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.

In daily life we use information for various purposes and use network for communication and exchange information between different parties. In many cases these information are sensitive so we need to take care that only authorized party can get that information. For maintaining such privacy we require some mechanism or physical device which ensures that it is safe. Such mechanism or physical devices are known as **security system**.

Computer Security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the **integrity**, **availability**, and **confidentiality** of information system resources.

Security Services

1. Authentication - assurance that the communicating entity is the one claimed
2. Access Control - prevention of the unauthorized use of a resource
3. Data Confidentiality –protection of data from unauthorized disclosure
4. Data Integrity - assurance that data received is as sent by an authorized entity
5. Non-Repudiation - protection against denial by one of the parties in a communication

Classify Security Attacks

Passive attacks - eavesdropping on, or monitoring of, transmissions to:

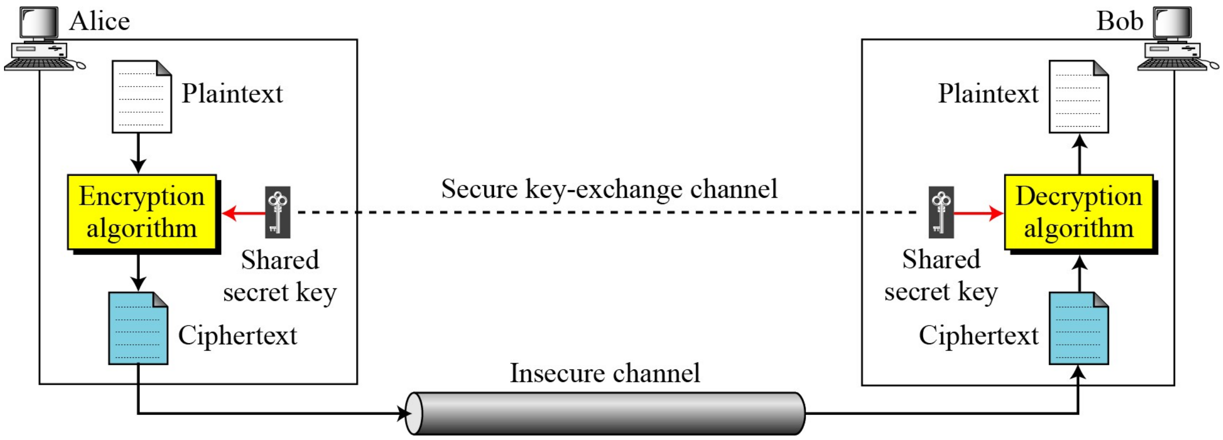
- obtain message contents, or
- monitor traffic flows

Active attacks – modification of data stream to:

- masquerade of one entity as some other
- replay previous messages
- modify messages in transit

- denial of service

Symmetric Cipher Model



If P is the plaintext, C is the cipher text, and K is the key,

$$\text{Encryption: } C = E_k(P) \qquad \text{Decryption: } P = D_k(C)$$

$$\text{In which, } D_k(E_k(x)) = E_k(D_k(x)) = x$$

We assume that Bob creates P_1; *we prove that* $P_1 = P$*:*

$$\text{Alice: } C = E_k(P)$$

Substitution Techniques

- Various conventional encryption schemes or substitution techniques are asunder:
  Symmetric Cipher Models

SUBSTITUTION CIPHERS

A substitution cipher replaces one symbol with another. Substitution ciphers can be categorized as either mono alphabetic ciphers or poly alphabetic ciphers.

A substitution cipher replaces one symbol with another.

Monoalphabetic Substitution Cipher

- Instead of shifting alphabets by fixed amount as in Caesar cipher, any random permutation is assigned to the alphabets. This type of encryption is called monoalphabetic substitution cipher.
- For example, A is replaced by Q, B by D, C by T etc. then it will be comparatively stronger than Caesar cipher.
- The number of alternative keys possible now becomes26!.
- Thus, Brute Force attack is impractical in this case.

In mono alphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the cipher text is always one-to-one.

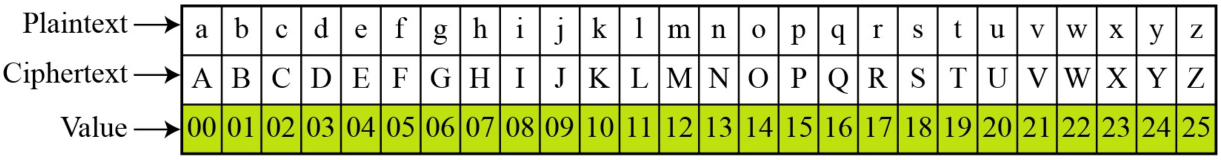
Example 1: In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the cipher text is always one-to-one.

Plaintext: hello **Ciphertext:** KHOOR

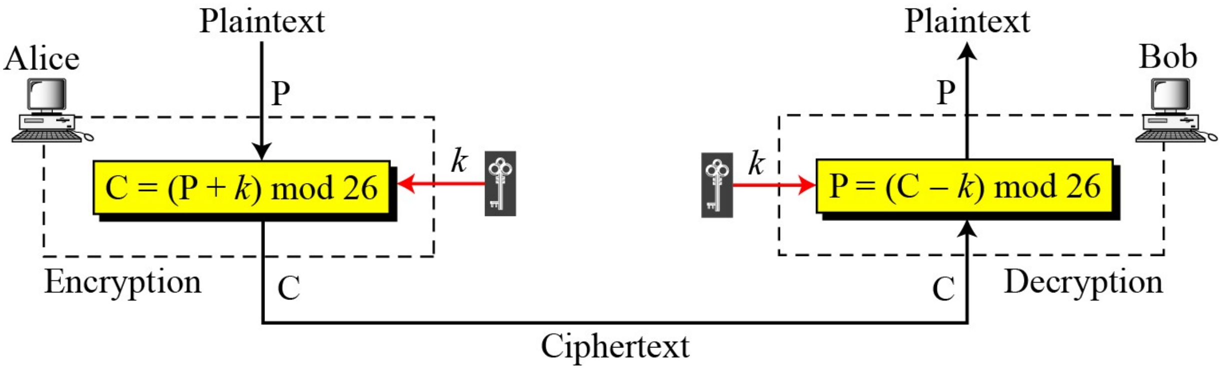
Monoalphabetic Substitution Cipher are:

1. Additive Cipher

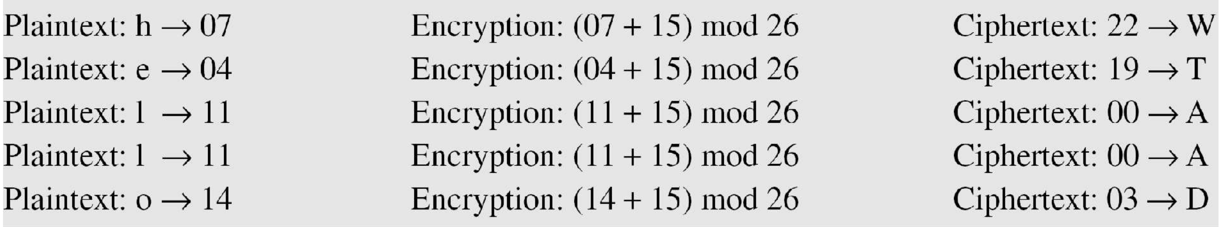
The simplest monoalphabetic cipher is the additive cipher. This cipher is sometimes called a shift cipher and sometimes a Caesar cipher, but the term additive cipher better reveals its mathematical nature.



Plaintext →	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
Ciphertext →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



Eg: Use the additive cipher with key = 15 to encrypt the message “hello”.



Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

2. Shift Cipher and Caesar Cipher

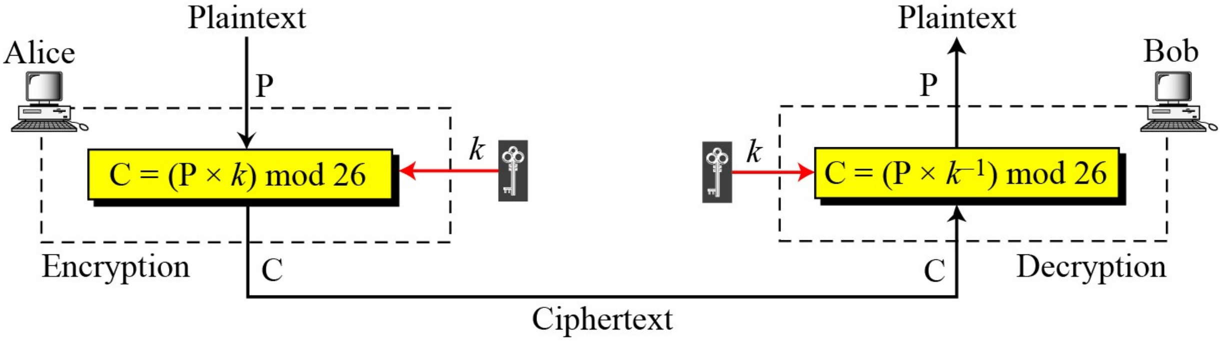
- Additive ciphers are sometimes referred to as shift ciphers or Caesar cipher.
- The encryption rule is simple; replace each letter of the alphabet with the letter standing 3 places further down the alphabet.
- Mathematically, starting from a=0, b=1 and so on, Caesar cipher can be writtenas:

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

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

- This cipher can be broken
  - If we know one plaintext-cipher text pair since the difference will be same.
  - By applying Brute Force attack as there are only 26 possible keys.

3. Multiplicative Ciphers

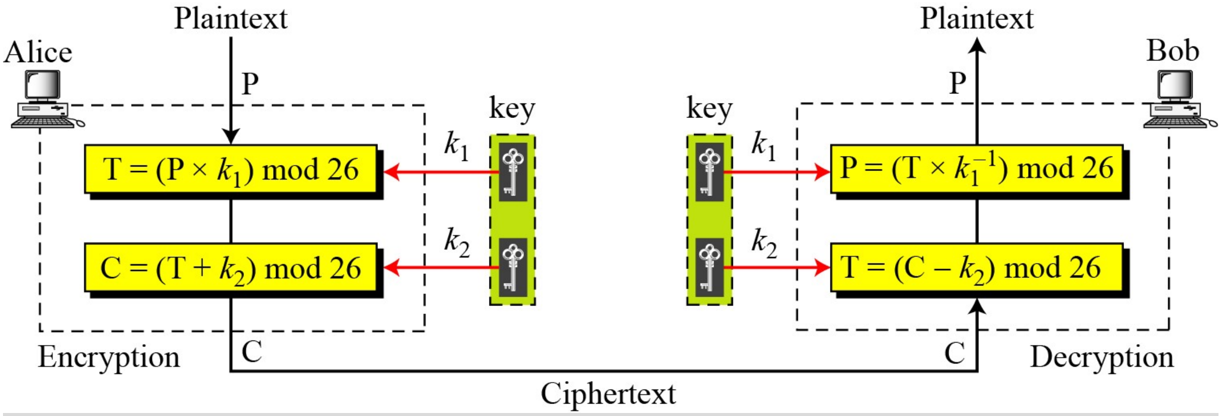


In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z_{26}^*.

Eg: We use a multiplicative cipher to encrypt the message “hello” with a key of 7. The cipher text is “XCZZU”.

Plaintext: h → 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 → X
Plaintext: e → 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 → C
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: o → 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 → U

4. Affine Ciphers



$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Eg: Use an affine cipher to encrypt the message “hello” with the key pair (7, 2).

P: h → 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 → W

Polyalphabetic Ciphers

- In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the cipher text is one-to-many.

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3\ldots \qquad k = (k_1, P_1, P_2, \ldots)$$

- $$\text{Encryption: } C_i = (P_i + k_i) \bmod 26 \qquad \text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

Eg: Assume that Alice and Bob agreed to use an autokey cipher with initial key value k_1 = 12. Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P’s Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C’s Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

1. Playfair Cipher

- In this technique multiple (2) letters are encrypted at atime.
- This technique uses a 5 X 5 matrix which is also called keymatrix.

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**

- The plaintext is encrypted **two letters at atime**:
  1. Break the plaintext into pairs of two consecutive letters.
  2. If a pair is a repeated letter, insert a filler like ‘X‘in the plaintext, eg. "Balloon" is treated as "ba lx lo on".
  3. If both letters fall in the same row of the key matrix, replace each with the letter to its right (wrapping back to start from end), eg. “AR" encrypts as "RM".
  4. If both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom), eg. “MU" encrypts to "CM".
  5. Otherwise each letter is replaced by the one in its row in the column of the other letter of the pair, eg. “HS" encrypts to "BP", and “EA" to "IM" or "JM" (as desired)

2. Hill Cipher

- This cipher is based on linear algebra.

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$

$$\begin{aligned} C_1 &= P_1\,k_{11} + P_2\,k_{21} + \cdots + P_m\,k_{m1} \\ C_2 &= P_1\,k_{12} + P_2\,k_{22} + \cdots + P_m\,k_{m2} \\ &\cdots \\ C_m &= P_1\,k_{1m} + P_2\,k_{2m} + \cdots + P_m\,k_{mm} \end{aligned}$$

- The key matrix in the Hill cipher needs to have a multiplicative inverse.
- **For example,** the plaintext “code is ready” can make a 3 × 4 matrix when adding extra bogus character “z” to the last block and removing the spaces. The ciphertext is “OHKNIHGKLISS”.

$$\overset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} = \overset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} \overset{K}{\begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}}$$

a. Encryption

$$\overset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} = \overset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} \overset{K^{-1}}{\begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}}$$

b. Decryption

- The strength of the Hill cipher is that it completely hides single-letter frequencies.
- Although the Hill cipher is strong against a cipher text-only attack, it is easily broken with a known plaintext attack.
-

3. Vigenere Cipher

$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad K = [(k_1, k_2, \ldots, k_m), (k_1, k_2, \ldots, k_m), \ldots]$

$\text{Encryption: } C_i = P_i + k_i \qquad \text{Decryption: } P_i = C_i - k_i$

Eg. We can encrypt the message "She is listening" using the 6-character keyword "PASCAL".

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***
C's values:	**07**	**07**	**22**	**10**	**18**	**22**	**23**	**18**	**11**	**6**	**13**	**19**	**02**	**06**
Ciphertext:	**H**	**H**	**W**	**K**	**S**	**W**	**X**	**S**	**L**	**G**	**N**	**T**	**C**	**G**

Let us see how we can encrypt the message "She is listening" using the 6-character keyword "PASCAL". The initial key stream is (15, 0, 18, 2, 0, 11). The key stream is the repetition of this initial key stream (as many times as needed).

TRANSPOSITION CIPHERS

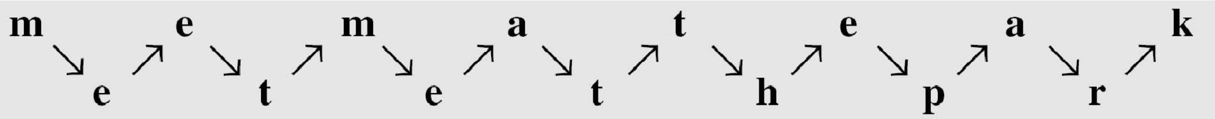
A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.

- Keyless Transposition Ciphers
- Keyed Transposition Ciphers
- Combining Two Approaches

Keyless Transposition Ciphers

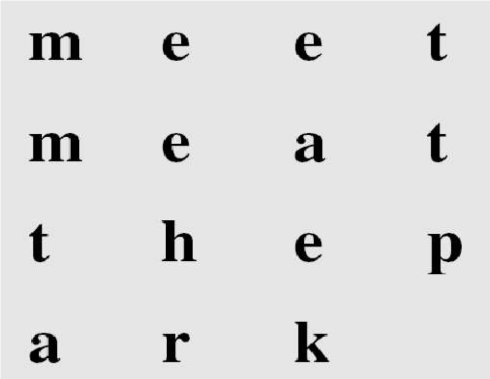
Simple transposition ciphers, which were used in the past, are keyless.

1. A good example of a keyless cipher using the first method is the rail fence cipher. The cipher text is created reading the pattern row by row. For example, to send the message "Meet me at the park" to Bob, Alice writes



She then creates the cipher text **"MEMATEAKETETHPR".**

2. Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.



m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

She then creates the cipher text **“MMTAEEHREAEKTTP”.**

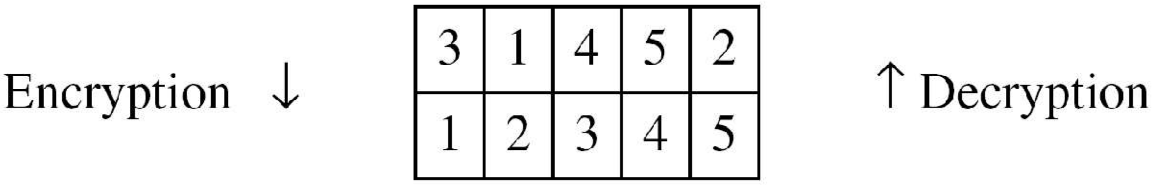
Keyed Transposition Ciphers

The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way The permutation is done on the whole plaintext to create the whole cipher text. Another method is to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.

Alice needs to send the message “Enemy attacks tonight” to Bob..

e	n	e	m	y	a	t	t	a	c	k	s	t	o	n	i	g	h	t	z

The key used for encryption and decryption is a permutation key, which shows how the character are permuted.



Encryption ↓

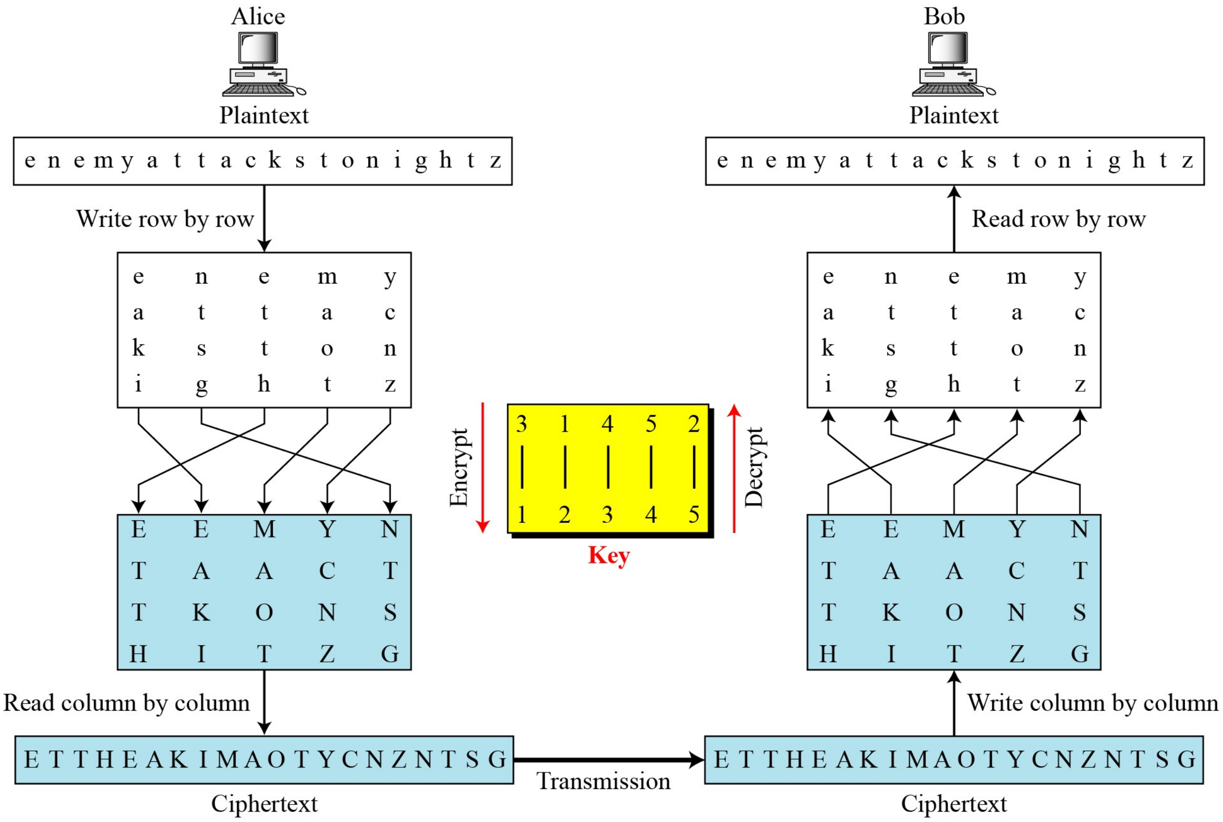
3	1	4	5	2
1	2	3	4	5

↑ Decryption

The permutation yields

E	**E**	**M**	**Y**	**N**	**T**	**A**	**A**	**C**	**T**	**T**	**K**	**O**	**N**	**S**	**H**	**I**	**T**	**Z**	**G**

Combining Two Approaches



Steganography

- Plaintext message may be hidden in one of two ways.
  1. Conceal the existence of the message-Steganography.
  2. Render the message unintelligible to outsiders by various transformations of the text- Cryptography
- A simple but time consuming form of steganography is the one in which an arrangement of words or letters within an apparently normal text spells out the real message.
- For example, the sequence of first letters of each word of the overall message spells out the hidden message.
- Some other techniques that have been used historically are listed below:
  - **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 alight.
- **Typewriter correction ribbon:** Used between lines typed with a ribbon the results of black typing with the correction tape are visible only under a strong light.
- Although these techniques may seem ancient, they have modern equivalents.
- For example, suppose an image has a resolution of 2048 X 3072 pixels where each pixel is denoted by 24 bits (Kodak CD photo format).
- The least significant bit of each 24-bit pixel can be changed without greatly affecting the quality of the image.
- The result is that you can hide a 2.3-megabyte message in a single digital snapshot.
- There are now a number of software packages available that take this type of approach to steganography.
- Steganography has a number of drawbacks when compared to encryption.
    a. It requires a lot of overhead to hide a relatively few bits of information.
    b. Once the system is discovered, it becomes virtually worthless.
- The advantage of steganography is that it can be employed by parties who have something to lose if the fact of their secret communication is discovered.

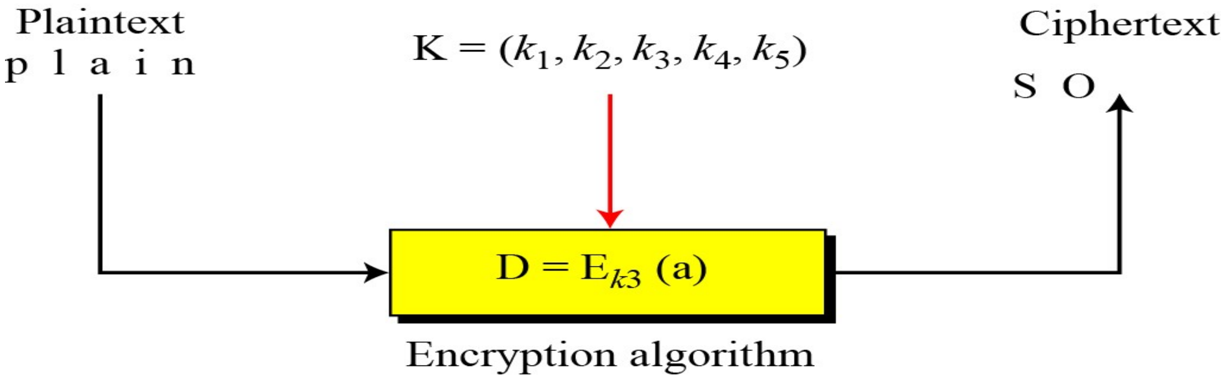
Rotor Machines

- The basic principle of the rotor machine is illustrated in figure. The machine consists of a set of independently rotating cylinders through which electrical pulse can flow.
- Each cylinder has 26 input and 26 output pins, with internal wiring that connect each input pin to unique output pin.
- If we associate each input and output pin with a letter of the alphabet, then a single cylinder defines a monoalphabetic sub stitution.
- If we use multiple cylinders then we will obtain polyalphabetic substitution.

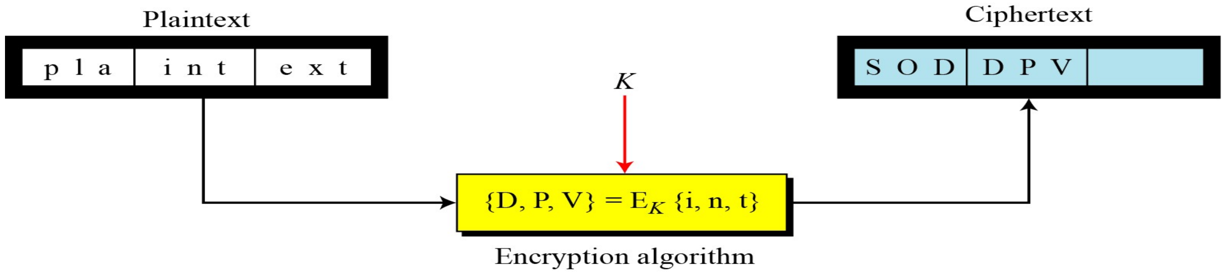
Block Cipher Principles

Stream Cipher and Block Cipher

- A stream cipher is one that encrypts a data stream one bit or one byte at a time. Example of stream cipher is the autokeyes, vigenere cipher and vernam cipher.



- A Block Cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Example of block cipher is DES.

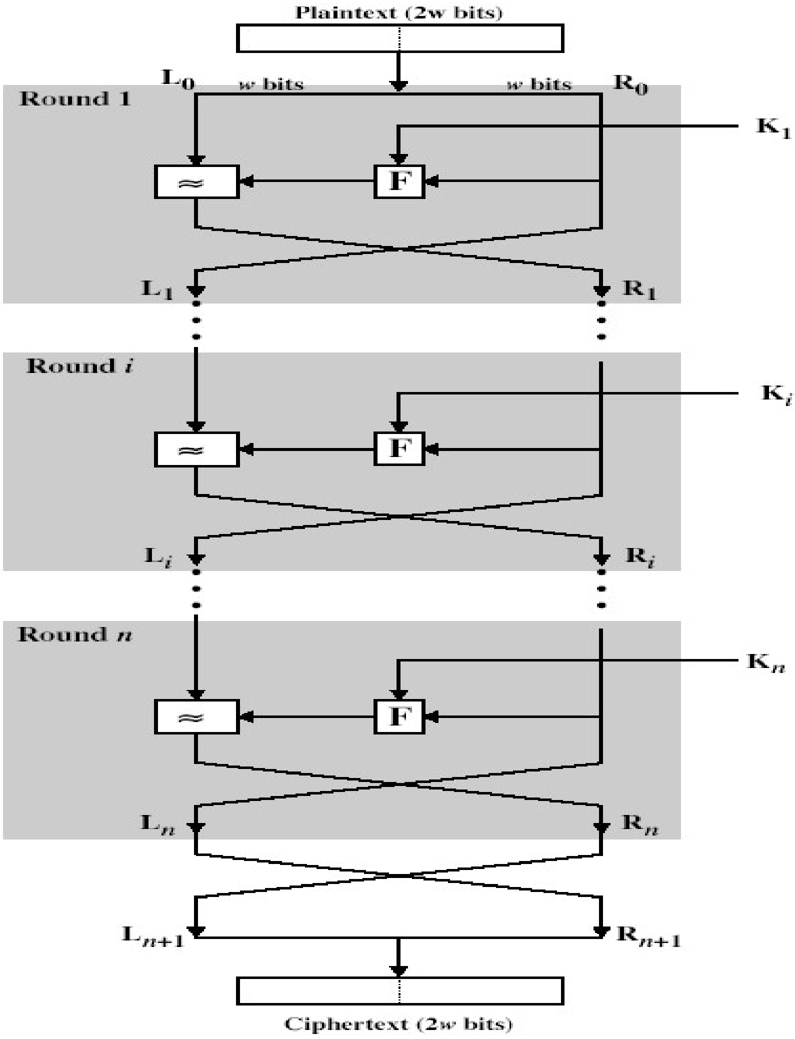


Block Cipher Principles

- block ciphers look like an extremely large substitution
- would need table of 2^{64} entries for a 64-bit block
- arbitrary reversible substitution cipher for a large block size is not practical
  - 64-bit general substitution block cipher, key size 2^{64}!
- most symmetric block ciphers are based on a Feistel Cipher Structure
- needed since must be able to decrypt cipher text to recover messages efficiently.

Feistel Cipher Structure

- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves



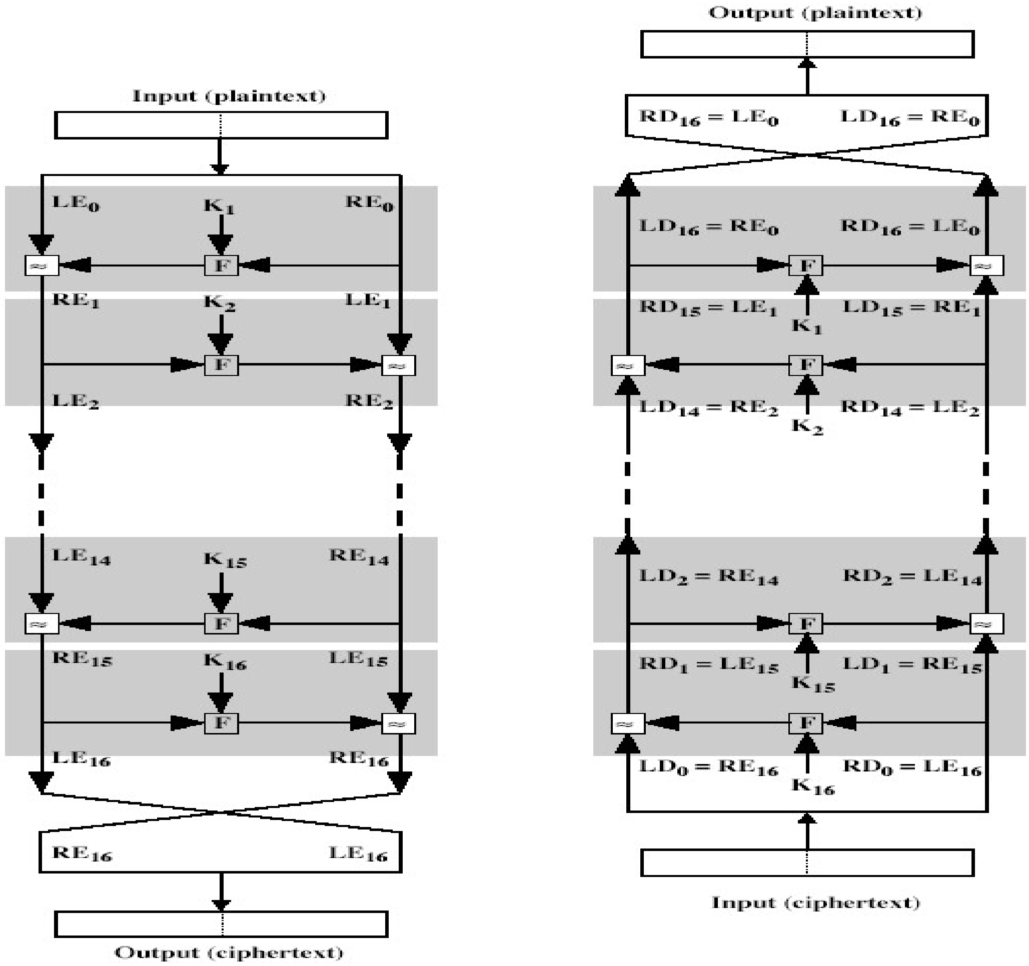
Feistel Cipher Design Principles

- **block size**
  - increasing size improves security, but slows cipher
- **key size**

- increasing size improves security, makes exhaustive key searching harder, but may slow cipher

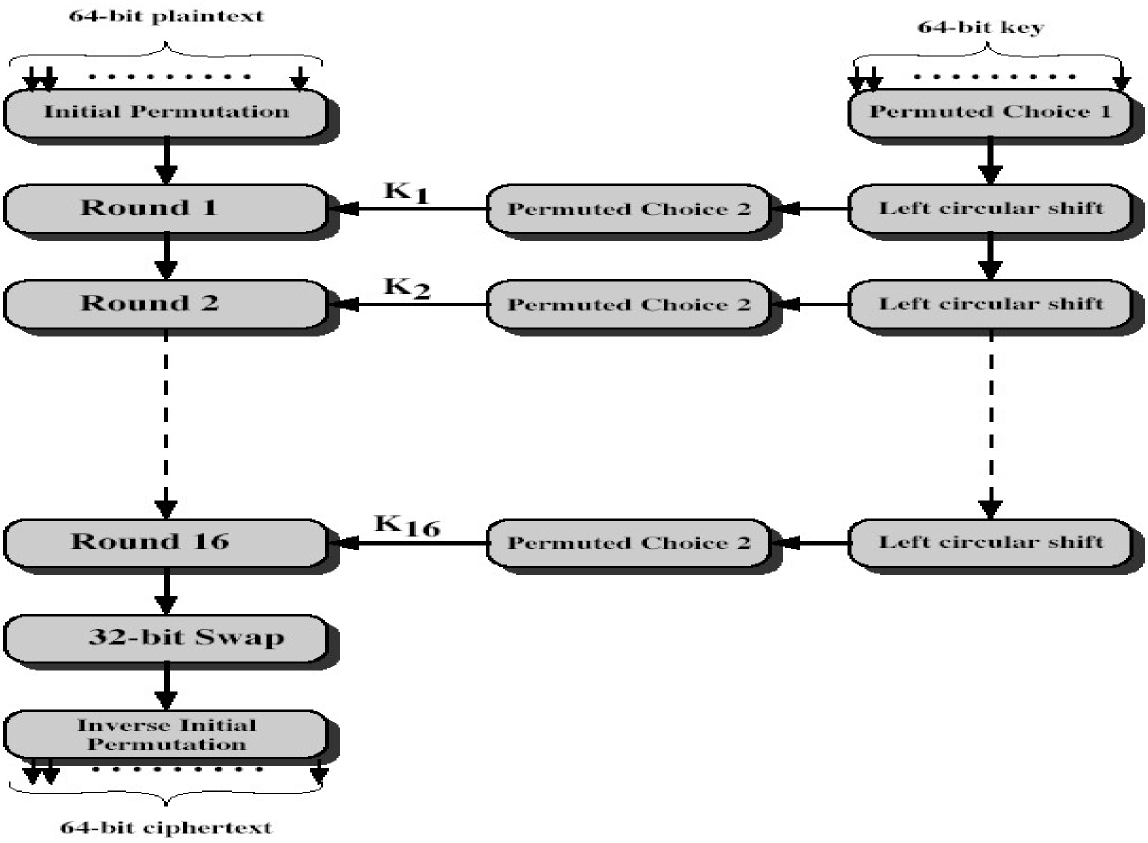
- **number of rounds**
  - increasing number improves security, but slows cipher
- **subkey generation**
  - greater complexity can make analysis harder, but slows cipher
- **round function**
  - greater complexity can make analysis harder, but slows cipher
- **fast software en/decryption & ease of analysis**
  - are more recent concerns for practical use and testing

Feistel Cipher Decryption



Data Encryption Standard (DES)

- encrypts 64-bit data using 56-bit key



Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

Initial and Final Permutation

- The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other. They have no cryptography significance in DES.

Round Function

- The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.
- uses two 32-bit L & R halvesas for any Feistel cipher can describe as:

- $L_i = R_{i-1}$
- $R_i = L_{i-1}$ xor $F(R_{i-1}, K_i)$

- takes 32-bit R half and 48-bit subkey and expands R to 48-bits using Expansion Permutation E and adds to subkey.
- passes through 8 S-boxes to get 32-bit result
- finally permutes this using 32-bit Permutation Function P

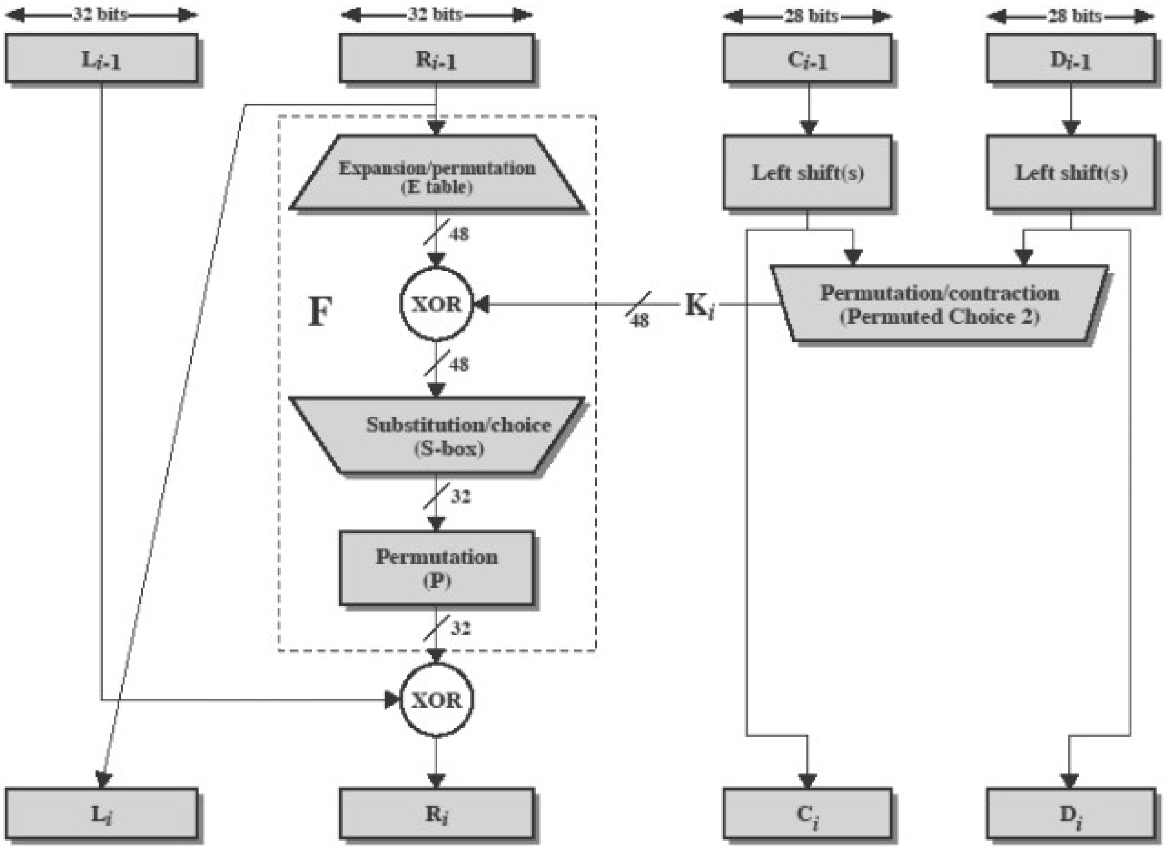
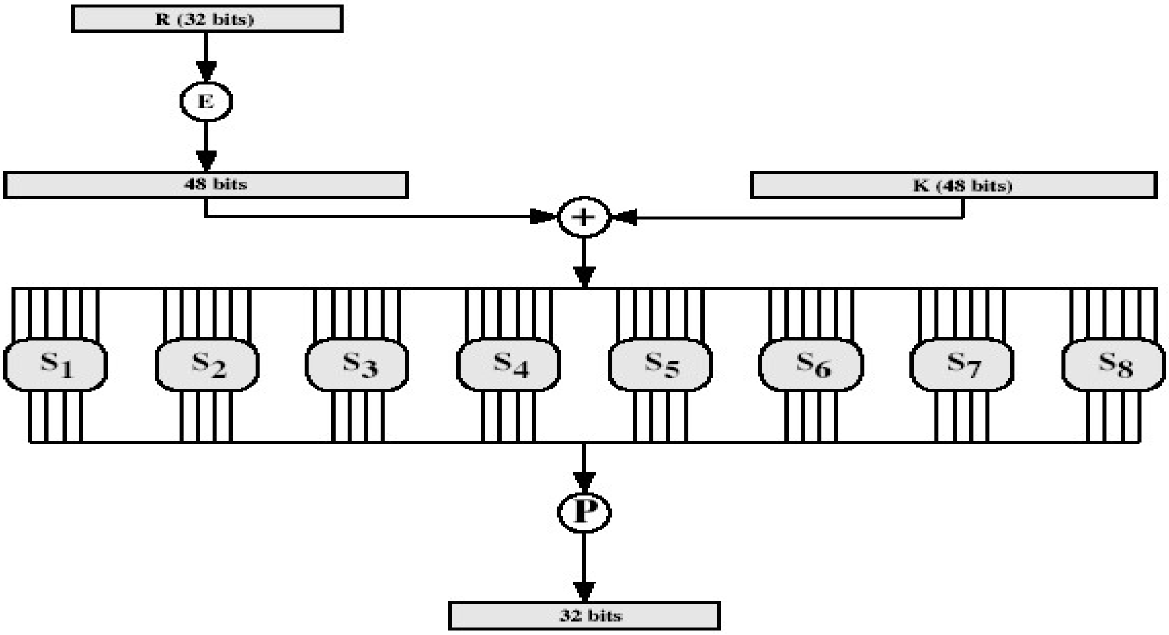


Figure 3.6 Single Round of DES Algorithm

Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits

Substitution Boxes S



- Each S-Box maps 6 to 4 bits
  - outer bits 1 & 6 (row bits) select the row
  - inner bits 2-5 (col bits) select the column
  - For example, in S1, for input 011001,
    - the row is 01 (row 1)
    - the column is 1100 (column 12).
    - The value in row 1, column 12 is 9
    - The output is 1001.
    - result is 8 X 4 bits, or 32 bits

DES Key Schedule

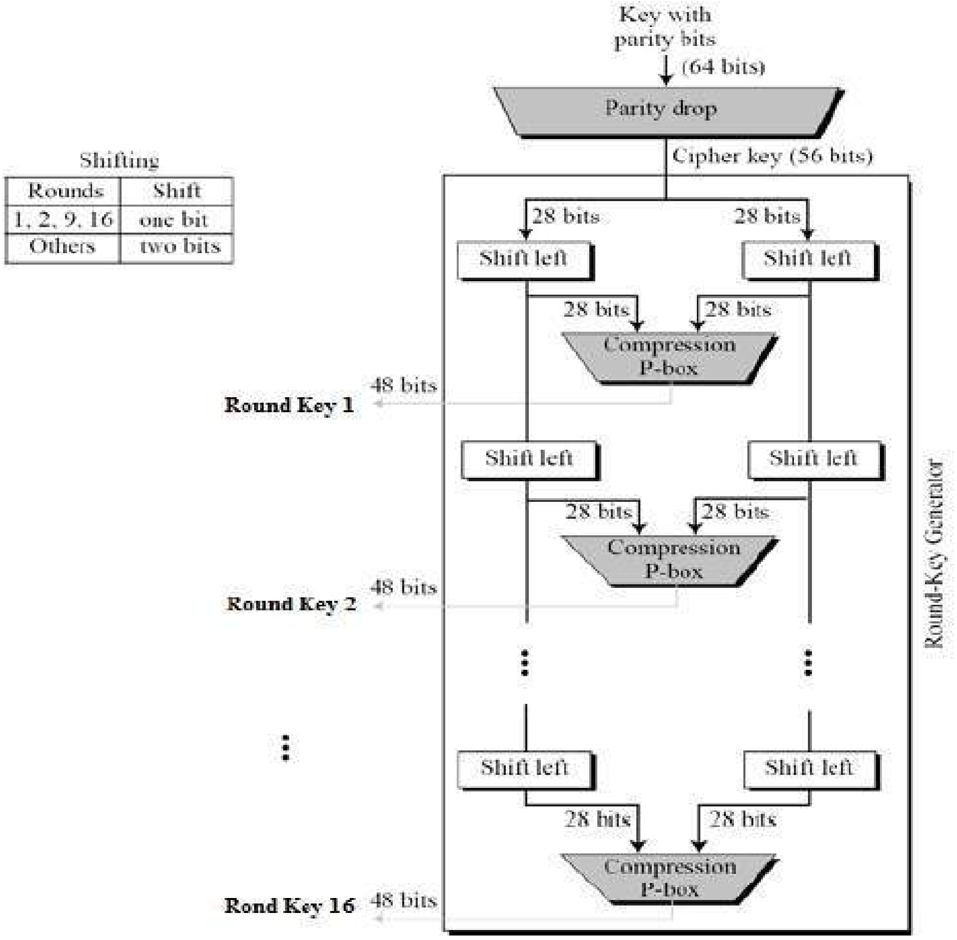
- forms subkeys used in each round

  1. initial permutation of the key.

  2. divide the 56-bits in two 28-bit halves

  3. at each round

- 3.1. Left shift each half (28bits) separately either 1 or 2 places based on the left shift schedule.
    - Shifted values will be input for next round
- 3.2. Combine two halfs to 56 bits, permuting them for use in function f



The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

- **Avalanche effect** − A small change in plaintext results in the very great change in the cipher text.
- **Completeness** − Each bit of cipher text depends on many bits of plaintext.

During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided.

DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

Strength of DES

1. **Key Size**
    - 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
    - brute force search looks hard
    - recent advances have shown is possible
        - in 1997 on Internet in a few months
        - in 1998 on dedicated hardware (EFF) in a few days
        - in 1999 above combined in 22hrs!
    - still must be able to recognize plaintext
    - now considering alternatives to DES
2. **Timing Attacks**
    - Attacks actual implementation of cipher.
    - Use knowledge of consequences of implementation to derive knowledge of some/all subkey bits.
    - Specifically use fact that calculations can take varying times depending on the value of the inputs to it.
3. **Analytic Attacks**
    - now have several analytic attacks on DES
    - these utilise some deep structure of the cipher
        - by gathering information about encryptions
        - can eventually recover some/all of the sub-key bits
        - if necessary then exhaustively search for the rest
    - generally these are statistical attacks include:
        - differential cryptanalysis
        - linear cryptanalysis
        - related key attacks

Differential Cryptanalysis

- a statistical attack against Feistel ciphers

- uses cipher structure not previously used
- design of S-P networks has output of function *f* influenced by both input & key
- hence cannot trace values back through cipher without knowing values of the key
- Differential Cryptanalysis compares two related pairs of encryptions

Linear Cryptanalysis

- another recent development
- also a statistical method
- based on finding linear approximations to model the transformation of DES
- can attack DES with 2^{47} known plaintexts, still in practise infeasible

Block Cipher Design Principles

- basic principles still like Feistel in 1970's
- number of rounds
    - more is better, exhaustive search best attack
- function f:
    - provides "confusion", is nonlinear, avalanche
- key schedule
    - complex subkey creation, key avalanche

Modes of Operation

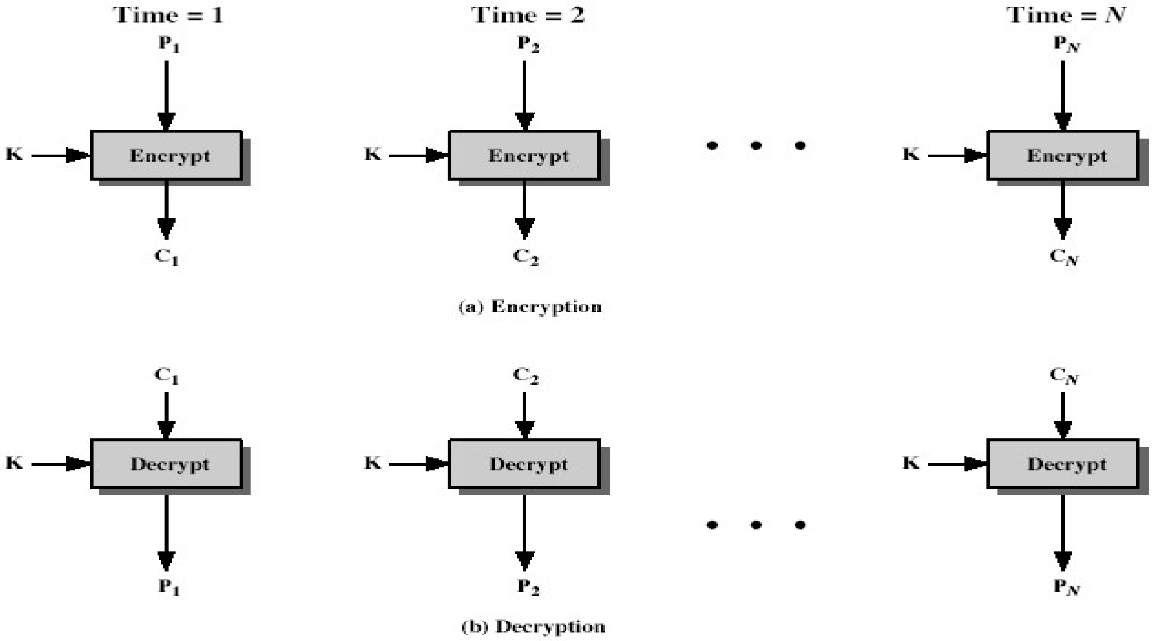
- block ciphers encrypt fixed size blocks
- e.g. DES encrypts 64-bit blocks, with 56-bit key
- need way to use in practice, given usually have arbitrary amount of information to encrypt
- four were defined for DES in ANSI standard ANSI X3.106-1983 Modes of Use
- subsequently now have 5 for DES and AES
- have block and stream modes

1. Electronic Codebook Book (ECB)

- message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
- each block is encoded independently of the other blocks

$$C_i = DES_{K1} (P_i)$$

- uses: secure transmission of single values



Advantages and Limitations of ECB

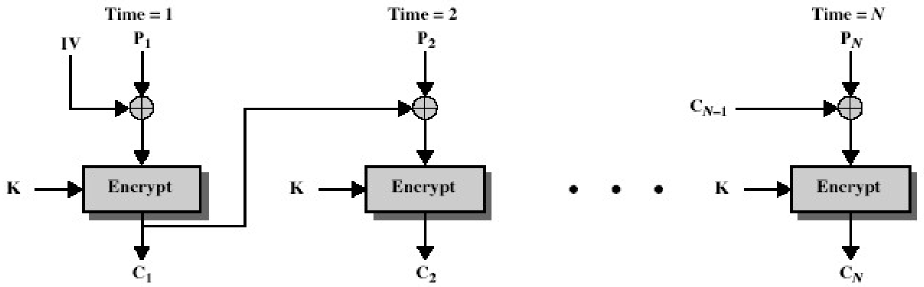
- repetitions in message may show in cipher text
  - if aligned with message block
  - particularly with data such as graphics
  - or with messages that change very little, which become a code-book analysis problem
- weakness due to encrypted message blocks being independent
- main use is sending a few blocks of data

2. Cipher Block Chaining (CBC)

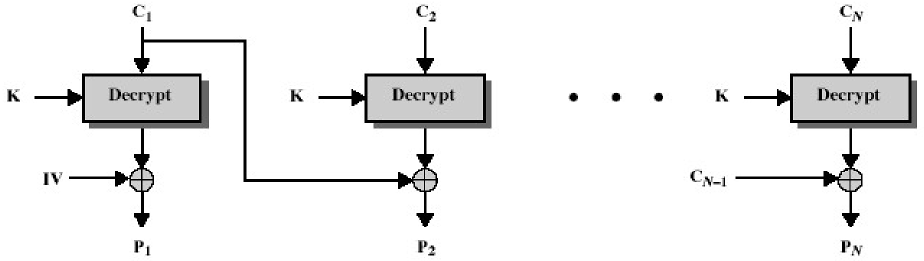
- message is broken into blocks
- but these are linked together in the encryption operation
- each previous cipher blocks is chained with current plaintext block, hence name

- use Initial Vector (IV) to start process
  - $C_i = DES_{K1}(P_i \text{ XOR } C_{i-1})$
  - $C_{-1} = IV$
- uses: bulk data encryption, authentication

Advantages and Limitations of CBC



(a) Encryption



(b) Decryption

Advantages and Limitations of CBC

- each ciphertext block depends on **all** message blocks
- thus a change in the message affects all ciphertext blocks after the change as well as the original block
- need **Initial Value** (IV) known to sender & receiver
  - however if IV is sent in the clear, an attacker can change bits of the first block, and change IV to compensate
  - hence either IV must be a fixed value or it must be sent encrypted in ECB mode before rest of message
- at end of message, handle possible last short block
  - by padding either with known non-data value (eg nulls)
  - or pad last block with count of pad size

  - eg. [b1 b2 b3 0 0 0 0 5] <- 3 data bytes, then 5 bytes pad+count

Cipher FeedBack (CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8 or 64 or whatever) to be feed back
  - denoted CFB-1, CFB-8, CFB-64 etc
- is most efficient to use all 64 bits (CFB-64)

$C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$

$C_{-1} = IV$

- uses: stream data encryption, authentication

Advantages and Limitations of CFB

- appropriate when data arrives in bits/bytes
- most common stream mode
- limitation is need to stall while do block encryption after every n-bits
- note that the block cipher is used in encryption mode at both ends
- errors propagate for several blocks after the error

Output FeedBack (OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance

$C_i = P_i \text{ XOR } O_i$

$O_i = DES_{K1}(O_{i-1})$

$O_{-1} = IV$

- uses: stream encryption over noisy channels

Advantages and Limitations of OFB

- used when error feedback a problem or where need to encryptions before message is available
- superficially similar to CFB
- but feedback is from the output of cipher and is independent of message
- a variation of a Vernam cipher
  - hence must **never** reuse the same sequence (key+IV)
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- originally specified with m-bit feedback in the standards
- subsequent research has shown that only **OFB-64** should ever be used

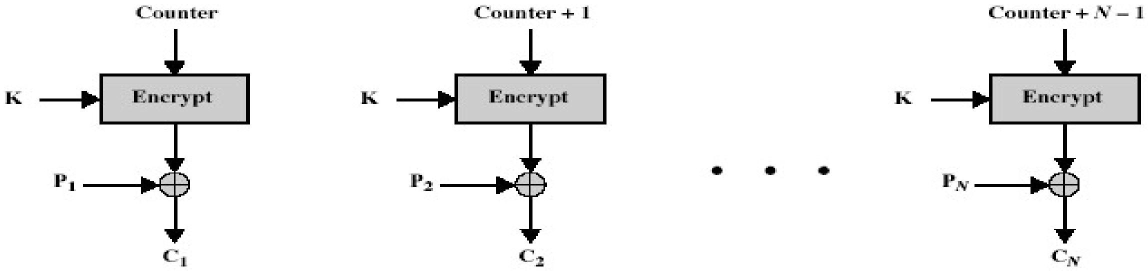
Counter (CTR)

- a "new" mode, though proposed early on
- similar to OFB but encrypts counter value rather than any feedback value
- must have a different key & counter value for every plaintext block (never reused)

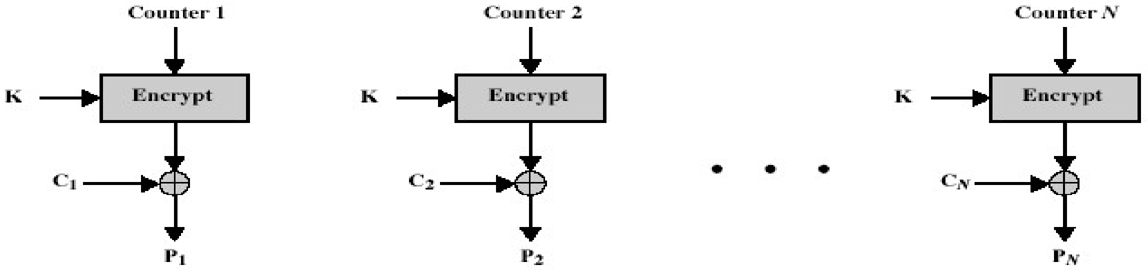
$C_i = P_i \text{ XOR } O_i$

$O_i = DES_{K1}(i)$

- uses: high-speed network encryptions



(a) Encryption



(b) Decryption

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for bursty high speed links
- random access to encrypted data blocks
- provable security (good as other modes)

– but must ensure never reuse key/counter values, otherwise could break (cf OFB).