MODULE 1

- Symmetric Cipher Models- Substitution techniques- Transposition techniques- Rotor machines-Steganography. Simplified DES- Block Cipher principles- The Data Encryption Standard, Strength of DES - Differential and linear Cryptanalysis. Block Cipher Design principles- Block Cipher modes of operations.

CRYTOGRAPHY

- Art of secret writing
- converts data into a format that is unreadable for an unauthorized user.
- Art of achieving security by encoding messages to make them non-readable

SYMMETRIC CIPHER MODELS (1)

- **Form of cryptosystem in which encryption & decryption are performed using the same key**
- A symmetric encryption scheme has five ingredients:

1) Plaintext:

original intelligible message or data that is fed into the algorithm as input.

2) Encryption algorithm:

performs various substitutions and transformations on the plaintext.

SYMMETRIC CIPHER MODELS (2)

3) **Secret key:**

- is also input to the encryption algorithm.
- value independent of the plaintext and of the algorithm.
- The algorithm will produce a different output depending on the specific key being used at the time.
- The exact substitutions and transformations performed by the algorithm depends on the key

SYMMETRIC CIPHER MODELS (3)

4) Ciphertext:

- scrambled message produced as output.
- It depends on the plaintext and the secret key.
- For a given message, two different keys will produce two different cipher texts.
- The cipher text is an apparently random stream of data and, as it stands, is unintelligible.

SYMMETRIC CIPHER MODELS (4)

5) **Decryption algorithm:**

- It takes the cipher text and the secret key and produces the original plaintext

SYMMETRIC CIPHER MODELS (5)

Cryptanalysis

- Techniques for deciphering a message
- without any knowledge of the enciphering details
- One of the attack

Cryptanalyst

- A person tries to break the code
- Or a person who is performing cryptanalysis

Cryptology

- Areas of cryptography & Cryptanalysis together

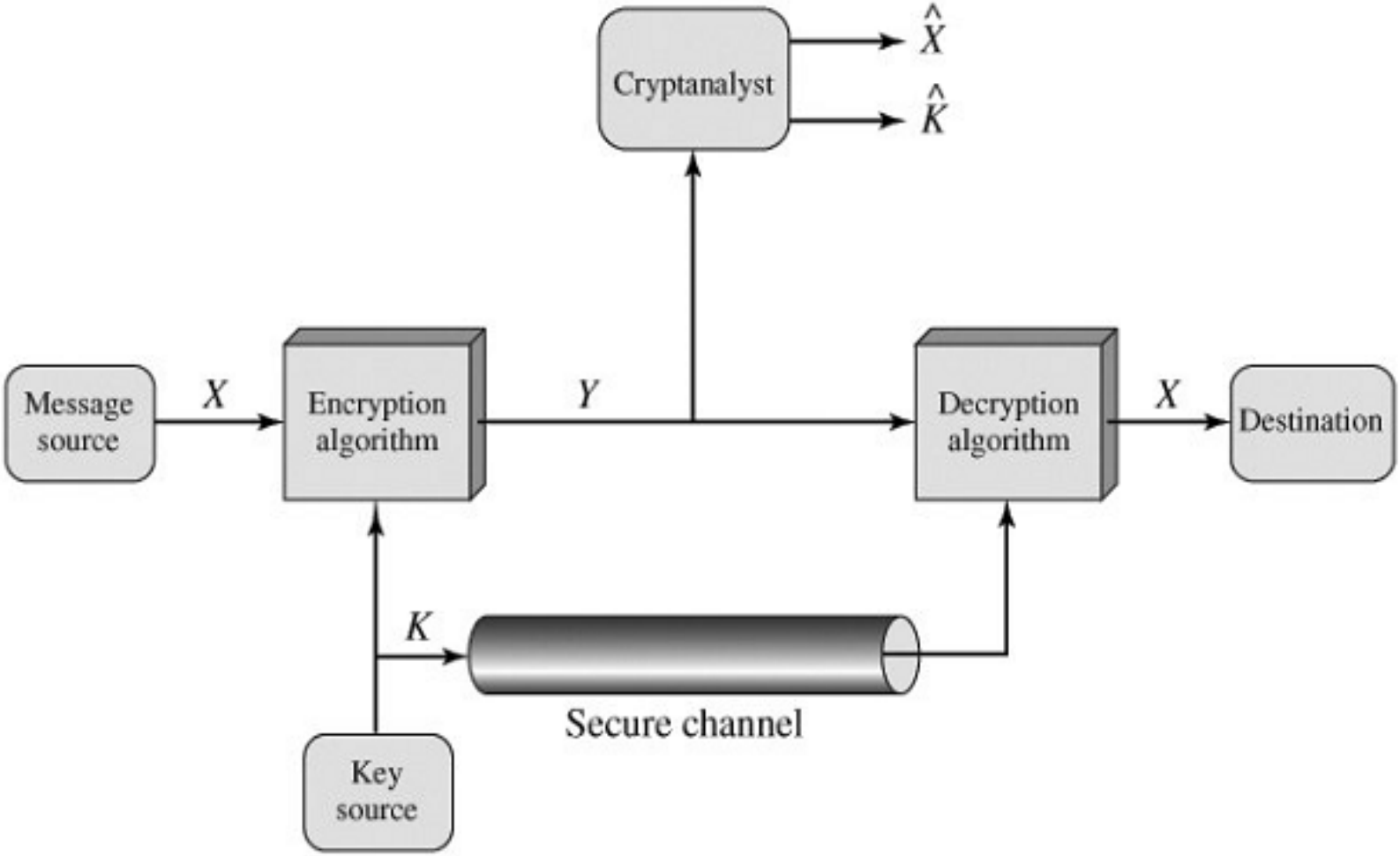
Forms of Cryptographic Systems

- Symmetric/Secret key/Single-Key/Conventional Cryptographic Systems
- Asymmetric/Public Key Cryptographic Systems
- Hash Functions

Requirements for secure use of conventional encryption

- strong encryption algorithm.
- Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the **key secure**.

Model of Conventional Cryptosystem (1)



Model of Conventional Cryptosystem (2)

- plaintext, $X = [X_1, X_2, ..., X_M]$.
- key of the form $K = [K_1, K_2, ..., K_J]$
- ciphertext $Y = [Y_1, Y_2, ..., Y_N]$.
- **$Y = E(K, X)$**
- **$X = D(K, Y)$**

Characteristics of Cryptographic systems: (1)

- 1. The **type of operations used for transforming plaintext to cipher text**. All encryption algorithms are based on two general principles:
  - **Substitution:** in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element.
  - **Transposition:** in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost.

Characteristics of Cryptographic systems: (2)

- 2. **number of keys used.**
  - If both sender and receiver use the **same key**, the system is referred to as **symmetric, single-key, secret-key, or conventional encryption**.
  - If the sender and receiver use **different keys**, the system is referred to as **asymmetric, two-key, or public-key encryption.**

Characteristics of Cryptographic systems: (3)

- 3. **way in which the plaintext is processed**.
  - **block cipher** processes the input one block of elements at a time, producing an output block for each input block.
  - **stream cipher** processes the input elements continuously, producing output one element at a time, as it goes along.

General approaches to attack a conventional encryption scheme

- Cryptanalysis
- Brute-force attack

1. Cryptanalysis

Cryptanalytic attacks rely on

- the **nature of the algorithm**
- some knowledge of the **general characteristics of the plaintext**
- or even some sample plaintext - ciphertext pairs.

2.Brute-force attack

- The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.

Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Cipher text only	❖ Encryption algorithm ❖ Cipher text
Known plaintext	❖ Encryption algorithm ❖ Cipher text ❖ One or more plaintext- cipher text pairs formed with the secret key
Chosen plaintext	❖ Encryption algorithm ❖ Cipher text ❖ Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key
Chosen Cipher text	❖ Encryption algorithm ❖ Cipher text ❖ Purported cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	❖ Encryption algorithm ❖ Cipher text ❖ Plaintext message chosen by cryptanalyst, together with its corresponding cipher text generated with the secret key ❖ Purported cipher text chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Encryption Techniques

- The two basic building blocks of all encryption techniques are:
  - **Substitution**
  - **Transposition**

Substitution Techniques

- the letters of plaintext are replaced by other letters or by numbers or symbols.
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with cipher text bit patterns.

Types of Substitution Techniques

- Caesar Cipher
- Monoalphabetic Ciphers
- Play fair Cipher
- Hill Cipher
- Polyalphabetic Ciphers

1. Caesar Cipher

- earliest known, and the simplest
- by Julius Caesar.
- involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.

Example

Plaintext: **meet me after the toga party**

Ciphertext: **PHHW PH DIWHU WKH WRJD SDUWB**

- alphabet is wrapped around, so that the letter following Z is A.

Encryption Algorithm

- assign a numerical equivalent to each letter. Then the algorithm can be expressed as follows.
- For each plaintext letter *p*, substitute the ciphertext letter *C*:

$$C = E(3, p) = (p + 3) \bmod 26$$

- A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(k, p) = (p + k) \bmod 26$$

Where ***k*** takes on a value in the range 0 to 25.

Decryption Algorithm

$p = D(k, C) = (C - k) \bmod 26$

Disadvantages

- a brute-force cryptanalysis is easily performed: simply try all the 25 possible keys.
- The encryption and decryption algorithms are known.
- The language of the plaintext is known and easily recognizable.

2. Monoalphabetic Ciphers (1)

- Replace an alphabet with any other alphabet randomly
- A can be replaced with any alphabet from B to Z & B can be replaced with any alphabet from A or C to Z
- Difficult to attack
- In general, there are *n*! permutations of a set of *n* elements.

Monoalphabetic Ciphers (2)

Drawback

- If the cryptanalyst knows the nature of the plaintext, then the analyst can exploit the regularities of the language.
- the relative frequency of the letters can be determined and compared to a standard frequency distribution for English.

Monoalphabetic Ciphers (3)

- If the message were long enough, this technique alone might be sufficient
- A powerful tool is to look at the frequency of two-letter combinations, known as **digrams**.
- common digram is ‘th’.
- “The” is the most frequent trigram (three-letter combination) in English.

Monoalphabetic Ciphers (4)

- Homophonic Substitution Cipher
  - Encrypt multiple letters of plaintext
  - Use multiple cipher alphabets.
- Eg:- A - {D,H,P,R} and B – {E,F,Q,S}.
- Polygram Substitution Cipher
  - rather than replacing a plaintext alphabet with a ciphertext alphabet, a block of alphabet is replaced with another block (HELLO – YUQQW)

Play fair Cipher (1)

- best-known multiple-letter encryption cipher
- which treats digrams in the plaintext as single units and translates these units into cipher text digrams.
- based on the use of a 5 * 5 matrix of letters constructed using a keyword.

Step 1

Matrix construction

- The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top tobottom,
- then fill the remainder of the matrix with the remaining letters in alphabetic order.
- The letters I and J count as one letter.

Step 2

Rules for encrypting Plaintext

1. Encrypt two letters at a time.

2. Repeating plaintext letters that are in the same pair are separated with filler letter, such as x, so that **balloon → ba lx lo on**

3. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. **ar → RM**

Step 3

- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. **mu** → **CM**
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. **hs** → **BP** and **ea** → **IM** (or JM, as the encipherer wishes).

In this case, the keyword is ***monarchy***.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Advantages

- The Playfair cipher is a great advance over simple monoalphabetic ciphers.
- There are only 26 letters; there are 26 * 26 = 676 digrams, so that identification of individual digrams is more difficult.

4. Hill Cipher (1)

- developed by Lester Hill in 1929.
- takes *m* successive plaintext letters and substitutes for them *m* cipher text letters
- substitution is determined by *m* linear equations in which each character is assigned a numerical value
- (a = 0, b = 1, c, z = 25).

Hill Cipher (2)

- For $m = 3$, the system can be described as

$$c_1 = (k_{11}p_1 + k_{21}p_2 + k_{31}p_3) \bmod 26$$

$$c_2 = (k_{12}p_1 + k_{22}p_2 + k_{32}p_3) \bmod 26$$

$$c_3 = (k_{13}p_1 + k_{23}p_2 + k_{33}p_3) \bmod 26$$

Hill Cipher (3)

$$(c_1\ c_2\ c_3) = (p_1\ p_2\ p_3)\begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \bmod 26$$

C = PK mod 26

Where,

C and P --- row vectors of length 3 representing the plaintext and ciphertext

K - -- 3 * 3 matrix representing the encryption key

Operations are performed mod 26.

Example – Hill Cipher

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

- PT:- pay more money
- CT:- LNSHDLEWWWMTRW

Multiletter Substitution Cipher (Hill Cipher)

- Substitute m sucessive plaintext letter with m ciphertext letters (e.g. m=3)
- encryption algorithm:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

where $K = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix}$ is the key

- decryption algorithm:

$$\begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix}^{-1} \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} \bmod 26$$

- key space = 26^{m^2}

Polyalphabetic **substitution** Ciphers

- improve security using multiple cipher alphabets
- make cryptanalysis harder with more alphabets to guess
- use a key to select which alphabet is used for each letter of the message
- use each alphabet in turn
- repeat from start after end of key is reached

Vigenère Cipher (1)

- simplest polyalphabetic substitution cipher
- effectively multiple caesar ciphers
- Matrix known as vigenere tableau is constructed.
- Each of 26 ciphers is laid out horizontally with the key letter for each cipher to its left.
- Normal alphabet for the plaintext runs across the top

Vigenère Cipher (2)

- key is multiple letters long $K = k_1\ k_2\ \ldots\ k_d$
- repeat from start after d letters in message
- Key is a repeating keyword
- Given a key letter ‘x’ & a plaintext letter ‘y’ , the ciphertext letter is at the intersection of the row labeled ‘x’ & the coloumn labeled ‘y’.
- decryption simply works in reverse

Vigenère Cipher

- Write the plaintext out. Write the keyword repeated above it
- Use each key letter as a Caesar cipher key. Encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

key:	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t	i	v	e
plaintext:	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d	s	a	v	e	y	o	u	r	s	e	l	f
ciphertext:	Z	I	C	V	T	W	Q	N	G	R	Z	G	V	T	W	A	V	Z	H	C	Q	Y	G	L	M	G	J

Plaintext

Key

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
W	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Breaking the Vigenère Cipher

- Repetitions in ciphertext give clues to period, so find same plaintext an exact period apart which results in the same ciphertext
- Eg:- repeated “VTW” in previous example suggests size of 9
- If message is long enough , there will be a number of such repeated ciphertext sequences
- Analyst should be able to make a good guess about the keyword length.

Autokey System (1)

- Non-repeating keyword that is as long as the msg itself
- Proposed by vignere
- Keyword is concatenated with the plaintext to provide a running key.

key: ***deceptivewearediscoveredsav***

Plaintext: **wearediscoveredsaveyourself**

Ciphertext: **ZICVTWQNGKZEIIGASXSTSLVVWLA**

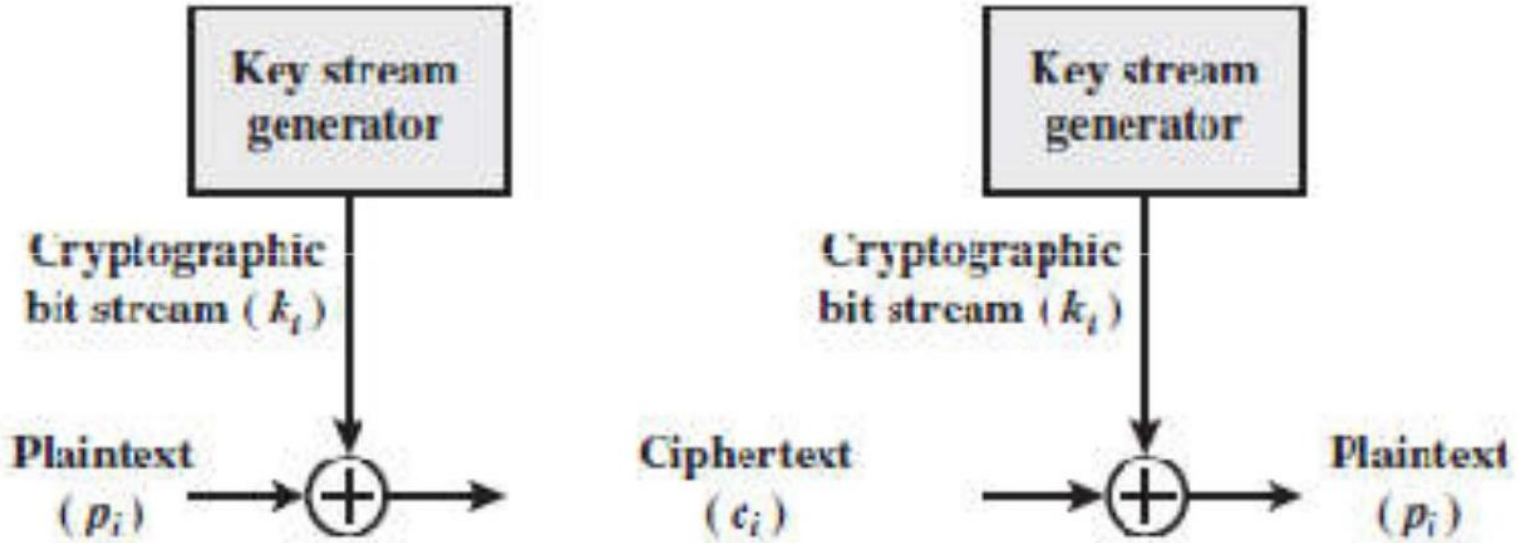
Autokey System (2)

- Vulnerable to cryptanalysis
- Key & the plaintext share the same

Vernam Cipher (1)

- choose a keyword that is as long as the plaintext and has no statistical relationship to it.
- This system works on binary data (bits) rather than letters.

Vernam Cipher (2)



Vernam Cipher (3)

Encryption

$$ci = pi \oplus ki$$

where

pi = ith binary digit of plaintext

ki = ith binary digit of key

ci = ith binary digit of ciphertext

$\oplus$ = exclusive-or (XOR) operation

Vernam Cipher (4)

Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key.

Decryption

Decryption simply involves the same bitwise operation:

$$pi = ci \oplus ki$$

One-Time Pad(1)

- a random key is used that is as long as the message, so that the key need not be repeated.
- the key is to be used to encrypt and decrypt a single message, and then is discarded.
- Each new message requires a new key of the same length as the new message
- **one-time pad**, is unbreakable.

One-Time Pad(2)

- It produces random output that bears no statistical relationship to the plaintext.
- Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code.

TRANSPOSITION TECHNIQUES

- A kind of mapping is achieved by performing some sort of permutation on the plaintext letters.
- This technique is referred to as a transposition cipher.

1) Rail fence (1)

- The simplest cipher is the **rail fence** technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.
- To encipher the message "**meet me after the toga party**" with a railfence of depth 2, we write the following:

Rail Fence cipher (2)

- write message letters out diagonally over a number of rows
- then read off cipher row by row
- eg. write message out as:

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

- giving ciphertext

```
MEMATRHTGPRYETEFETEOAAT
```

Row Transposition Ciphers

- a more complex transposition
- write letters of message out in rows over a specified number of columns
- then reorder the columns according to some key before reading off the rows

Key: **3 4 2 1 5 6 7**

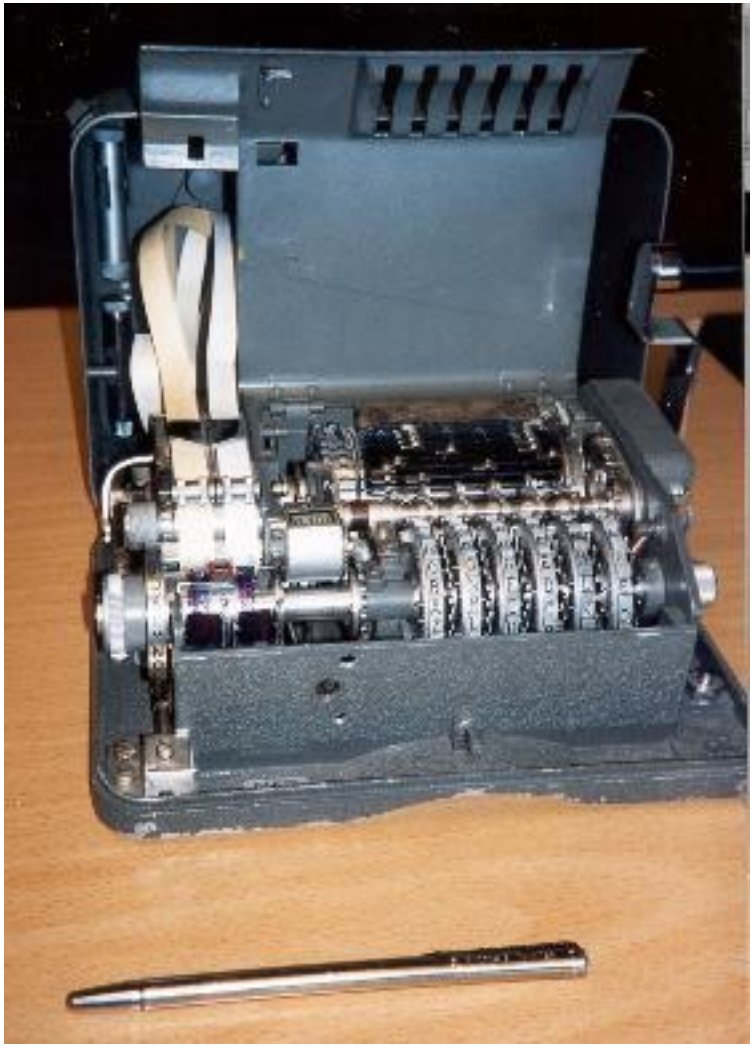
Plaintext: **a t t a c k p**
o s t p o n e
d u n t i l t
w o a m x y z

Ciphertext: **TTNAAPTMTSUOAODWCOIXKNLYPETZ**

Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics
- Hence consider using several ciphers in succession to make harder, but:
  - Two substitutions make a more complex substitution
  - Two transpositions make more complex transposition
  - But a substitution followed by a transposition makes a new much harder cipher
- This is bridge from classical to modern ciphers

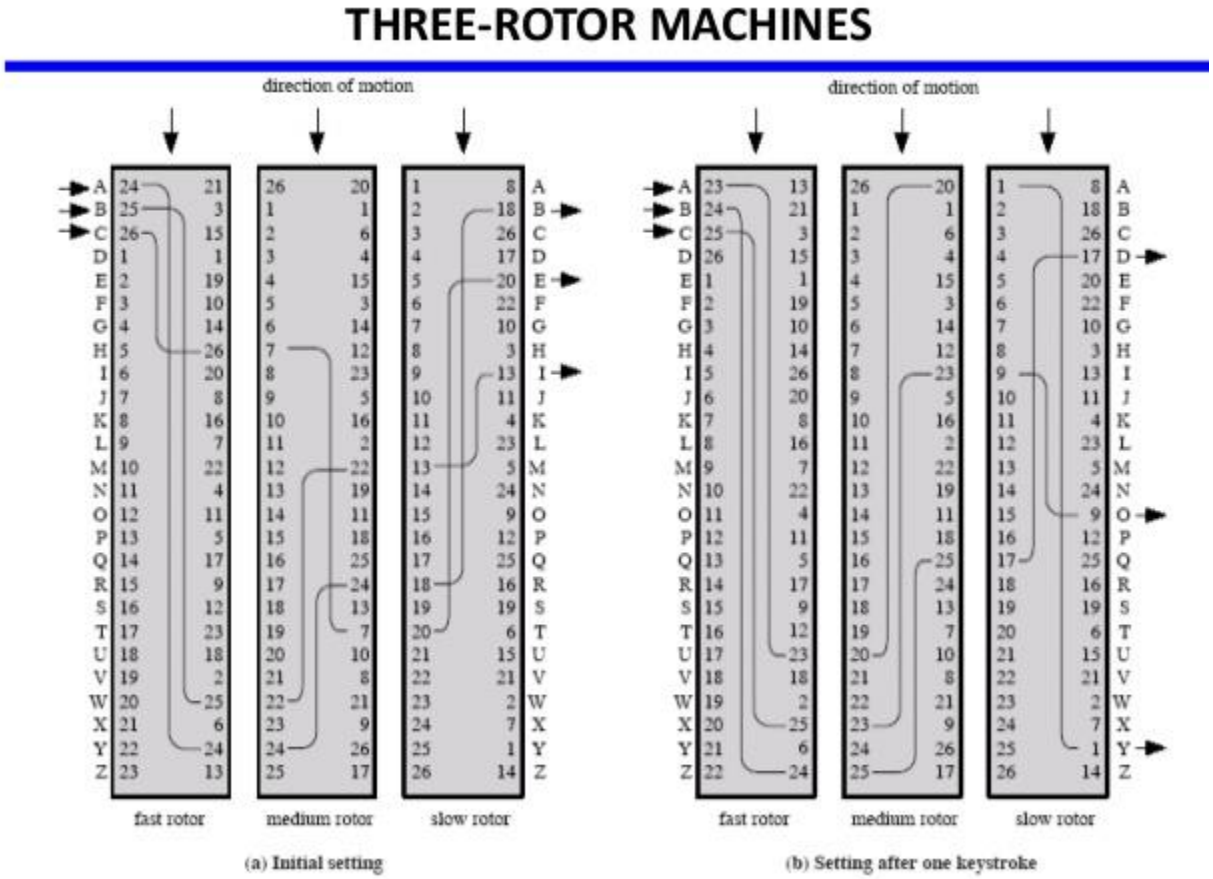
Hagelin Rotor Machine



Rotor Machines (1)

- before modern ciphers, rotor machines were most common complex ciphers in use
- implemented a very complex, varying substitution cipher
- used a series of cylinders, each giving one substitution, which rotated and changed after each letter was encrypted
- with 3 cylinders having 26^3=17576 alphabets

Rotor Machines (2)



Rotor Machines (2)

- Multiple stages of encryption
- Consist of a set of independently rotating cylinders through which electric pulses can flow
- Each cylinder has 26 input pins & 26 output pins.
- Each input pin connects to a unique output pin
- A single cylinder defines a monoalphabetic substitution

- If an operator depresses the key for the letter A, an electric signal is applied to the first pin of the first cylinder & flows through the internal connection to the 25^{th} output pin.
- After each input key is depressed,the cylinder rotates one position,so that the internal connections are shifted accordingly

- After 26 letters of plain text, the cylinder would be back to the initial position
- Output pins are connected to the input pins of the next
- For every complete rotation of the output cylinder, the middle cylinder rotates one position
- Finally for every complete rotation of the middle cylinder, the inner cylinder rotates one pin position

Steganography (1)

- Plain text can be hidden in two ways
  - methods of steganography conceal the existence of the message
  - methods of cryptography render the message unintelligible to outsiders by various transformations of the text.
- Simple but time consuming
  - Arrangement of words or letters with in an apparently normal text spells out the real message

    Eg: first letter of each word of the overall msg spells out the hidden message

Steganography (2)

- Various other techniques are the following:
  - Character marking
  - Invisible ink
  - Pin punctures
  - Typewriter correction ribbon

Character marking

- Selected letters of printed or typewritten text are overwritten in pencil.
- The marks are ordinarily not visible unless the paper is held at an angle to bright light.

Invisible ink

- A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.

Pin punctures

- Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.

Typewriter correction ribbon

- Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

Steganography

Drawbacks

- It requires a lot of overhead to hide a relatively few bits of information.
- Once the system is discovered, it becomes virtually worthless.
- This problem, too, can be overcome if the insertion method depends on some sort of key.
- Alternatively, a message can be first encrypted and then hidden using steganography.

Steganography

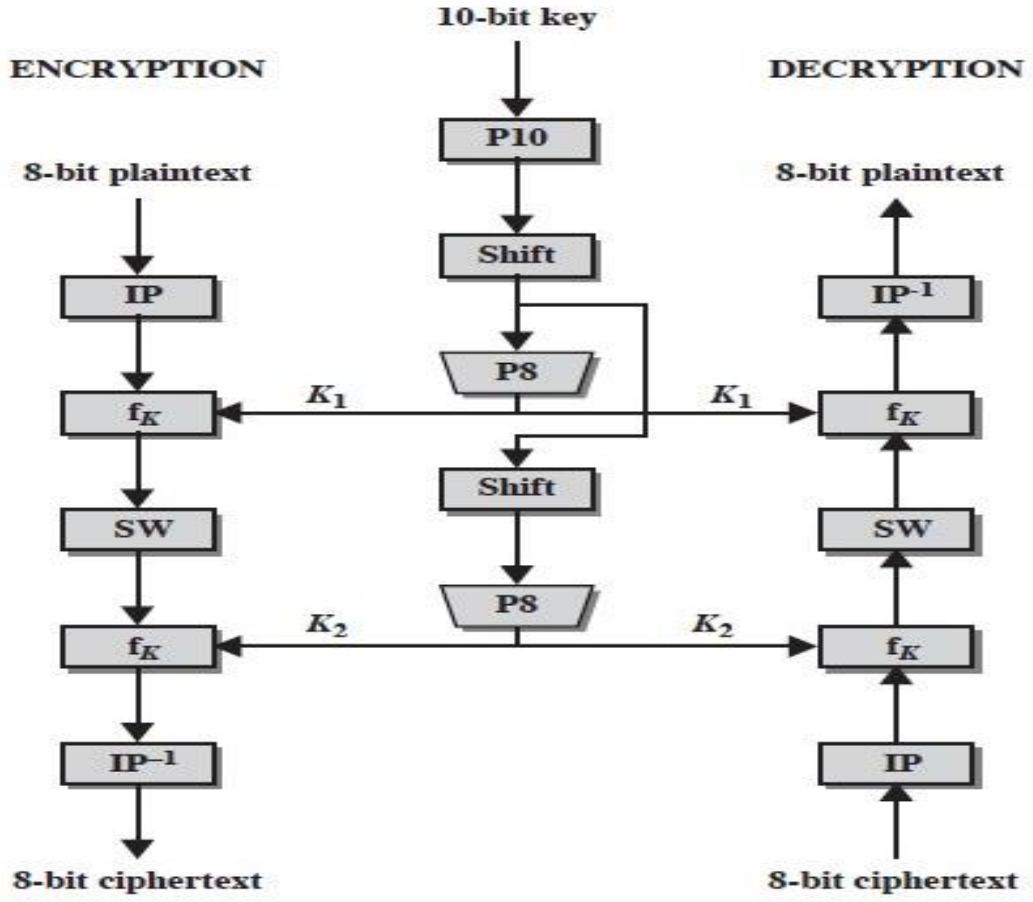
Advantages

- It can be employed by parties who have something to lose should the fact of their secret communication be discovered.
- Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

SIMPLIFIED DES

- **Encryption**
- **Simplified Data Encryption Standard**
- The S-DES encryption algorithm takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input
- produces an 8-bit block of ciphertext as output.

Overall structure of the simplified DES



S-DES encryption algorithm involves five functions:

1) an initial permutation (IP)
2) a complex function labeled f_K, which involves both permutation and substitution operations and depends on a key input
3) a simple permutation function that switches (SW) the two halves of the data
4) the function f_K again
5) a permutation function that is the inverse of the initial permutation (IP^{-1}).

S-DES

- The use of multiple stages of permutation and substitution results in a more complex algorithm, which increases the difficulty of cryptanalysis.
- The function f_K takes as input not only the data passing through the encryption algorithm, but also an 8-bit key.

S-DES

- In this case, the key is first subjected to a permutation (P10).
- Then a shift operation is performed.
- The output of the shift operation then passes through a permutation function that produces an 8-bit output (P8) for the first subkey (K_1).
- The output of the shift operation also feeds into another shift and another instance of P8 to produce the second subkey (K_2).

The encryption algorithm is expressed as a composition of functions:

$$IP^{-1} \circ f_{K_2} \circ SW \circ f_{K_1} \circ IP$$

which can also be written as:

$$\text{ciphertext} = IP^{-1}\left(f_{K_2}\left(SW\left(f_{K_1}\left(IP(\text{plaintext})\right)\right)\right)\right)$$

where

$$K_1 = P8\left(\text{Shift}\left(P10(\text{key})\right)\right)$$

$$K_2 = P8\left(\text{Shift}\left(\text{Shift}\left(P10(\text{key})\right)\right)\right)$$

Decryption is also shown in Figure C.1 and is essentially the reverse of encryption:

$$\text{plaintext} = IP^{-1}\left(f_{K_1}\left(SW\left(f_{K_2}\left(IP(\text{ciphertext})\right)\right)\right)\right)$$

S-DES KEY GENERATION

- First, permute the key in the following fashion.
- Let the 10-bit key be designated as (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}).
- Then the permutation P10 is defined as:
- P10 (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)

P10 can be defined by:

P10									
3	5	2	7	4	10	1	9	8	6

- This table is read from left to right
- each position in the table gives the identity of the input bit that produces the output bit in that position.
- So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on.

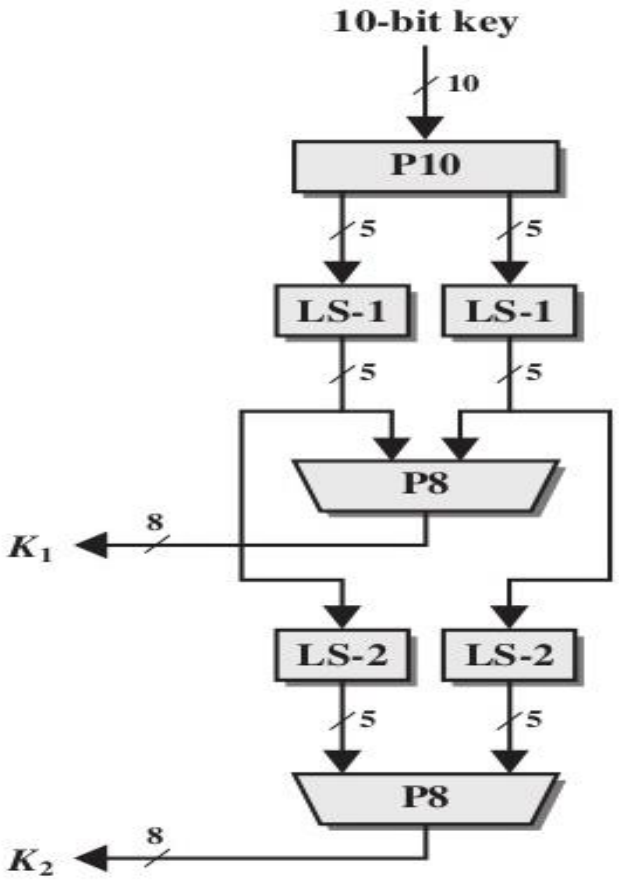
- For example, the key (1010000010) is permuted to (1000001100).
- Next, perform a circular left shift (LS-1), or rotation, separately on the first
- five bits and the second five bits. In our example, the result is (00001 11000).

Next apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:

P8							
6	3	7	4	8	5	10	9

- The result is subkey 1 (K1). In our example, this yields (10100100).
- We then go back to the pair of 5-bit strings produced by the two LS-1
- functions and performs a circular left shift of 2 bit positions on each string. In our
- example, the value (00001 11000) becomes (00100 00011). Finally, P8 is applied
- again to produce K2. In our example, the result is (01000011).

S-DES Key Generation

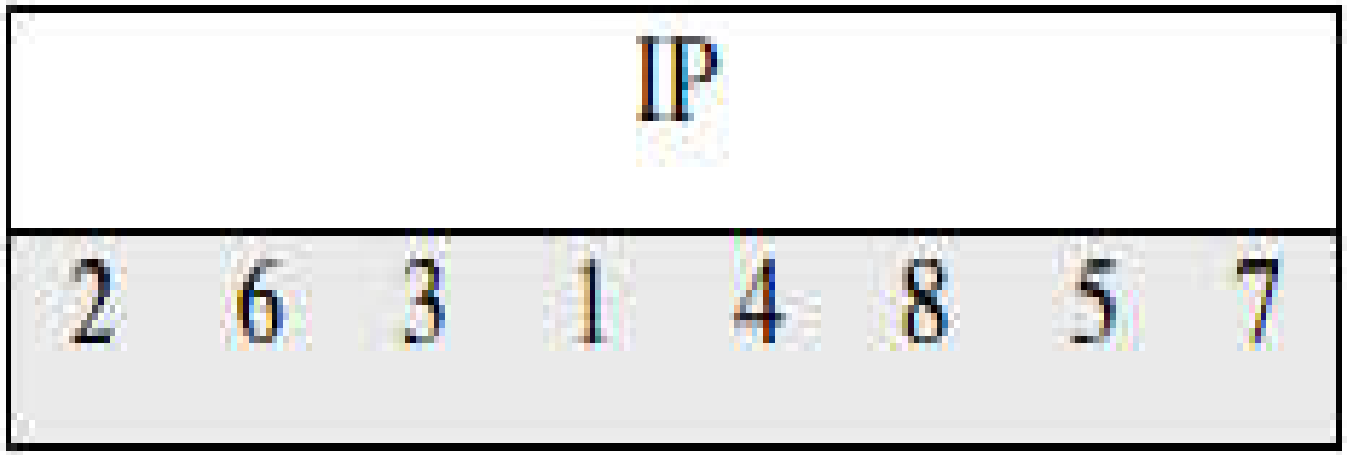


S-DES ENCRYPTION

1) Initial and Final Permutations

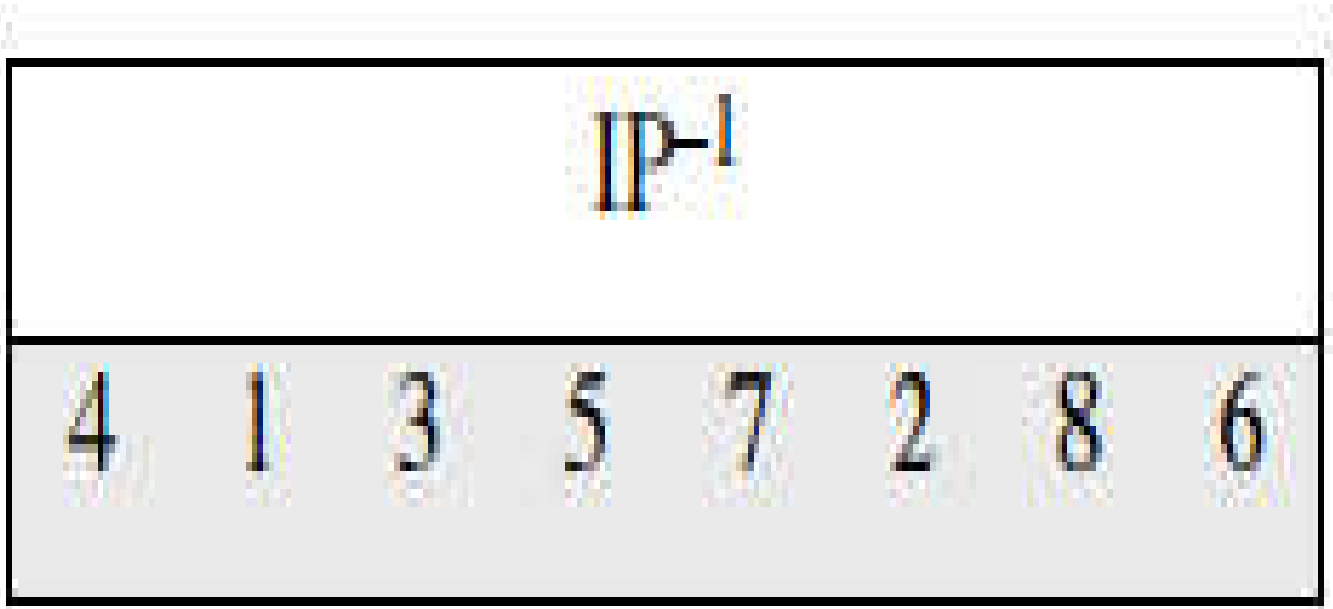
- The input to the algorithm is an 8-bit block of plaintext, which we first permute using the IP function:

Initial Permutation



Inverse of IP

$IP^{-1}(IP(X)) = X$



2) The Function f_K

- The function $\mathbf{f_K}$, consists of a combination of permutation and substitution functions. The functions can be expressed as follows:
- Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to $\mathbf{f_K}$, and let F be a mapping (not necessarily one to one) from 4-bit strings to 4-bit strings.

- SK – Sub key

$$f_K(L, R) = (L \oplus F(R, SK), R)$$

Mapping F.

i/p → 4-bit numbes $(n_1, n_2\ n_3\ n_4)$ →

Right most 4 bits

Step 1 :- expansion/ permutation operation.

E/P						
4	1	2	3	2	3	4	1

Step 2 :- depict the result in this fashion

n_4	n_1	n_2	n_3
n_2	n_3	n_4	n_1

Step3 :- The 8-bit sub-key is add
to this value using XOR.

$n_A + k_{11}$	$n_1 + k_{12}$	$n_2 + k_{13}$	$n_3 +$
$n_2 + k_{15}$	$n_3 + k_{16}$	$n_4 + k_{17}$	$n_1 +$

↓ result renamed as

$P_{0.0}$	$P_{0.1}$	$P_{0.2}$	$P_{0.3}$ → fed into S-bo
$P_{1.0}$	$P_{1.1}$	$P_{1.2}$	$P_{1.3}$ → fed into S-box

Step A

First 4-bits (1st row) fed into S-box S_0 and second 4-bits (2nd row) fed into S-box S_1

The two boxes are defined as

S_0 =

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

S_1 =

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

Operation

1st & 4th i/p bits are treated as the row of the S-box → specified by the 2-bit number.

$S_0 \rightarrow$ (1st & 4th bit) eg:- $P_{0.0}, P_{0.3} = (0\ 0) = 0$ → base 2

2nd & 3rd i/p bits are treated as the column of the S-box → specified by the 2-bit number.

(2nd & 3rd bit) → S_1

eg:- $P_{0.1}, P_{0.2} = (10) = 2 \rightarrow$ base 2

o/p → row 0, colum 2 of $S_0 = 3 \rightarrow 11$ in binary

Similarly operation for S1 also.

Step 5

The 4 bits produced by S_0 & S_1 undergo a further permutation.

P_4			
2	4	3	1

- The output of p4 is the output of function F
- Switch function
  - interchanges left & right 4 bits so that the second instance of fk operates on different 4 bits

Example:

P.T : 1 0 1 1 1 1 0 1

Sub key 1 : 1 0 1 0 0 1 0 0

Sub key 2 : 0 1 0 0 0 0 1 1

Encryption.

Step 1: Initial Permutation

IP							
2	6	3	1	4	8	5	7

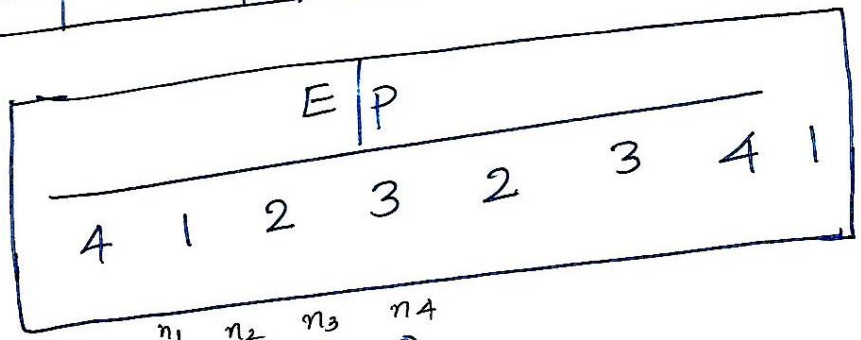
1	2	3	4	5	6	7	8
1	0	1	1	1	1	0	1

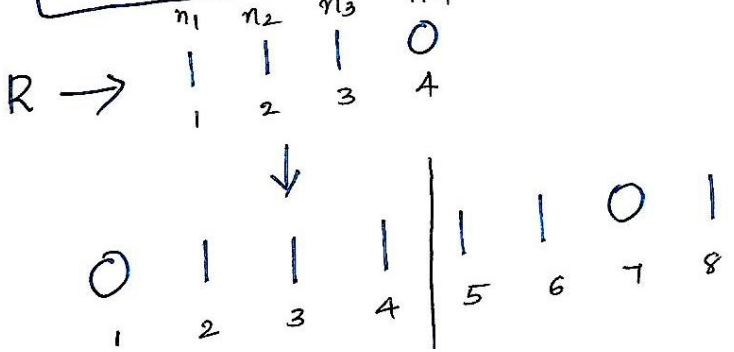
0 1 1 1 / 1 1 1 0

R.

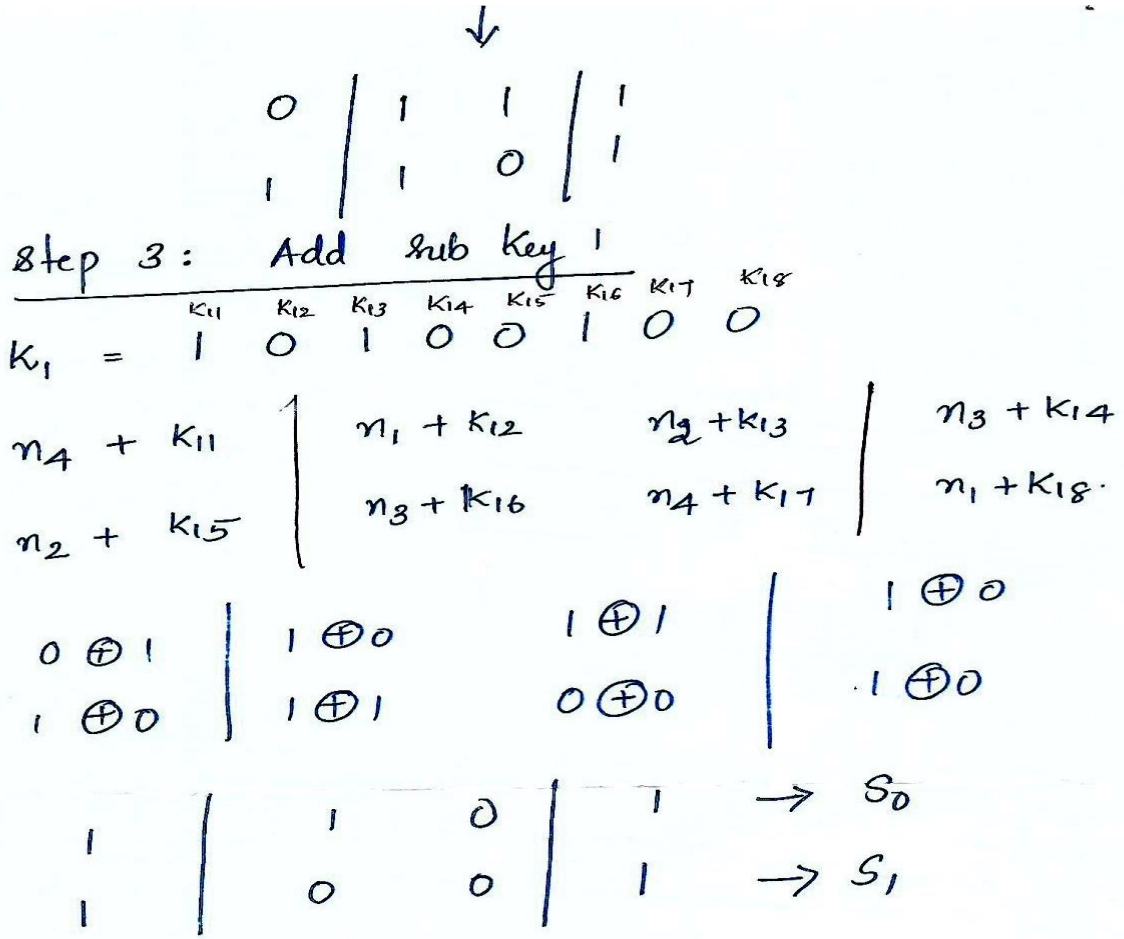
Step 2: Function f_K mapping $f(R, SK), K$

1) Expansion | permutation





n_4	n_1	n_2	n_3
n_2	n_3	n_4	n_1

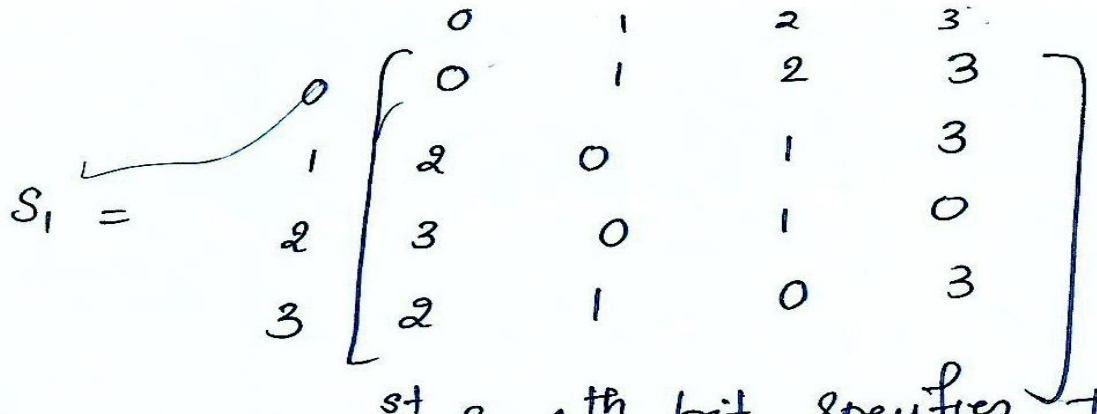

0 | 1 1 | 1
1 | 1 0 | 1
Step 3: Add sub key 1
K11 K12 K13 K14 K15 K16 K17 K18
K1 = 1 0 1 0 0 1 0 0
n4 + K11 | n1 + K12 n2 + K13 | n3 + K14
n2 + K15 | n3 + K16 n4 + K17 | n1 + K18.
0 ⊕ 1 | 1 ⊕ 0 1 ⊕ 1 | 1 ⊕ 0
1 ⊕ 0 | 1 ⊕ 1 0 ⊕ 0 | 1 ⊕ 0
1 | 1 0 | 1 → S0
1 | 0 0 | 1 → S1

$P_{0,0}$	$P_{0,1}$	P_{02}	P_{03}
P_{10}	P_{11}	P_{12}	P_{13}

Step 4 : S-box operation

$S_0 =$

	0	1	2	3
00	0	1	0	3	2
01	1	3	2	1	0
02	2	0	2	1	3
03	3	3	1	3	2



00 → 0
01 → 1
10 → 2
11 → 3

S_0 row → 1st & 4th bit specifies the row

1	2	3	4	
1	1	0	1	→ S_0
1	0	0	1	→ S_1
1	2	3	4	

S_0 row → 11 → 3

S_0 col → 2nd & 3rd bit specifies the column.

S_0 col → 10 → 2

S_1 row → 11 → 3

S_0 col → 00 → 0

o/p of S_0 box.

row-3, col-2 → 3 → 11

o/p of S_1 box.

row-3, col-0 → 2 → 10

o/p of S box → 1110

Step 5 P_4 permutation.

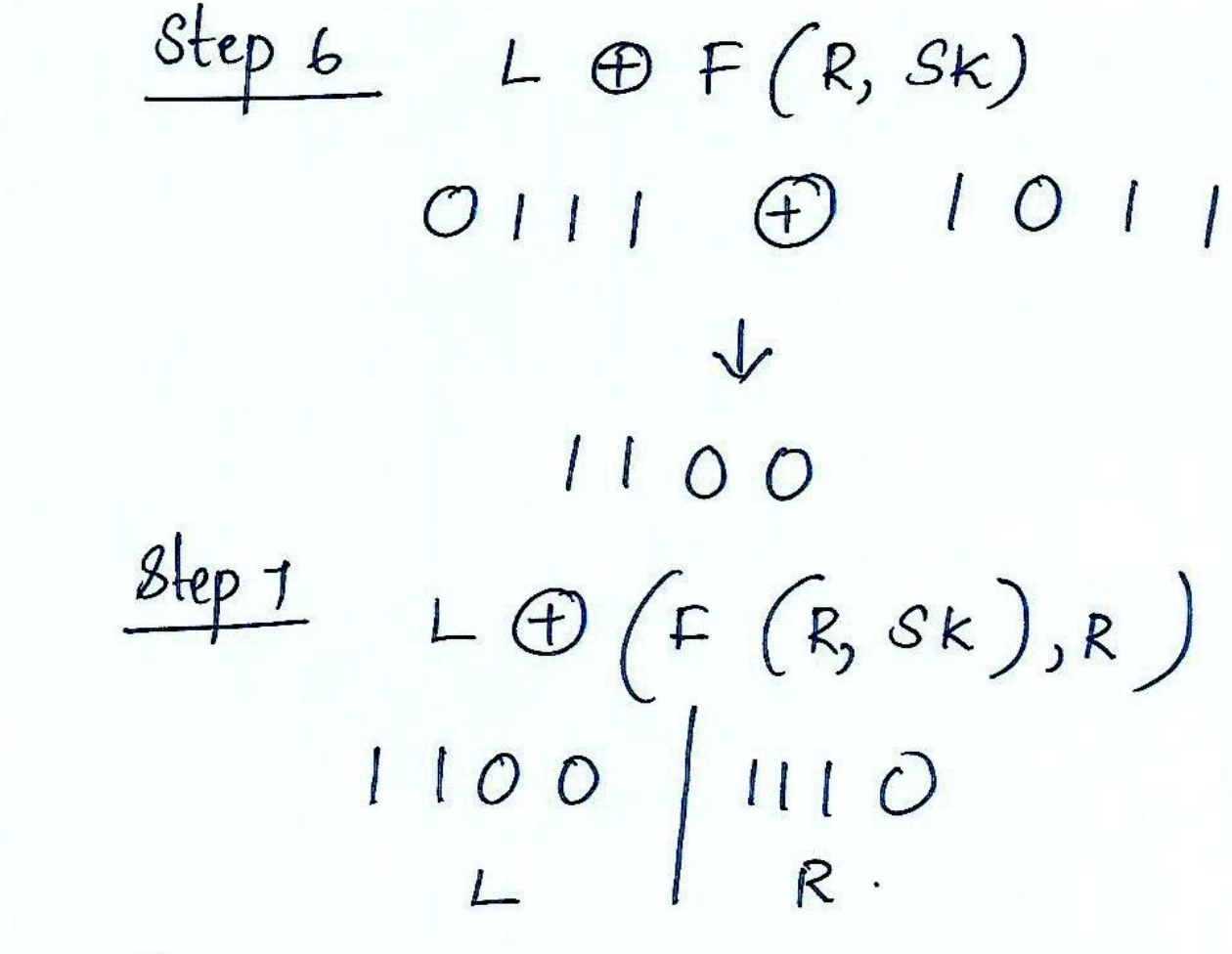
1	2	3	4
1	1	1	0

↓

1	0	1	1

P_4

2	4	3	1


Step 6
L ⊕ F(R, Sk)
0111 ⊕ 1011
1100
Step 7
L ⊕ (F(R, Sk), R)
1100 | 1110
L
R.

Step 8

Switch function SW

1100 1110

↓

1110 1100

Step 9 E|P.

E	P
4 1 2 3	2 3 4 1

1 2 3 4

R → 1 1 0 0

0 1 1 0, 1 0 0 1

0	1 1	0
1	0 0	1

n_4	n_1	n_2	n_3
n_2	n_3	n_4	n_1

Step 10 .

Add subkey 2 .

$K_{11}\ K_{12}\ K_{13}\ K_{14}\ K_{15}\ K_{16}\ K_{17}\ K_{18}$

$K_2 = 0\ 1\ 0\ 0\ 0\ 0\ 1\ 1$

$n_4 + k_{11}$	$n_1 + k_{12}$	$n_2 + k_{13}$	$n_3 + k_{14}$
$n_2 + k_{15}$	$n_3 + k_{16}$	$n_4 + k_{17}$	$n_1 + k_{18}$
$0 \oplus 0$	$1 \oplus 1$	$1 \oplus 0$	$0 \oplus 0$
$1 \oplus 0$	$0 \oplus 0$	$0 \oplus 1$	$1 \oplus 1$
1	2	3	4
$= 0$	0	1	0 $\rightarrow S_0$
1	0	1	0 $\rightarrow S_1$
1	2	3	4

$S_0, (1, 4) \rightarrow 0\ 0 \rightarrow 0$

row.

S_0 column $\rightarrow 0\ 1 \rightarrow 1$

S_0 output $\rightarrow 0 \rightarrow 00$

S_1 row $\rightarrow 10 \rightarrow 2$

S_1 col $\rightarrow 01 \rightarrow 1$

S_1 o/p $\rightarrow 0 \rightarrow 00$.

o/p of S box $\rightarrow \overset{1}{0}\overset{2}{0}\overset{3}{0}\overset{4}{0}$

Step 12 P_4 permutation.

P_4
2 4 3 1

0 0 0 0

↓

0 0 0 0.

Step 13

$L \oplus F(R, S, K)$

```
  1 1 1 0   ⊕
  0 0 0 0
-------------
  1 1 1 0
```

Step 14 $F(R, SK), R$.

1110 1100

Step 15 Inverse permutation, IP^{-1}

```
1 2 3 4 5 6 7 8
1 1 1 0 1 1 0 0
```

IP^{-1}							
4	1	3	5	7	2	8	6

↓

0 1 1 1 0 1 0 1

C.T → 01110101

Analysis of S-DES

- Brute-force attack on S-DES is certainly feasible
- With a 10 bit key, there are only 2^{10} possiblities

Relationship to DES

- DES operates on 64 bit blocks of input

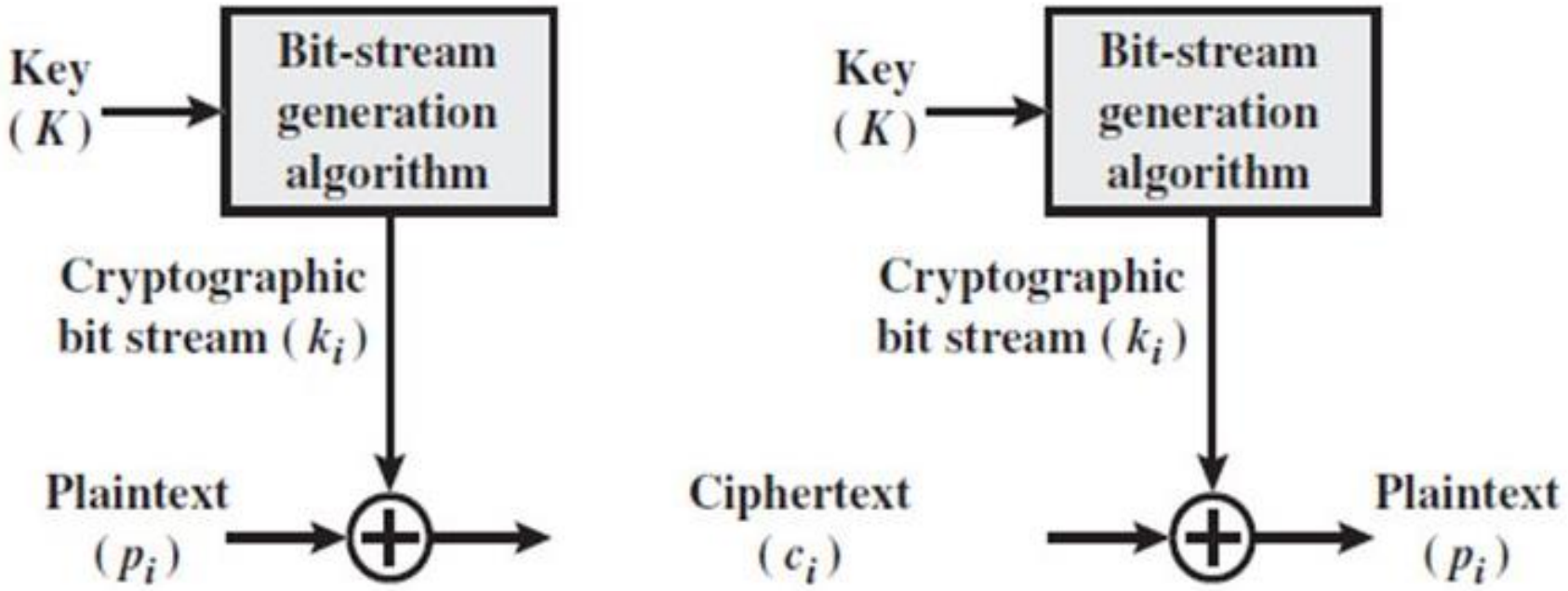
BLOCK CIPHER DESIGN PRINCIPLES

Stream Ciphers

- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
- **Examples: the auto keyed Vigenère cipher and the Vernam cipher.**

Bit-stream generator

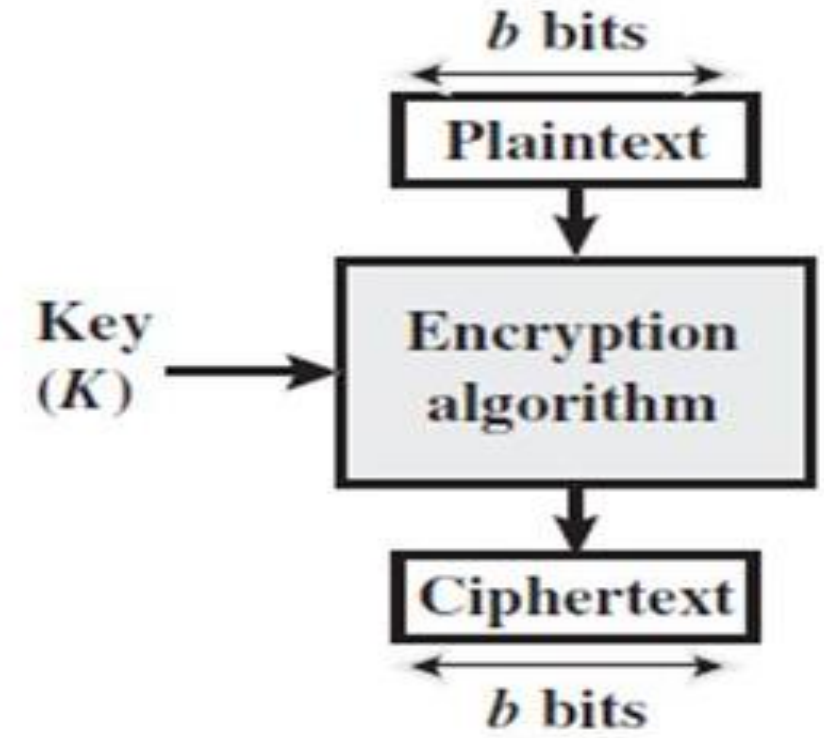
- is a key-controlled algorithm
- must produce a bit stream that is cryptographically strong.
- That is, it must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream.
- The two users need only share the generating key, and each can produce the key stream.



(a) Stream cipher using algorithmic bit-stream generator

Block cipher

- a block of plaintext is treated as a whole and used to produce a cipher text block of equal length
- A block size of 64 or 128 bits is used.
- the two users share a symmetric encryption key.
- Block ciphers are applicable to a broader range of applications than stream ciphers.



(b) Block cipher

The three critical aspects of block cipher design are:

- **The number of rounds**
- **Design of the function f**
- **Key scheduling**

Number of Rounds

- The greater the number of rounds, the more difficult it is to perform cryptanalysis,even for a relatively weak F.
- the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack.

Design of Function F(1)

- The more nonlinear F, the more difficult any type of cryptanalysis will be.
- good avalanche properties
- A change in one bit of the input should produce a change in many bits of the output.
- **strict avalanche criterion (SAC),which** states that any output bit *j of an S-box should change with probability 1/2 when* any single input bit *i is inverted for all i, j.*

Design of Function F(2)

- Another criterion is the **bit independence criterion(BIC),**
- **which states that** output bits *j and k should change independently when any single input bit i is* inverted for all *i, j, and k.*

Key Schedule Algorithm

- Select subkeys to maximize the difficulty of deducing individual sub keys and the difficulty of working back to the main key.
- At minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion.

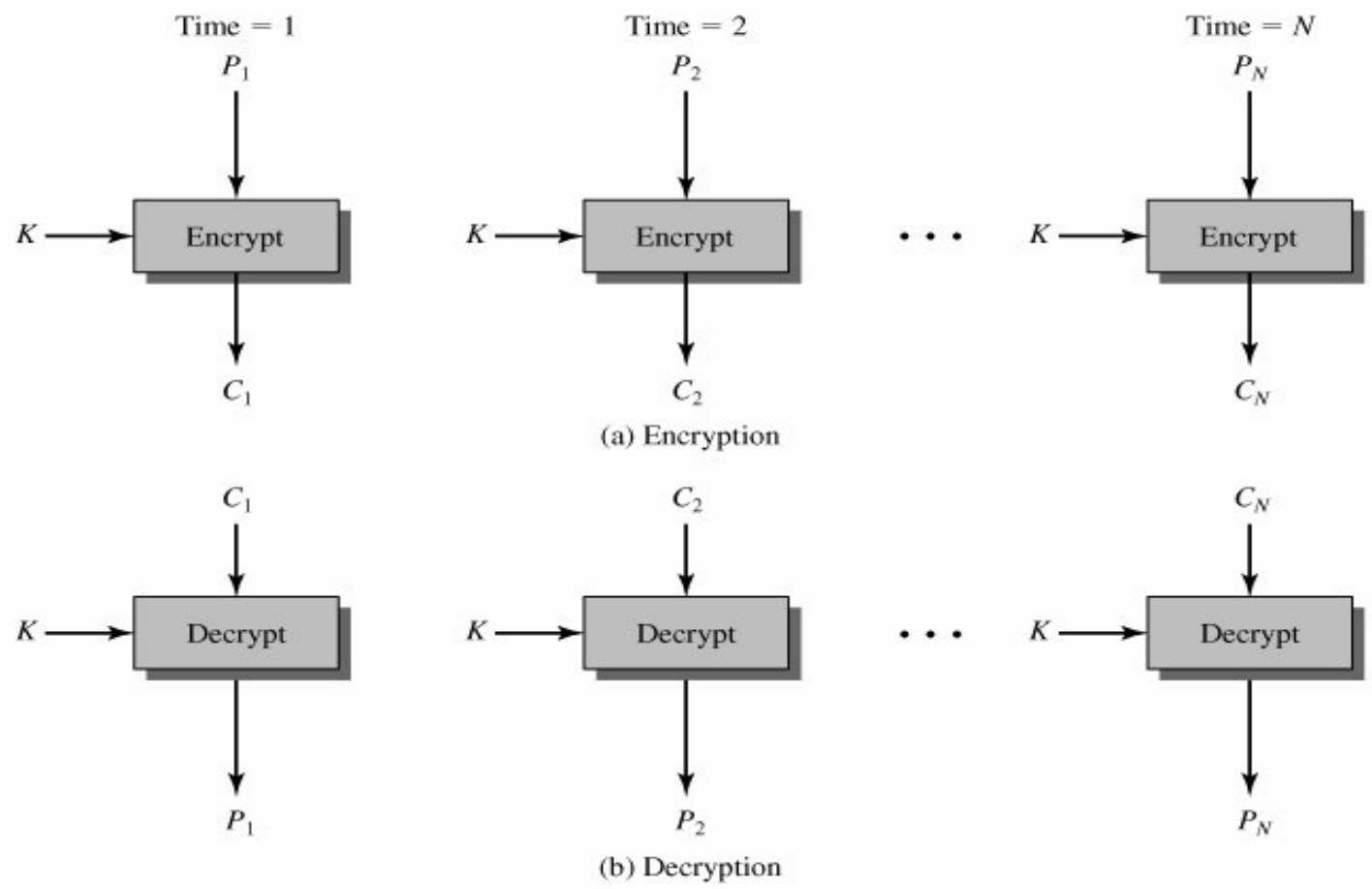
BLOCK CIPHER MODES OF OPERATIONS

- Electronic Codebook Mode
- Cipher Block Chaining Mode
- Cipher Feedback Mode
- Output Feedback Mode
- Counter Mode

Electronic Codebook Mode (ECB)

- simplest mode
- plaintext is handled one block at a time
- each block of plaintext is encrypted using the same key.
- The term codebook is used because, for a given key, there is a unique ciphertext for every b-bit block of plaintext.

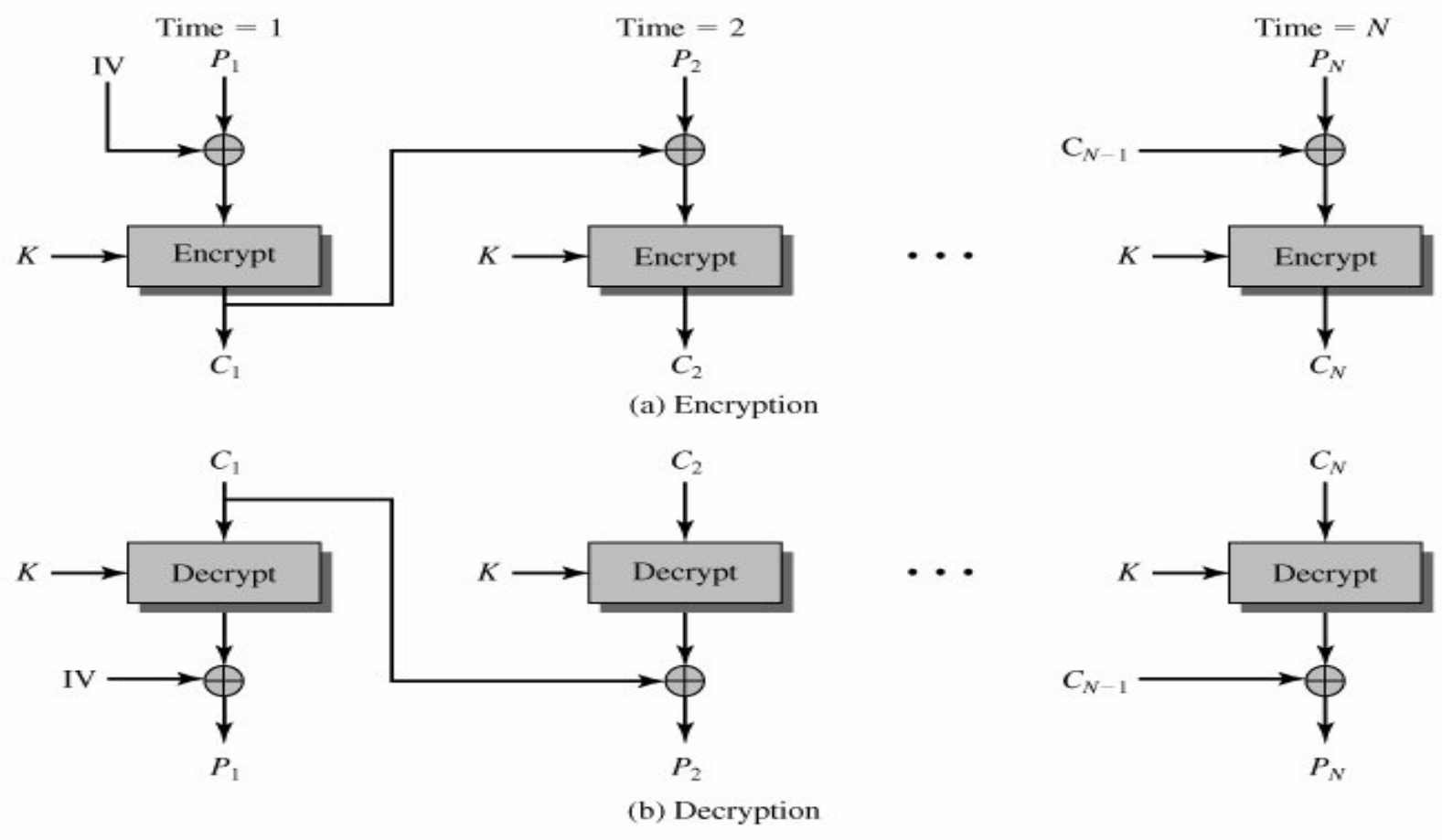
Electronic Codebook Mode



Cipher Block Chaining Mode

- input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block;
- the same key is used for each block.
- The input to the encryption function for each plaintext block bears no fixed relationship to the plaintext block

Cipher Block Chaining Mode



$$C_i = E_k[C_{i-1} \oplus P_i]$$

$$D_K[C_i] = D_K[E_K(C_{i-1} \oplus P_i)]$$

$$D_K[C_i] = (C_{i-1} \oplus P_i)$$

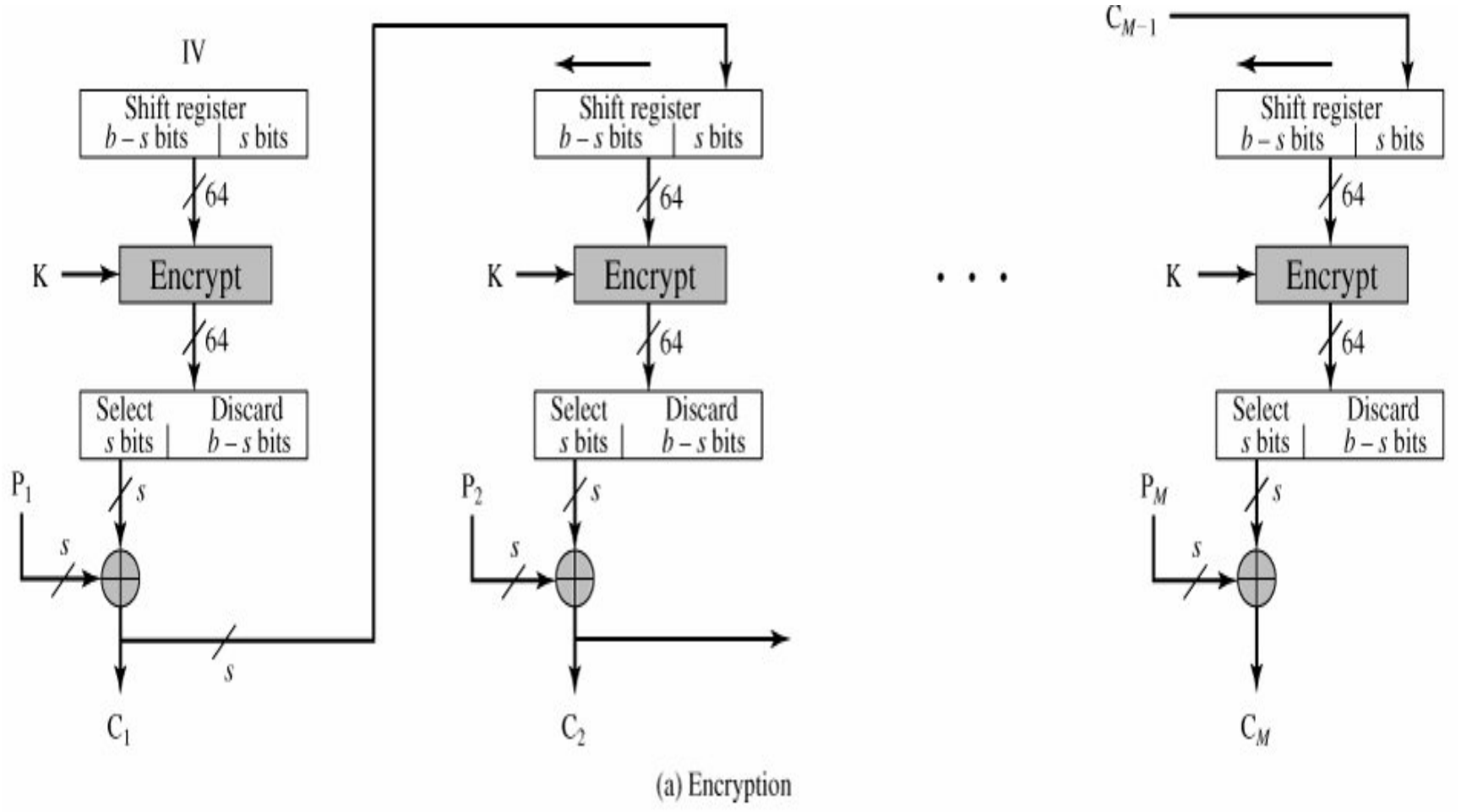
$$C_{i-1} \oplus D_K[C_i] = C_{i-1} \oplus C_{i-1} \oplus P_i = P_i$$

Henric Johnson

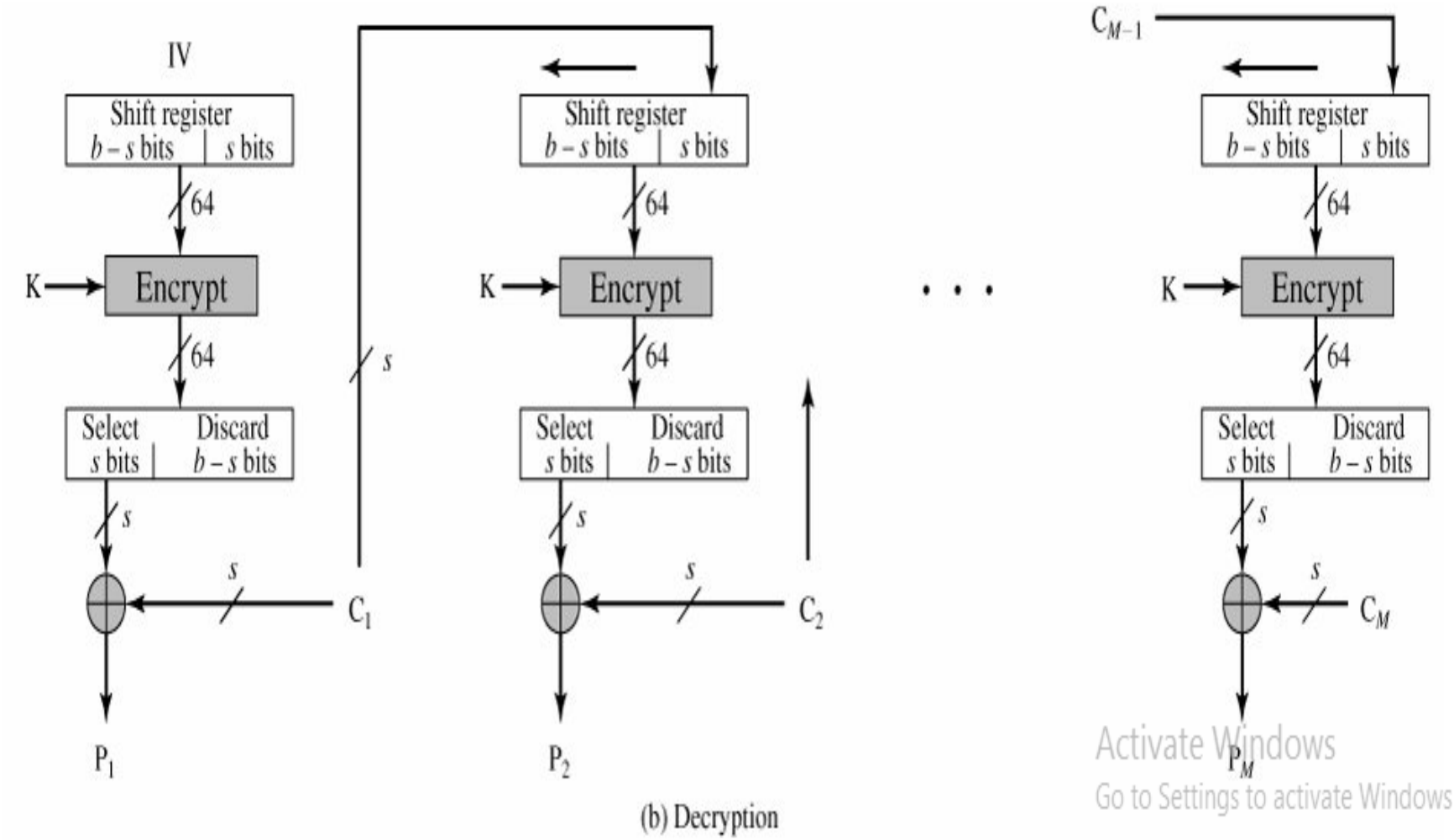
- To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext.
- On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plainThe IV must be known to both the sender and receiver but be
- unpredictable by a third party.text.
- The IV is a data block that is the same size as the cipher block.

- For maximum security, the IV should be protected against unauthorized changes.
- This could be done by sending the IV using ECB encryption.

3) Cipher Feedback Mode



(a) Encryption

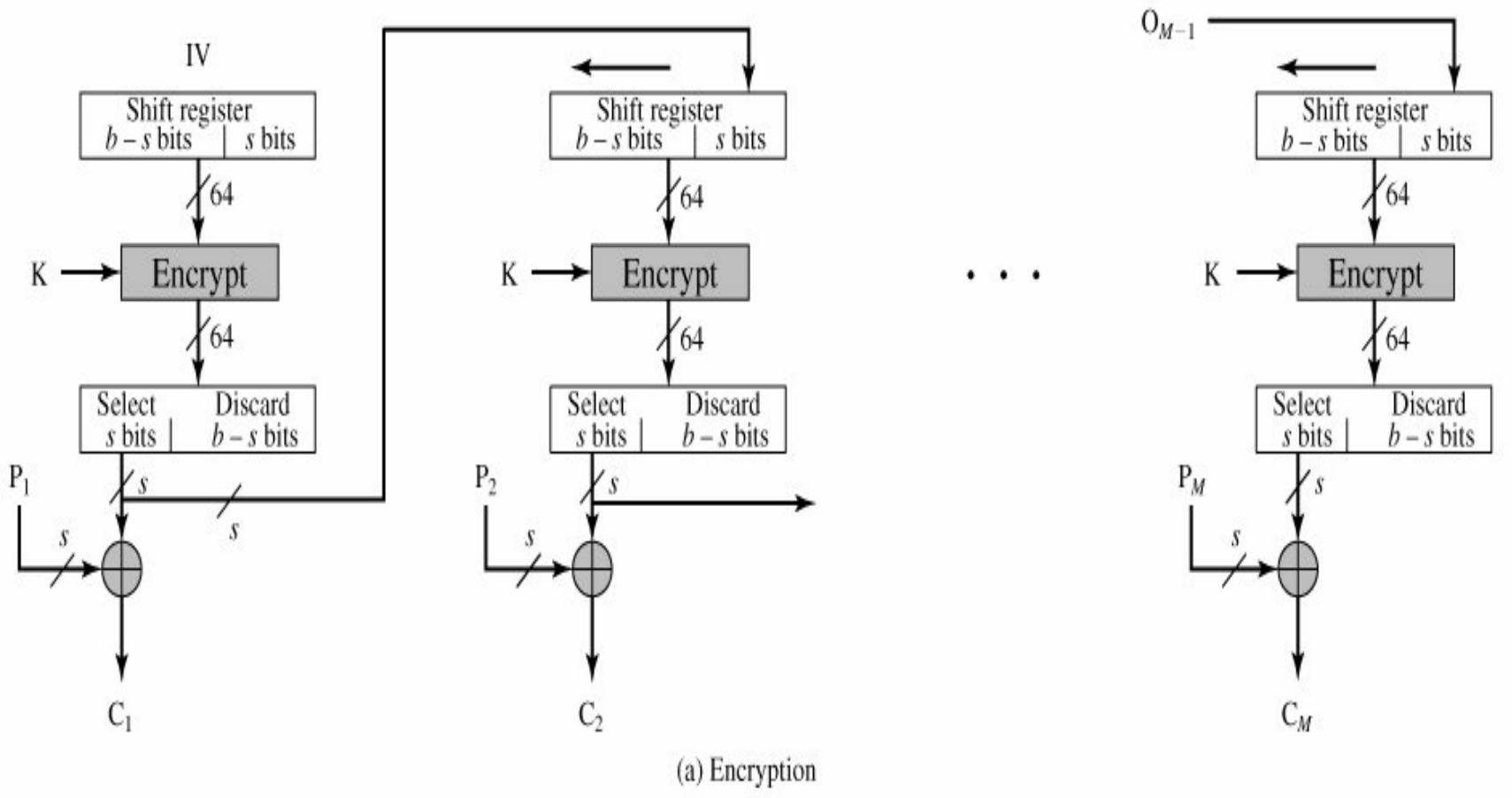


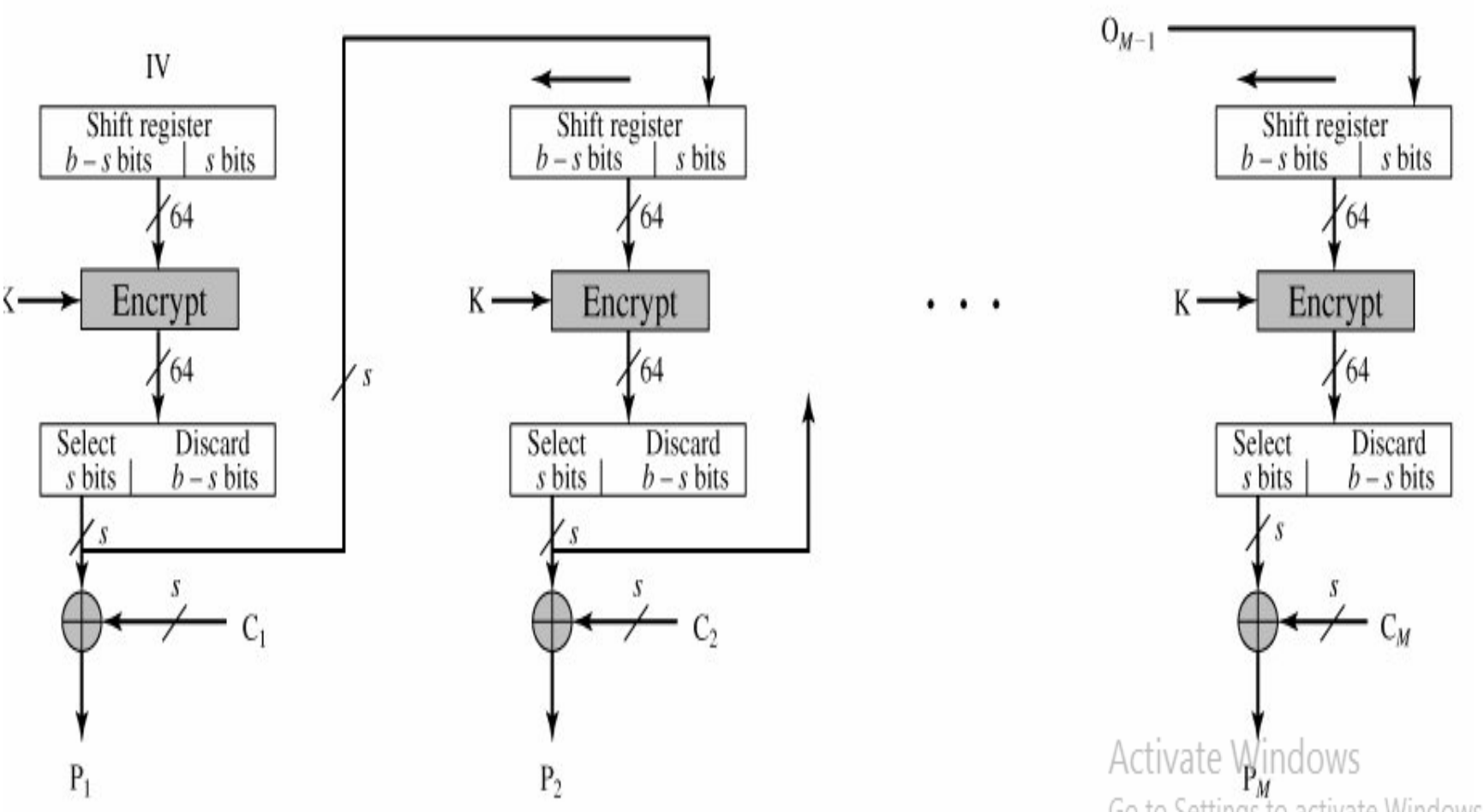
(b) Decryption

3) Cipher Feedback Mode

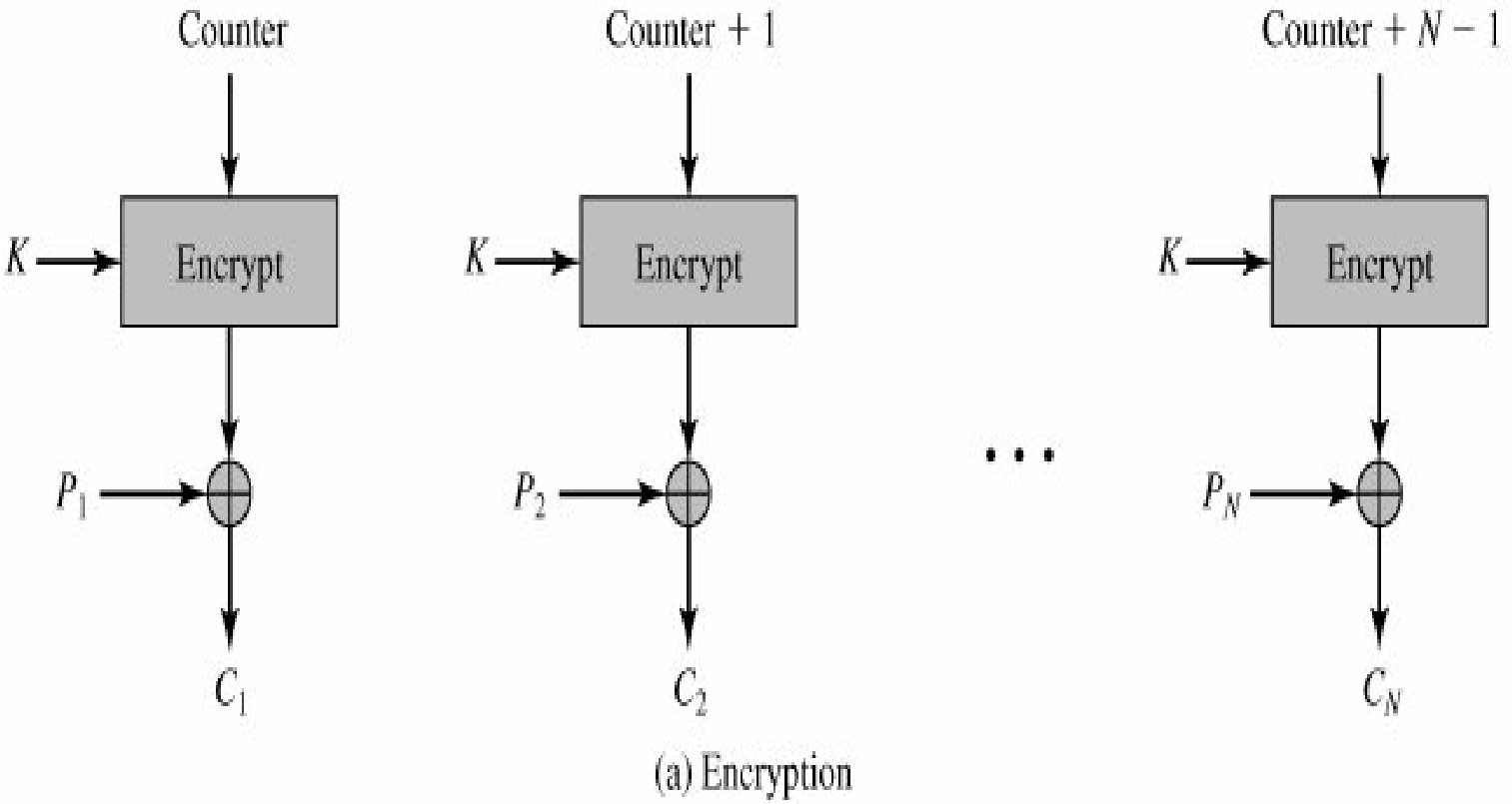
- it is assumed that the unit of transmission is s bits; a common value is s = 8.
- the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext.
- In this case, rather than units of b bits, the plaintext is divided into segments of s bits.

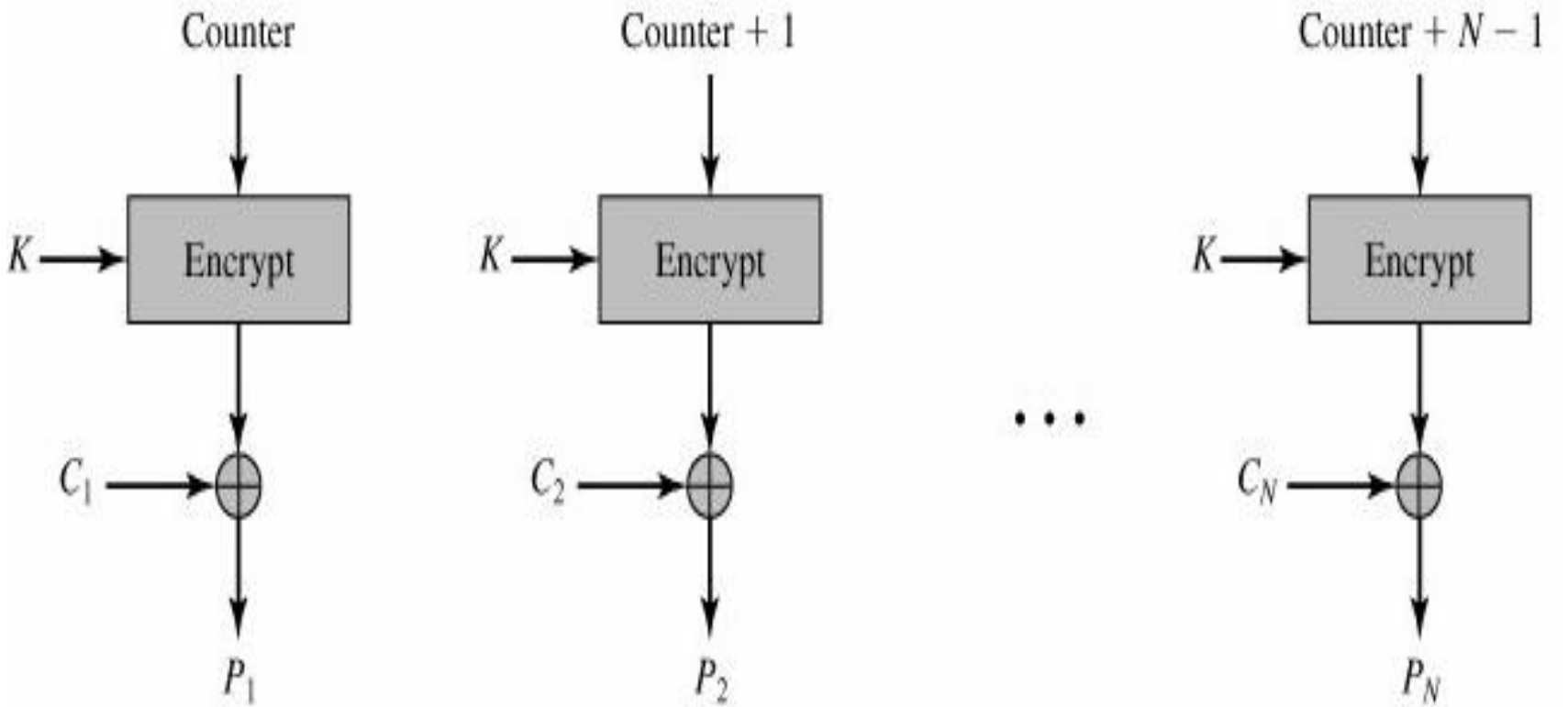
4)Output Feedback Mode





5)Counter Mode





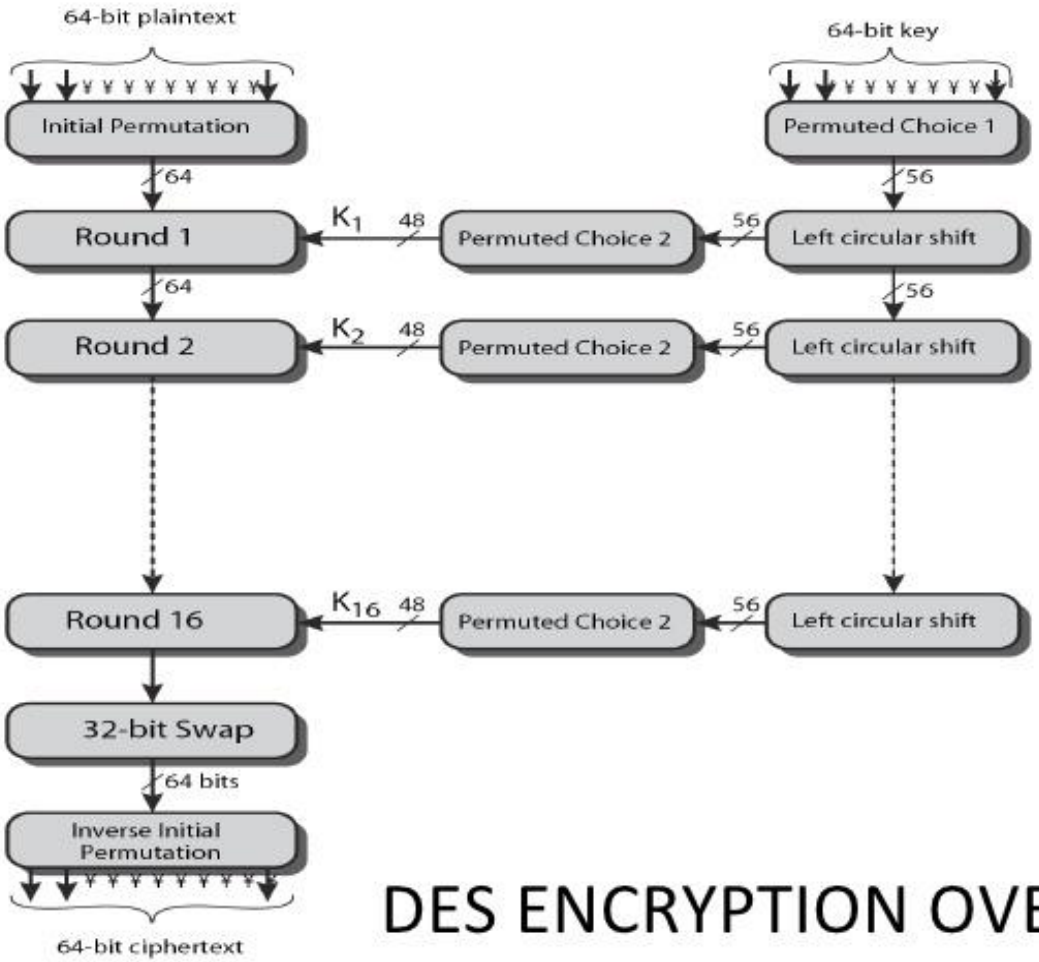
(b) Decryption

5)Counter Mode

- For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining.
- For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block.

THE DATA ENCRYPTION STANDARD

- data are encrypted in 64-bit blocks
- 56-bit key.
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output.
- The same steps, with the same key, are used to reverse the encryption.



DES ENCRYPTION OVERVIEW

Processing of Plaintext

- At the left-hand side of the figure, the processing of the plaintext proceeds in **three phases.**
  - ❖ First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input**.**
  - ❖ This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.
  - ❖ The output of the last (sixteenth)round consists of 64 bits that are a function of the input plaintext and the key.
  - ❖ The left and right halves of the output are swapped to produce the preoutput.
  - ❖ Finally, the preoutput is passed through the inverse of the initial permutation function, to produce the 64-bit C.T

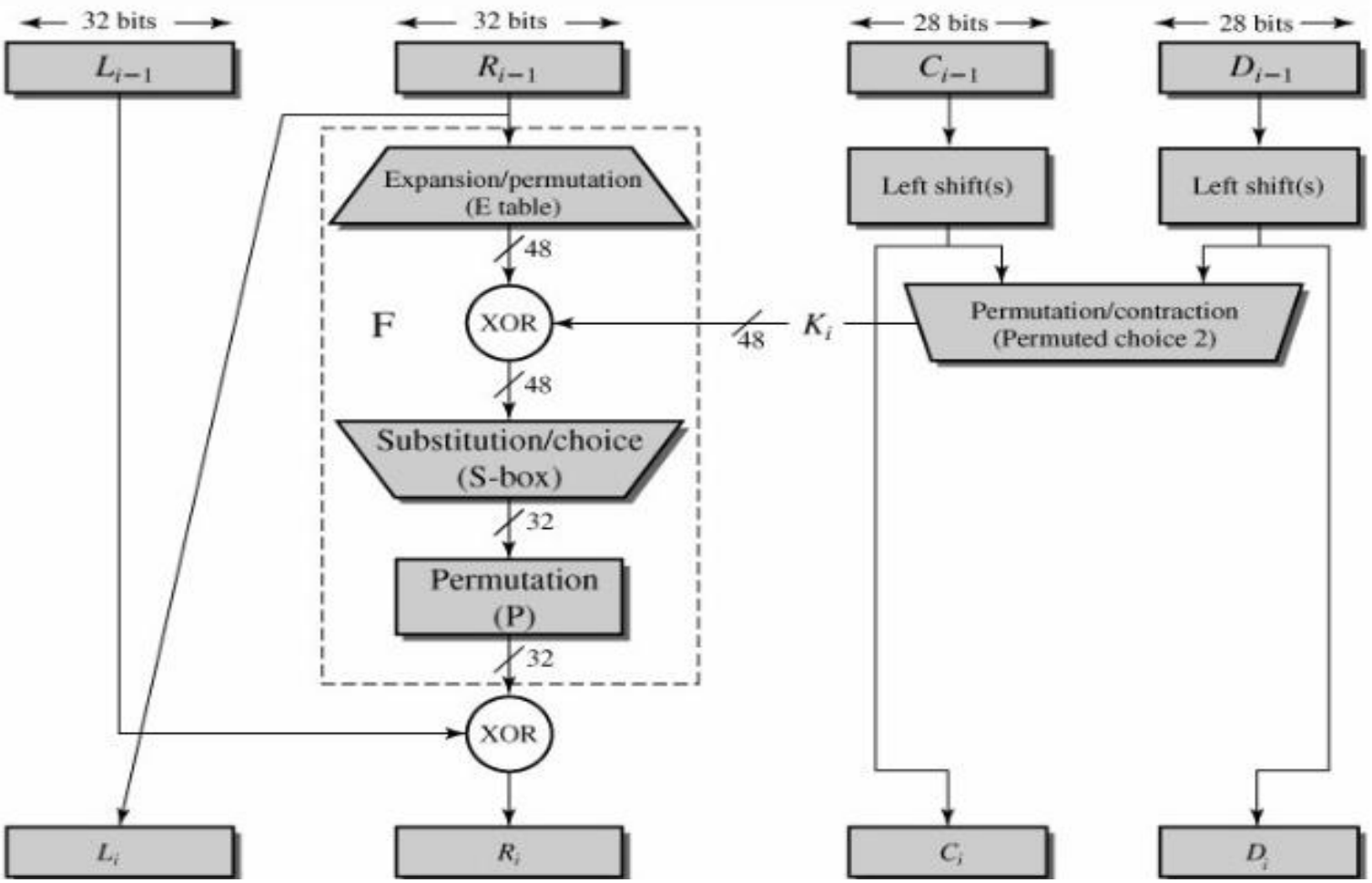
(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (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

Details of Single Round

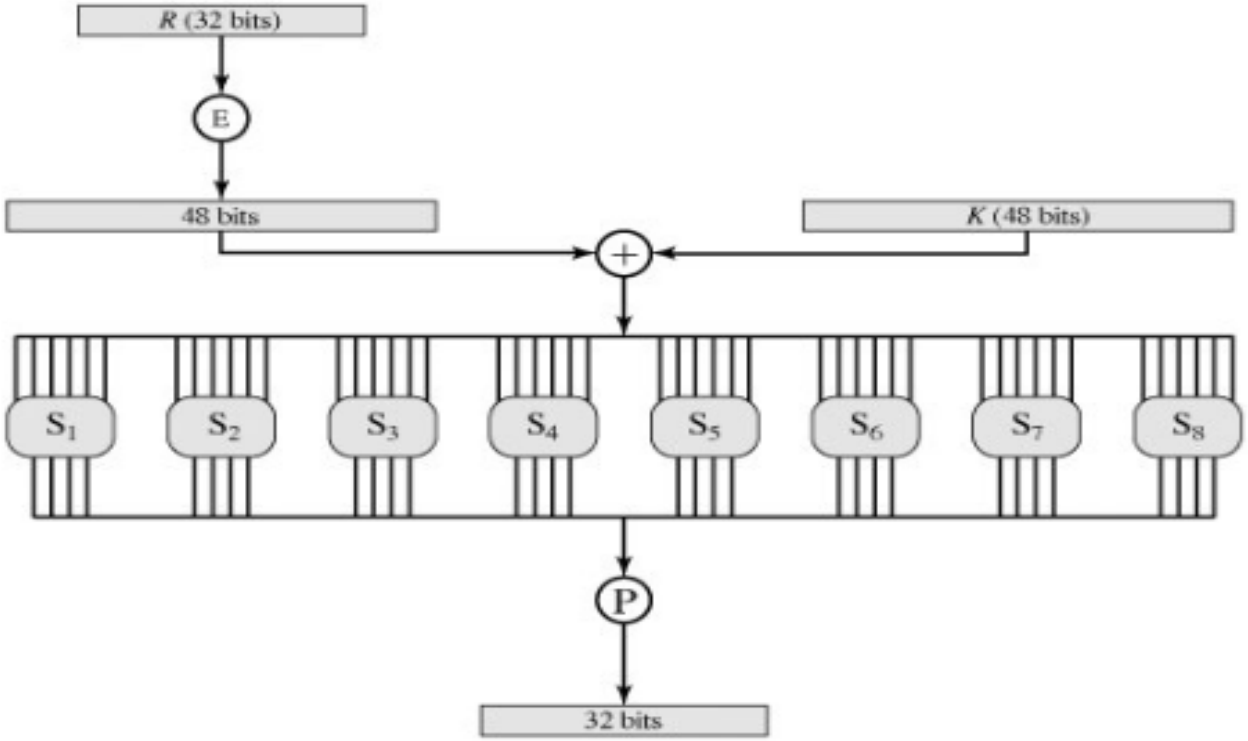


The overall processing at each round can be summarized in the following formulas:

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

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

The role of the S-boxes in the function F is illustrated in Figure below



The Avalanche Effect

- A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext.
- In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.

- For example, in S1 for input 011001, the row is 01 (row 1) and the column is 1100 (column 12). The value in row 1, column 12 is 9, so the output is 1001.

THE STRENGTH OF DES

- Use of 56-Bit Keys
- brute-force attack appears impractical.
- With current technology, it is not even necessary to use special purpose built in hardware.
- Rather, the speed of commercial processors threatens the security of DES.

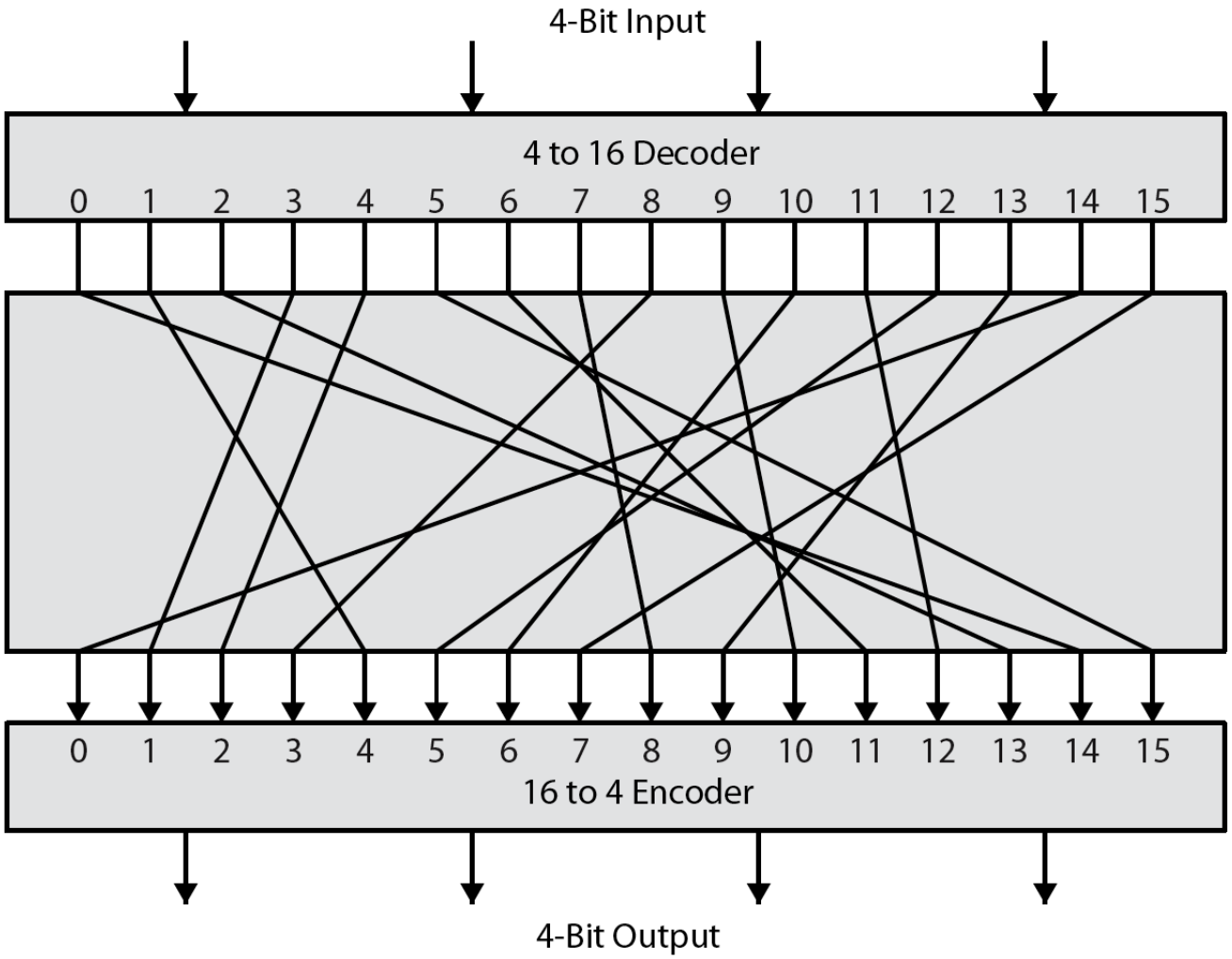
The time required fora brute-force attack for various key sizes:

- A single PC can break DES in about a year.
- If multiple PCs work in parallel, the time is drastically shortened.
- Today's super computers should be able to find a key in about an hour.
- Key sizes of 128 bits or greater are effectively unbreakable using simply a brute force approach.

Block Cipher Principles

- most symmetric block ciphers are based on a Feistel Cipher Structure
- needed since must be able to decrypt ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher

Ideal Block Cipher



Diffusion and Confusion

Diffusion

- To make the statistical relationship between the **plaintext** and **cipher text** as complex as possible in order to thwart attempts to discover the key.
- Can be achieved by a **Permutation** followed by a function

Confusion

- To make the relationship between the statistics of the **cipher text** and the value of the **encryption key** as complex as possible to thwart attempts to discover the key.
- Can be achieved by a **Substitution**.