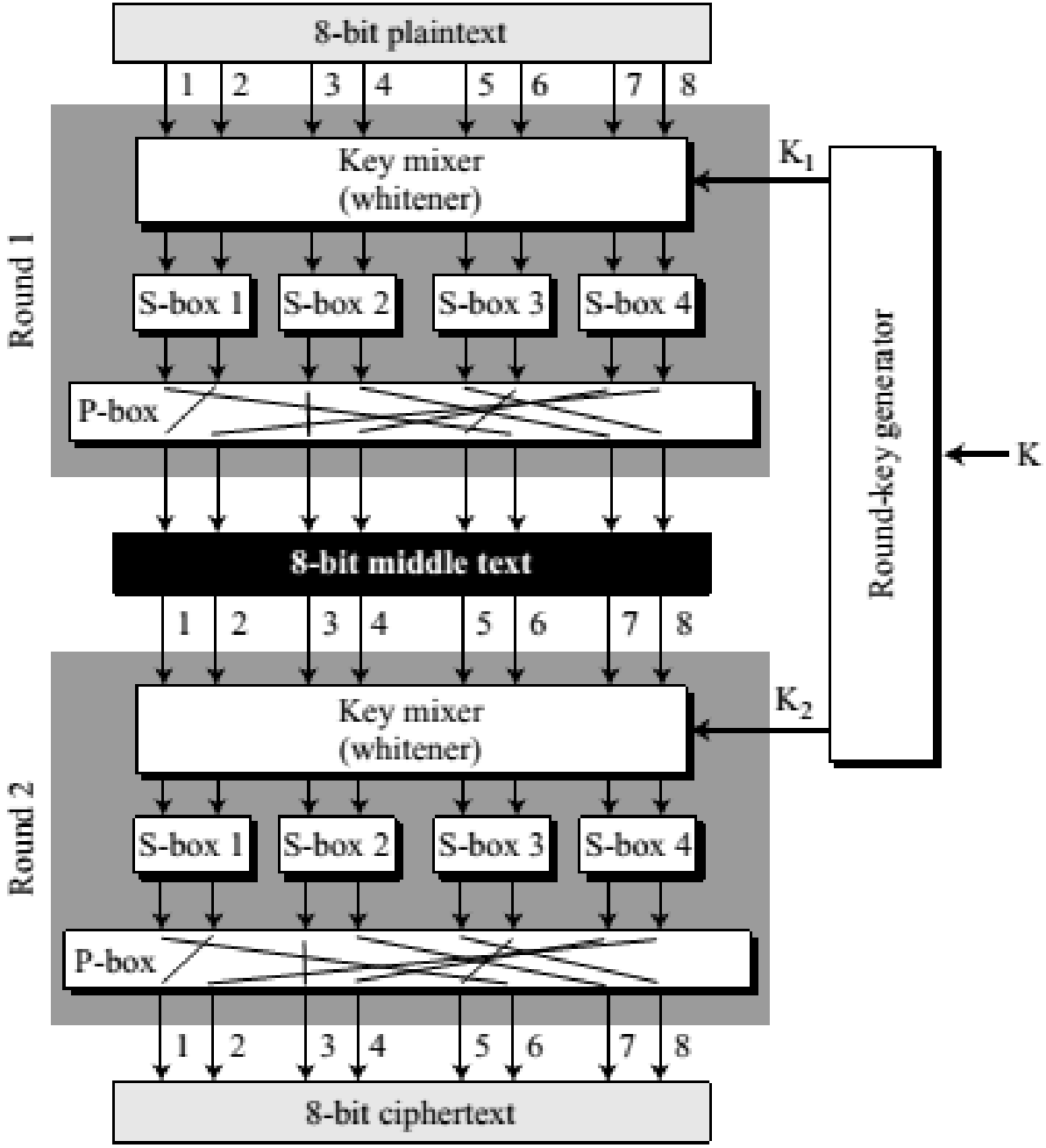
- Because the two mixers are inverses of each other, and the swappers are inverses of each other, it should be clear that the encryption and decryption ciphers are inverses of each other

$$L_5 = R_4 \oplus f(L_4, K_2) = R_3 \oplus f(R_2, K_2) = L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) = L_2$$
$$R_5 = L_4 = L_3 = R_2$$

$$L_6 = R_5 \oplus f(L_5, K_1) = R_2 \oplus f(L_2, K_1) = L_1 \oplus f(R_1, K_1) \oplus f(R_1, K_1) = L_1$$
$$R_6 = L_5 = L_2 = R_1$$

Non-Feistel Ciphers

- A non-Feistel cipher uses only invertible components.
- A component in the encryption cipher has the corresponding component in the decryption cipher.
- For example, S-boxes need to have an equal number of inputs and outputs to be compatible.
- No compression or expansion P-boxes are allowed, because they are not invertible.
- In a non-Feistel cipher, there is no need to divide the plaintext into two halves as we saw in the Feistel ciphers.



Rounds

A simple product cipher with two rounds.

The 8-bit text is mixed with the key to whiten the text (hide the bits using the key).
This is normally done by exclusive-oring the 8-bit word with the 8-bit key.

An example of a non-Feistel cipher

Non-Feistel Ciphers

- Because the only components in each round are the exclusive-or operation (self-invertible), 2 ×2 S-boxes that can be designed to be invertible, and a straight P-box that is invertible using the appropriate permutation table.
- Because each component is invertible, it can be shown that each
- round is invertible.
- We only need to use the round keys in the reverse order.
- The encryption uses round keys K_1 and K_2.
- The decryption algorithm needs to use round keys K_2 and K_1

Attacks on Block Ciphers

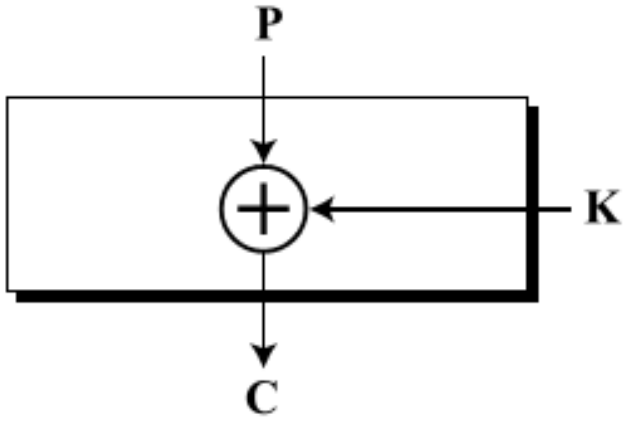
- Brute-force attack on the key is usually infeasible because the keys normally are very large.
- New attacks on block ciphers have been devised that are based on the structure of the modern block ciphers.
- These attacks use differential and linear cryptanalysis techniques.

Differential Cryptanalysis

- This is a chosen-plaintext attack.
- Eve can somehow access Alice's computer, submitting chosen plaintext and obtaining the corresponding ciphertext.
- The goal is to find Alice's cipher key.
- Some ciphers have weaknesses in their structures that can allow Eve to find a relation-ship between the plaintext differences and ciphertext differences without knowing the key.
- Differential cryptanalysis aims to map bitwise differences in inputs to differences in the output in order to reverse engineer the action of the encryption algorithm.

Differential Cryptanalysis

- Example: Assume that the cipher is made only of one exclusive-or operation
- Without knowing the value of the key, Eve can easily find the relationship between plaintext differences and ciphertext differences if by plaintext difference we mean P1 ⊕ P2 and by cipher-text difference, we mean C1⊕C2.
- The following proves that C1 ⊕C2= P1⊕ P2



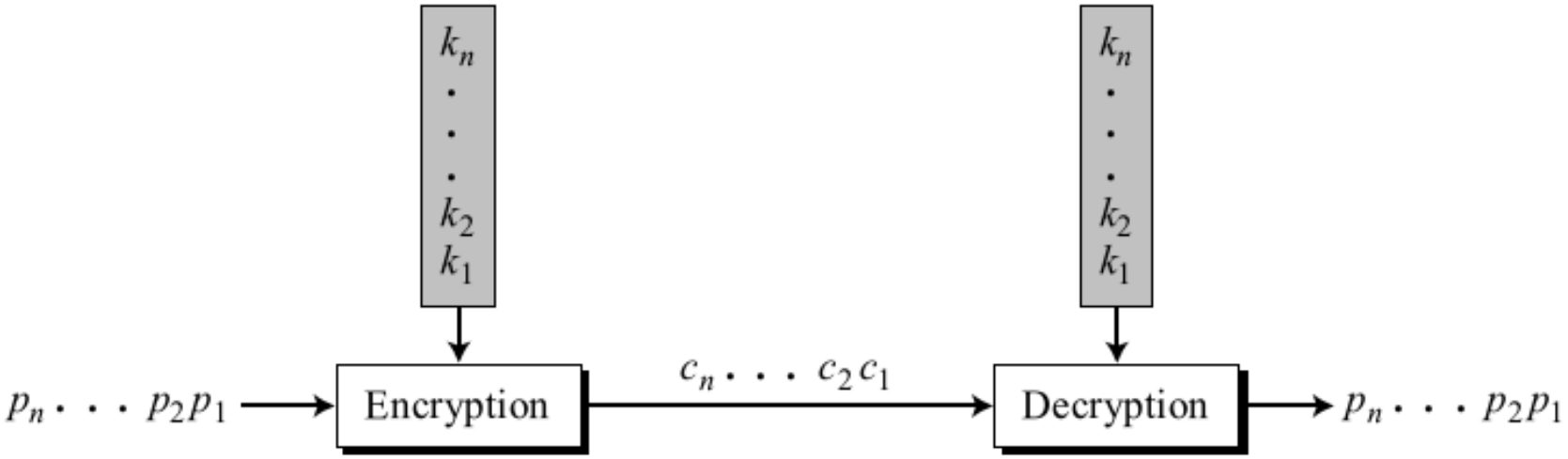
Linear Cryptanalysis

- The analysis uses known-plaintext attacks (versus the chosen-plaintext attacks in differential cryptanalysis).
- The thorough discussion of this attack is based on some probability concepts.
- Linear Approximation
  - In some modern block ciphers, it may happen that some S-boxes are not totally nonlinear.
  - They can be approximated, probabilistically, by some linear functions.
  - In general, given a cipher with plaintext and ciphertext of n bits and a key of m bits, we are looking for some equations of the form

$$(k_0 \oplus k_1 \oplus \cdots \oplus k_x) = (p_0 \oplus p_1 \oplus \cdots \oplus p_y) \oplus (c_0 \oplus c_1 \oplus \cdots \oplus c_z)$$

MODERN STREAM CIPHERS

- In a modern stream cipher,encryption and decryption are done r bits at a time.
- We have a plaintext bit stream P = $p_n \ldots p_2 p_1$, a ciphertext bit stream C = $c_n \ldots c_2 c_1$, and a key bit stream K = $k_n \ldots k_2 k_1$, in which p_i, c_i, and k_i are r-bit words.



MODERN STREAM CIPHERS

- Modern stream ciphers are divided into two broad categories: synchronous and nonsynchronous.
- In a synchronous stream cipher the key is independent of the plaintext or ciphertext.
  - One-Time Pad
- In a nonsynchronous stream cipher, the key depends on either the plaintext or ciphertext.