# Classical Ciphers - Block and Stream Ciphers

19CSE311 Computer Security

Jevitha KP
Department of CSE

# Stream Ciphers

- In a stream cipher, encryption and decryption are done one symbol (such as a character or a bit) at a time.
- We have a plaintext stream, a ciphertext stream, and a key stream.
- Call the plaintext stream P, the ciphertext stream C, and the key stream K.

$$P = P_1P_2P_3, \ldots \qquad C = C_1C_2C_3, \ldots \qquad K = (k_1, k_2, k_3, \ldots)$$

$$C_1 = E_{k1}(P_1) \qquad C_2 = E_{k2}(P_2) \qquad C_3 = E_{k3}(P_3) \ldots$$

# Stream Ciphers

- Characters in the plaintext are fed into the encryption algorithm, one at a time;
- the ciphertext characters are also created one at a time.
- The key stream, can be created in many ways -
  - It may be a stream of predetermined values;
  - it may be created one value at a time using an algorithm.
  - The values may depend on the plaintext or ciphertext characters.
  - The values may also depend on the previous key values

# Stream Ciphers

- **Additive ciphers** can be categorized as stream ciphers in which the key stream is the **repeated value of the key.**

- The key stream is considered as a predetermined stream of keys or K = (k, k, …, k).

- In this cipher, however, **each character in the ciphertext depends only on the corresponding character in the plaintext,** because the key stream is generated independently.

# Stream Ciphers

- The **monoalphabetic substitution ciphers** are also stream ciphers.
- Each value of the key stream in this case is the **mapping of the current plaintext character to the corresponding ciphertext character** in the mapping table.
- **Vigenere ciphers** are also stream ciphers according to the definition.
- The key stream is a repetition of m values, where **m is the size of the keyword. K = (k1, k2, … km, k1, k2, … km, …)**

# Stream Ciphers

- We can  divide stream ciphers based on their key streams.
- We can say that a stream cipher is a **monoalphabetic cipher** if the value of **ki does not depend on the position of the plaintext character** in the plaintext stream; otherwise, the cipher is **polyalphabetic**.
- Additive ciphers are monoalphabetic because **ki in the key stream is fixed**; it does not depend on the position of the character in the plaintext.
- **Monoalphabetic substitution ciphers** are definitely monoalphabetic because ki does not depend on the position of the corresponding character in the plaintext stream; it depends **only on the value of the plaintext character**

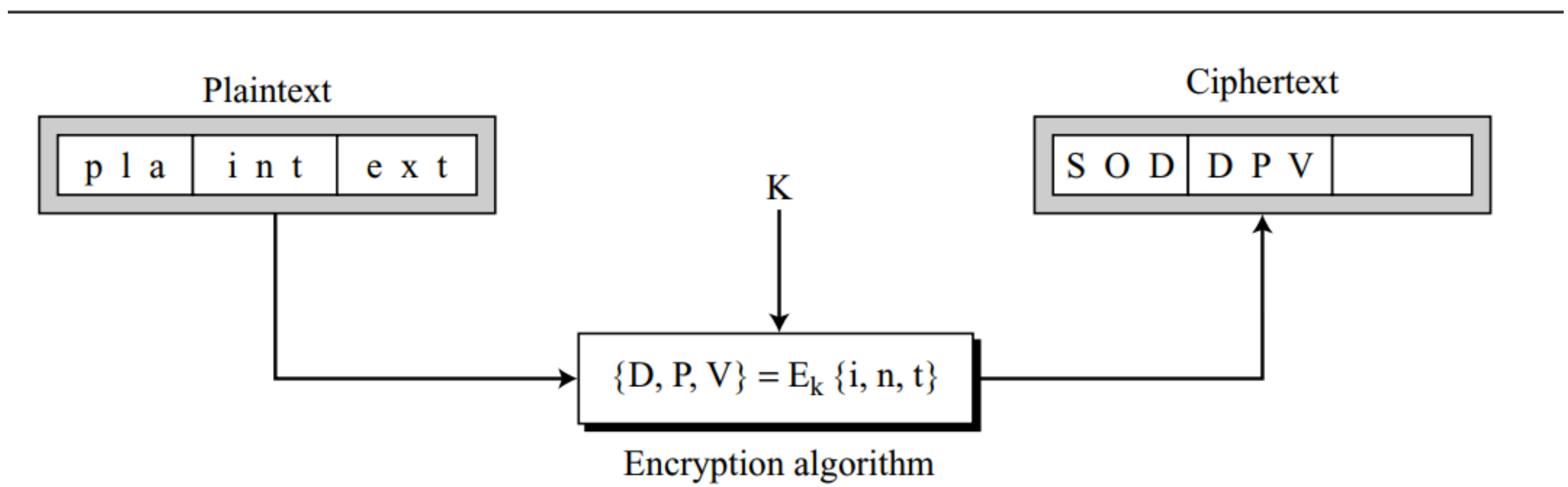
# Stream Ciphers

- Vigenere ciphers are polyalphabetic ciphers because **ki depends on the position of the plaintext character.**
- The dependency is cyclic.
- The key is the same for two characters **m positions** apart

# Block Ciphers

- In a block cipher, a **group of plaintext symbols of size m** (m > 1) are **encrypted together** creating a group of ciphertext of the same size.
- A **single key is used to encrypt the whole block** even if the key is made of multiple values.
- Every block cipher is a polyalphabetic cipher because each character in a ciphertext block **depends on all characters in the plaintext block.**

# Block Ciphers



# Block Ciphers

- **Playfair ciphers** are block ciphers.
  - The size of the block is $m = 2$.
  - Two characters are encrypted together.
- **Hill ciphers** are block ciphers.
  - A block of plaintext, of size 2 or more is encrypted together using a single key (a matrix).
  - In these ciphers, the value of each character in the ciphertext depends on all the values of the characters in the plaintext.
  - Although the key is made of $m \times m$ values, it is considered as a single key

# Combination

- Blocks of plaintext are encrypted individually, but they use a **stream of keys** to encrypt the whole message **block by block.**

- The cipher is a **block cipher** when looking at the **individual blocks**, but it is a **stream cipher** when looking at the **whole message** considering each block as a single unit.

- Each block uses a **different key** that may be generated before or during the encryption process.

# Traditional vs Modern Ciphers

- The traditional symmetric-key ciphers are **character-oriented ciphers** vs modern ciphers are **bit-oriented ciphers,** suitable for computers
- Tranformation to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data.
- Convert these types of data into a stream of bits, to encrypt the stream, and then to send the encrypted stream.
- When text is treated at the bit level, each character is replaced by 8 (or 16) bits, which means that the number of symbols becomes 8 (or 16) times larger.
- Mixing a larger number of symbols increases security.

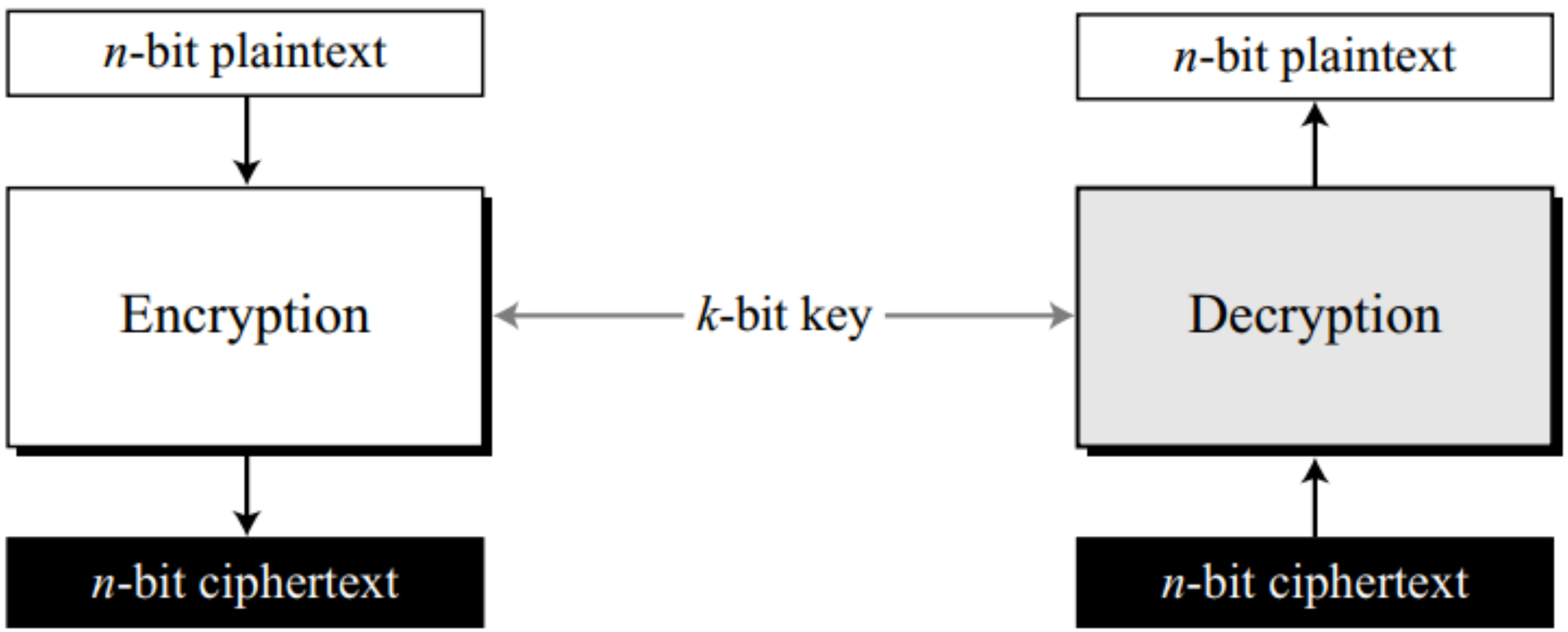
# MODERN BLOCK CIPHERS

- A **symmetric-key modern block cipher** encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext.

- The encryption or decryption algorithm uses a k-bit key.

- The decryption algorithm must be the inverse of the encryption algorithm, and both operations must use the same secret key.

# MODERN BLOCK CIPHERS



# MODERN BLOCK CIPHERS

- If the message has fewer than n bits, padding must be added to make it an **n-bit block;**

- if the message has more than n bits, it should be divided into n-bit blocks and the appropriate padding must be added to the last block if necessary.

- The common values for n are **64, 128, 256, or 512 bits.**

# MODERN BLOCK CIPHERS

- Example:
- 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?

# MODERN BLOCK CIPHERS

- Encoding 100 characters using 8-bit ASCII results in an 800-bit message.
- The plaintext must be divisible by 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.
- Only the last block contains padding.
- The cipher uses the encryption algorithm **thirteen times** to create **thirteen ciphertext blocks.**
- If |M| and |Pad| are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \quad \rightarrow \quad |Pad| = -800 \bmod 64 \quad \rightarrow \quad 32 \bmod 64$$

# Substitution or Transposition

- A modern block cipher can be designed to act as a substitution cipher or a transposition cipher.
- This is the same idea as is used in traditional ciphers, except that the symbols to be substituted or transposed are **bits instead of characters.**
- If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either a 0 or a 1.
- This means that the plaintext and the ciphertext can have a different number of 1's.
- A 64-bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.

# Substitution or Transposition

- If the cipher is designed as a transposition cipher, the bits are only reordered (transposed);
  - there is the same number of 1’s in the plaintext and in the ciphertext.
  - In either case, the number of n-bit possible plaintexts or ciphertexts is 2n, because each of the n bits in the block can have one of the two values, 0 or 1.
- Modern block ciphers are designed as substitution ciphers because the inherent **characteristics of transposition** (preserving the number of 1’s or 0’s) makes the cipher vulnerable to exhaustive-search attacks.

# Keyless Ciphers

- **Keyless Transposition Ciphers**
  - A keyless (or fixed-key) transposition cipher (or unit) can be thought of as a **prewired transposition cipher** when implemented in hardware.
  - The **fixed key (single permutation rule)** can be represented as a table when the unit is implemented in software.
  - The keyless transposition ciphers, called **P-boxes,** are used as building blocks of modern block ciphers.

# Keyless Ciphers

- **Keyless Substitution Ciphers**
    - A keyless (or fixed-key) substitution cipher (or unit) can be thought of as a predefined mapping from the input to the output.
    - The mapping can be defined as a table, a mathematical function, and so on.
    - The keyless substitution ciphers, called **S-boxes**, are used as building blocks of modern block ciphers.

# Properties of Block cipher

- The block ciphers should have two important properties: **diffusion and confusion.**
- **Diffusion** hides 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.**

# Properties of Block cipher

- 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.**

# 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**.

- 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),

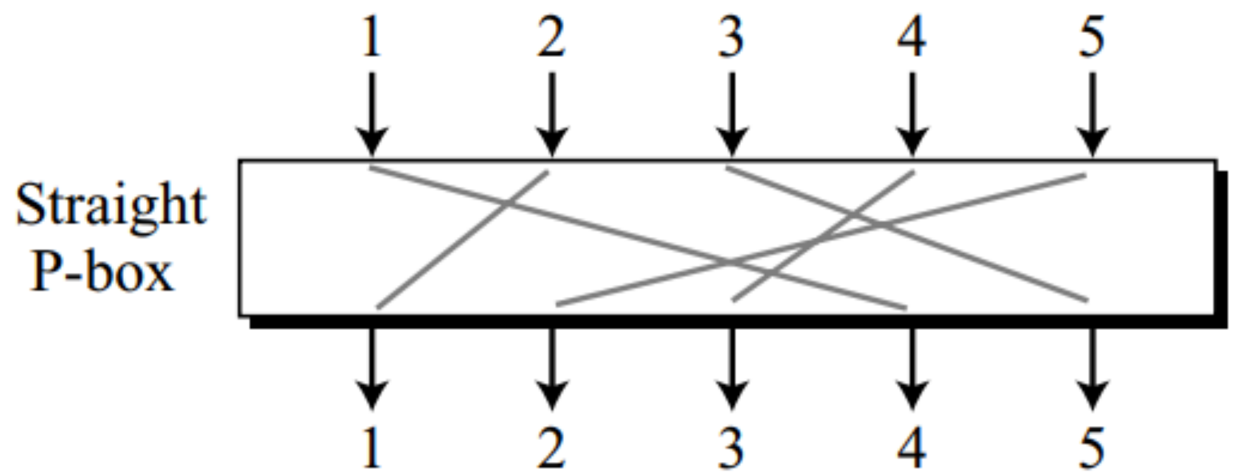
  - substitution units (called S-boxes),

  - and some other units

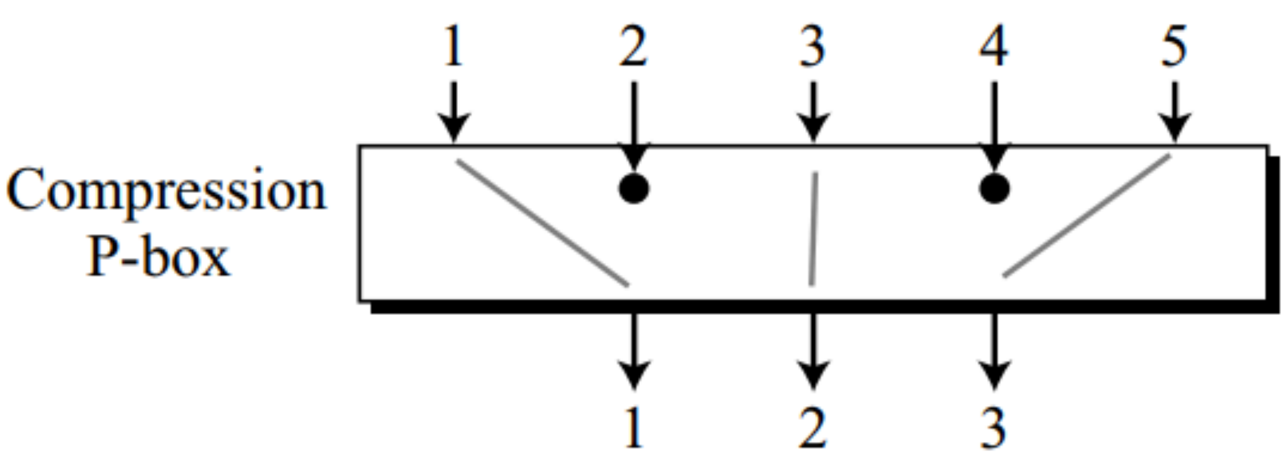
# P-Boxes

- 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,
  - expansion P-boxes, and
  - compression P-boxes

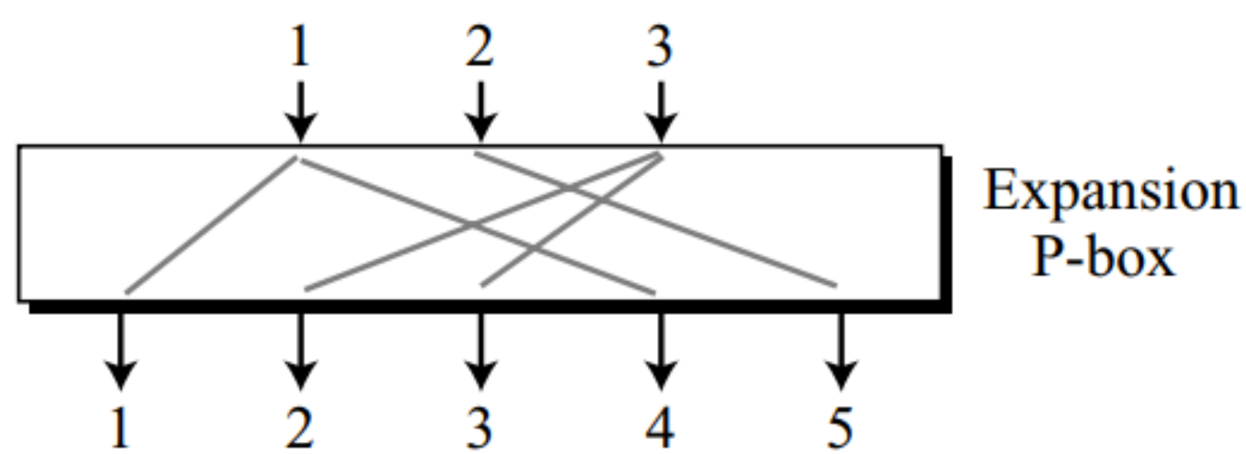
# P-Boxes




**5 × 5 straight P-box**




**5 × 3 compression P-box**




**3 × 5 expansion P-box**

# Straight P-Boxes

- A straight P-Box with n inputs and n outputs is a permutation.
- There are n! possible mappings.
- A P-box can use a key to define one of the **n!** mappings
- P-boxes are normally keyless, which means that the mapping is **predetermined**.
- If the P-box is implemented in hardware, it is prewired
- If it is implemented in software, a **permutation table shows the rule of mapping** and the entries in the table are the inputs and the positions of the entries are the outputs.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

# Straight P-Boxes

- Example:
- Design an $8 \times 8$ 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

# Straight P-Boxes

- Example:
- 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
- **Solution:**
- 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
- 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.
- The compression P-boxes used in modern block ciphers normally are **keyless** with a **permutation table** showing the rule for transposing bits.
- 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).

# Compression P-Boxes

- 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.
- Compression P-boxes are used when we need to **permute bits** and the same time **decrease the number of bits** for the next stage

**Example of a 32 × 24 permutation table**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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.
- A permutation table for an expansion P-box has **m entries, but m – n of the entries are repeated**

# Expansion P-Boxes

- 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.
- Expansion P-boxes are used when we need to permute bits and the same time **increase the number of bits for the next stage.**

**Example of a 12 × 16 permutation table**

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

# Invertibility

- A straight P-box is invertible.
- We can use a straight P-box in the encryption cipher and its inverse in the decryption cipher.
- The permutation tables, however, need to be the inverses of each other.
- **Compression and expansion P-boxes have no inverses**.
- In a compression P-box, an **input can be dropped** during encryption; the decryption algorithm does not know how to replace the dropped bit (a choice between a 0-bit or a 1-bit).
- In an expansion P-box, **an input may be mapped to more than one output** during encryption; the decryption algorithm does not know which of the several inputs are mapped to an output.

# 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.

- Although an S-box can be keyed or keyless, modern block ciphers normally use **keyless S-boxes**, where the mapping from the inputs to the outputs is predetermined.

# Linear Versus Nonlinear S-Boxes

- In an S-box with n inputs and m outputs, with the inputs x0, x1, …, xn and the outputs y1, …, ym, the relationship between the inputs and the outputs can be represented as a set of equations:
- y1 = ƒ1 (x1, x2, …, xn)
- y2 = ƒ2 (x1, x2, …, xn)
- ….
- ym = ƒm (x1, x2, …, xn)

# Linear Versus Nonlinear S-Boxes

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

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

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

# Example

- In an S-box with three inputs and two outputs, we have $y1 = x1 \oplus x2 \oplus x3$ , $y2 = x1$

- The S-box is linear because a1,1 = a1,2 = a1,3 = a2,1 =1 and a2,2 = a2,3 = 0.

- The relationship can be represented by matrices, as shown below:

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

# Invertibility

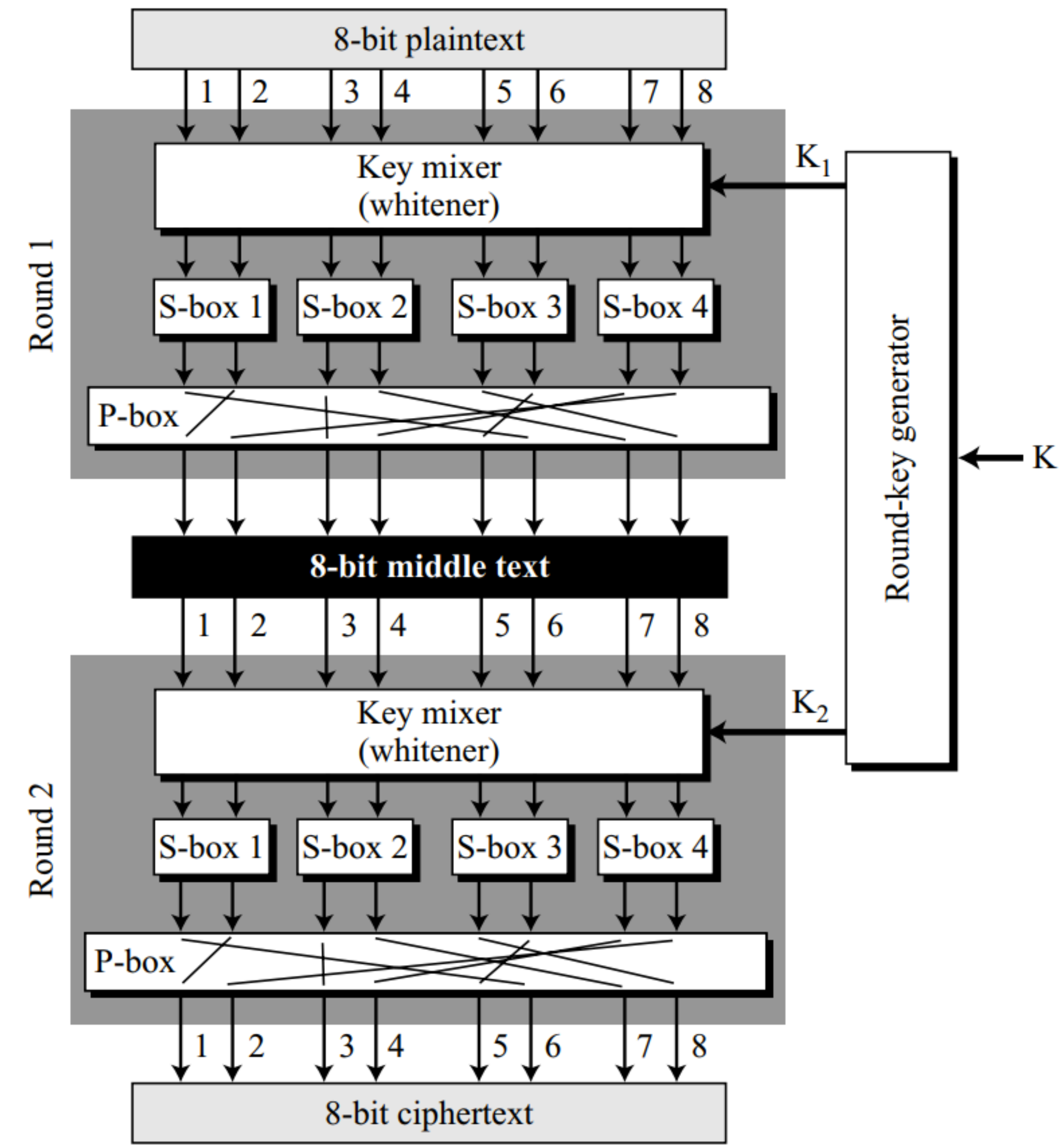
- S-boxes are substitution ciphers in which the relationship between input and output is defined by a table or mathematical relation.
- An S-box may or may not be invertible.
- In an invertible S-box, the number of input bits should be the same as the number of output bits

# Product Cipher

- A product cipher is a complex cipher combining substitution, permutation, and other components
- The product cipher enables the block ciphers to have two important properties: diffusion and confusion.
- 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.
- The block cipher uses a key schedule or key generator that creates different keys for each round from the cipher key.
- 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.

# Product Cipher

**A product cipher made of two rounds**



# 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**.
- Eg: DES
- The ciphers in the second class use only invertible components.
- We refer to ciphers in this class as **non-Feistel ciphers** . Eg: AES

# Feistel Ciphers

- Feistel designed a very intelligent and interesting cipher that has been used for decades.
- 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 question is how the encryption and decryption algorithms are inverses of each other if each has a noninvertible unit.
- Feistel showed that they can be canceled out.
- The mixer in the Feistel design is self-invertible.

# 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.

# Attacks on Block Ciphers

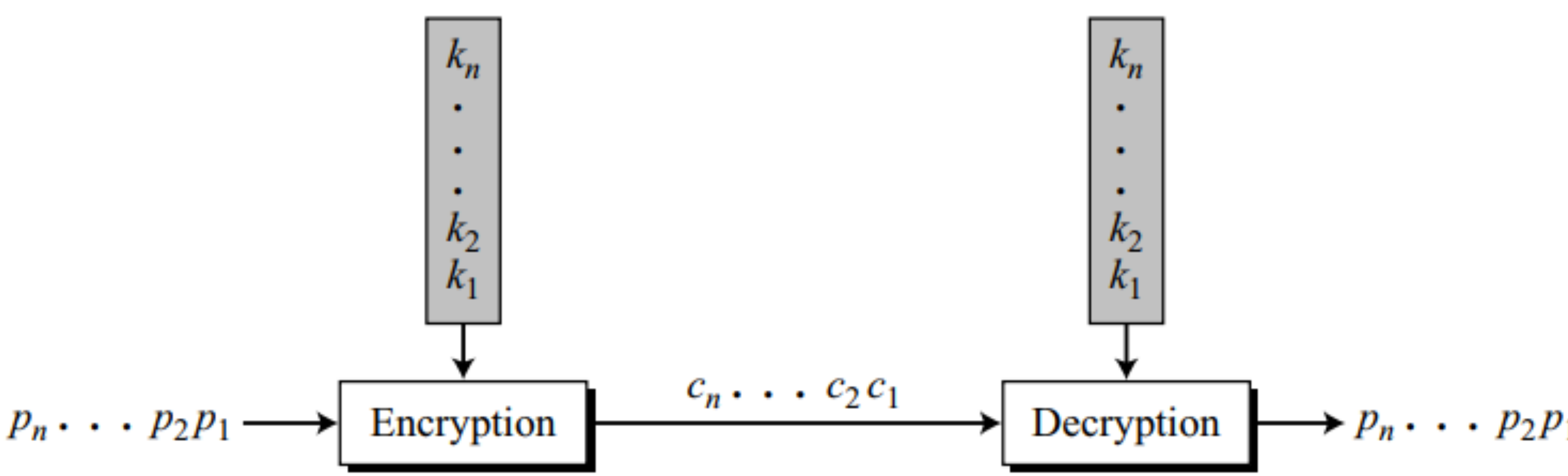
- Differential Cryptanalysis
  - Eli Biham and Adi Shamir introduced the idea of differential cryptanalysis.
  - This is a chosen-plaintext attack;
  - Differential cryptanalysis is based on a **nonuniform differential distribution table** of the **S-boxes** in a block cipher
  - Eve can somehow access Alice's computer, submitting chosen plaintext and obtaining the corresponding ciphertext.
  - The goal is to find Alice's cipher key.

# Attacks on Block Ciphers

- Linear Cryptanalysis
  - Linear cryptanalysis was presented by Mitsuru Matsui in 1993.
  - The analysis uses known plaintext attacks (versus the chosen-plaintext attacks in differential cryptanalysis).
  - The S-box is a linear transformation in which each output is a linear function of input.
  - With this linear component, we can create three linear equations between plaintext and ciphertext bits and solving them for the key

# 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 = pn…p2p1, a ciphertext bit stream C = cn…c2,c1, and a key bit stream K = kn…k2 k1, in which pi, ci, and ki are r-bit words.
- Encryption is **ci = E (ki, pi)**
- decryption is **pi = D (ki, ci)**



# MODERN STREAM CIPHERS

- Stream ciphers are faster than block ciphers.
- The hardware implementation is easier.
- Used when we need to encrypt binary streams and transmit them at a constant rate
- More immune to the corruption of bits during transmission.
- Main issue in modern stream ciphers is generation of the key stream **K = kn…k2k1.**

# Types of Stream Ciphers

- Modern stream ciphers are divided into two broad categories: synchronous and non-synchronous.
- **Synchronous Stream Ciphers**
    - The key stream is **independent** of the plaintext or ciphertext stream.
    - The key stream is generated and used with **no relationship between key bits and the plaintext or ciphertext bits.**
    - Eg: One Time Pad - Simplest and the most secure type of **synchronous stream cipher**
    - Eg: Feedback Shift Register - Linear Feedback Shift Register (LFSR) , Non-Linear Feedback Shift Register (NLFSR)
- **Non-Synchronous Stream Ciphers**
    - Each key in the key stream depends on previous plaintext or ciphertext.
    - Two methods that are used to create different modes of operation for block ciphers (**output feedback mode and counter mode**) actually create stream ciphers.