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.

1. INTRODUCTION

Security basics

- **Security** refers to any measures taken to protect something. Examples of security in the real world include locks on doors, alarms in our cars, police officers.

- **Computer security** is a field of computer science concerned with the control of risks related to computer use. It describe the methods of protecting the integrity of data stored on a computer. In computer security the measures taken are focused on securing individual computer hosts.

- **Network security** consists of the provisions made in an underlying computer network infrastructure, policies adopted by the network administrator to protect the network and the network-accessible resources from unauthorized access and the effectiveness (or lack) of these measures combined together. It starts from authenticating any user. Once authenticated, firewall enforces access policies such as what services are allowed to be accessed by the network users. Even though it prevents unauthorized access, it prevents harmful contents such as computer worms being transmitted over the network. An intrusion prevention system (IPS) helps detect and prevent such malware.

- **Goals of Computer Security**

❖ Integrity: Guarantee that the data is what we expect
❖ Confidentiality: The information must just be accessible to the authorized people
❖ Authentication: Guarantee that only authorized persons can access to the resources

- **SECURITY ATTACKS**

Any action that compromises security of information is called a security attack. Some of the common security attacks are given below.

- **Passive attack**: aims to learn or make use of information from the system but does not affect system resources.

- **Active attack**: attempts to alter system resources or affect their operation

Attacks can be active or passive

Passive Attacks

 goal to obtain information

 No modification of content or fabrication

 Eavesdropping to learn contents or other information (transfer patterns, traffic flows etc.)

 Two types

❖ Release of message contents

❖ Traffic analysis

- **Active Attacks**

  □ modification of content and/or participation in communication to

  □ Four types

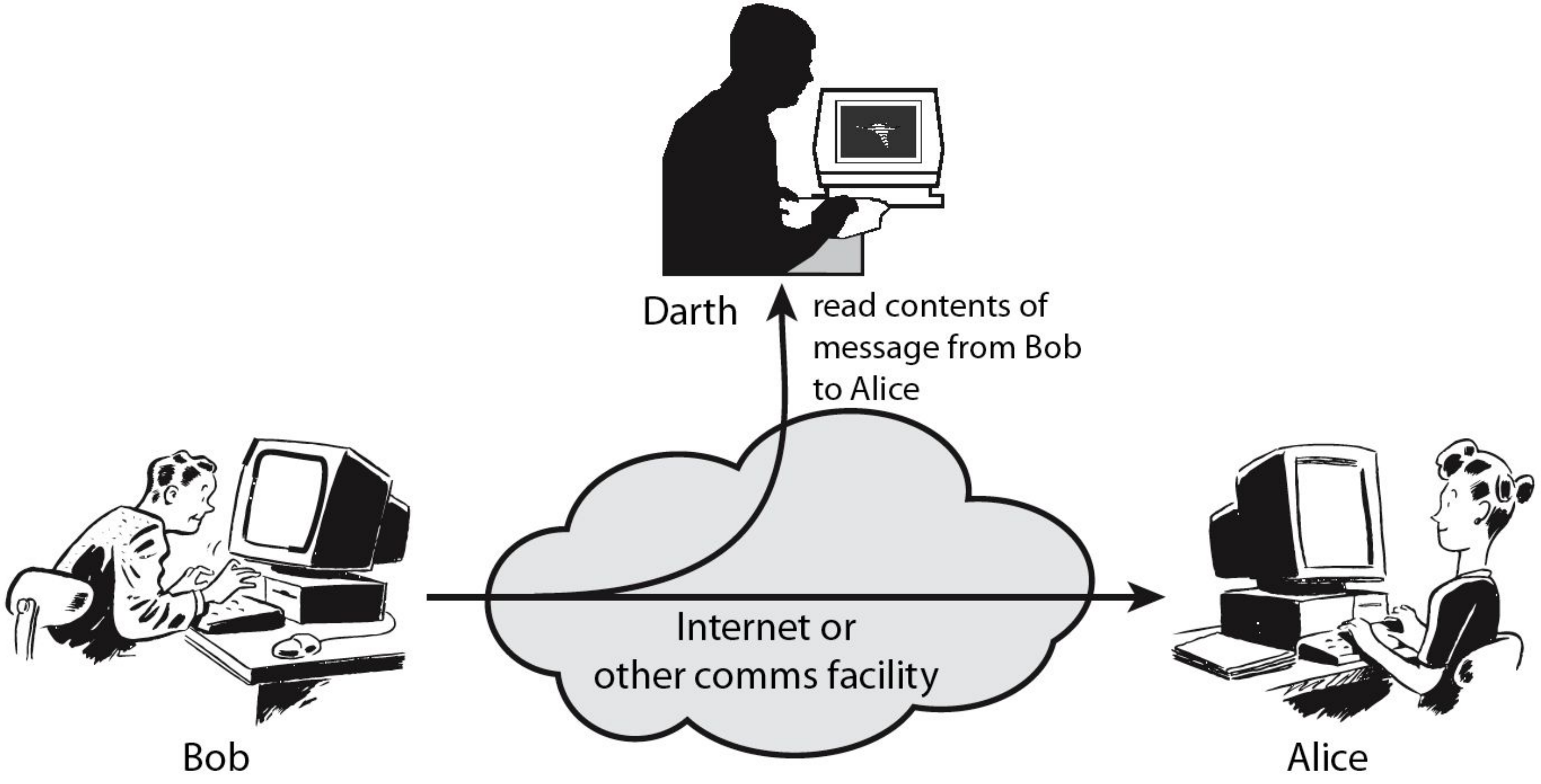
  ❖ Impersonate legitimate parties (Masquerade)

  ❖ Replay or retransmit

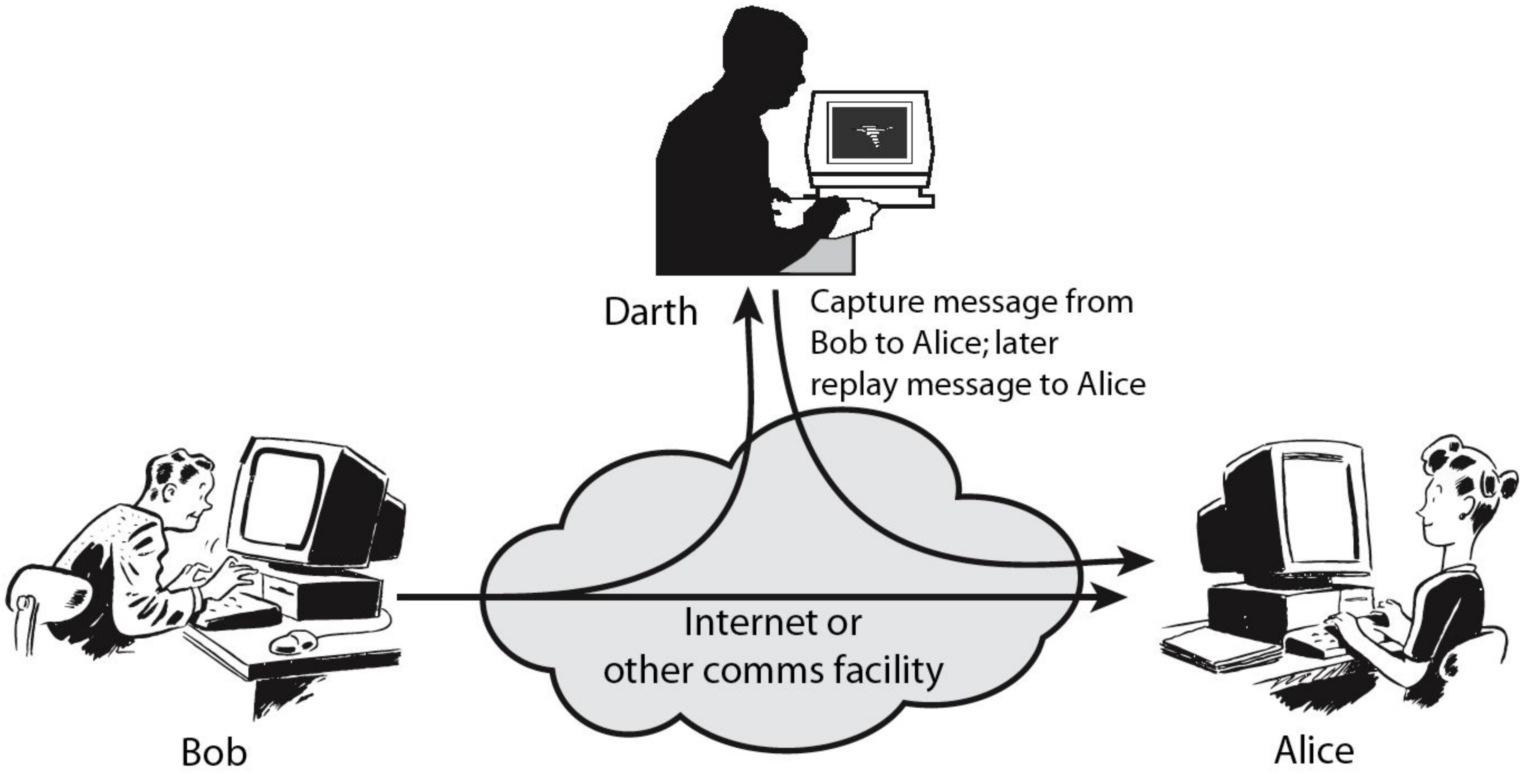
  ❖ Modify the content in transit

  ❖ Launch denial of service attacks

Passive Attacks


Darth
read contents of message from Bob to Alice
Internet or other comms facility
Bob
Alice

Active Attacks



Definitions

- Cryptography = the science (art) of encryption
- Cryptanalysis = the science (art) of breaking encryption
- Cryptology = cryptography + cryptanalysis

- Symmetric encryption, also referred to as conventional encryption or single-key encryption, was the only type of encryption in use prior to the development of public key encryption
- An original message is known as the **plaintext**, while the coded message is called the **ciphertext**.
- The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**; restoring the plaintext from the ciphertext is **deciphering** or **decryption**.
- The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**.
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**.
- Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called **cryptology**

Ques 1) What is cryptography?

Ans: Cryptography

Cryptography or Cryptology refers to **hiding secret**. It is a practice or technique of secure communication through unsecure channels due to presence of third parties (commonly referred to as adversaries). Cryptography is a technique of converting data in a form that is unreadable for third parties and transmitting to the destination, this data known as encrypted message and is not meaningful for any eavesdropper in the way. When the encrypted data reaches to the destination, the legitimate receiver (who knows how to convert it to original message) decrypts the message to original data.

The **ISO 7498-2 standard** uses the terms 'Encipher' in place of 'Encryption' and 'Decipher' in place of 'Decryption'. The reason is that the terms 'Encrypt' and 'Decrypt' thought to be offensive as they are used in terms of dead bodies.

In general, cryptography refers to constructing and analysing protocols to protect data confidentiality by preventing third parties to read private messages (**figure 1.1**).

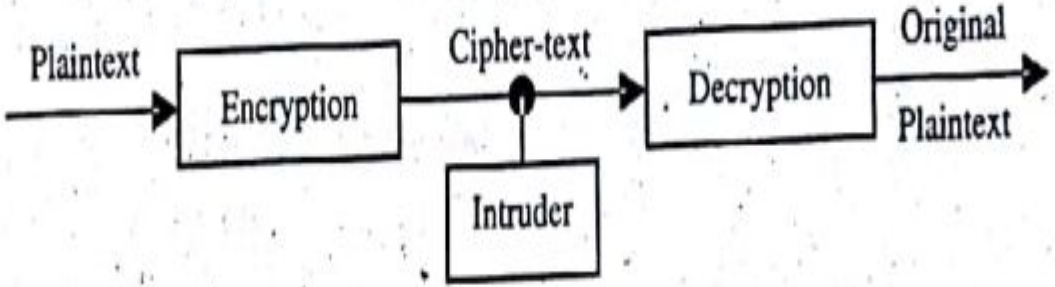


Figure 1.1: Cryptography

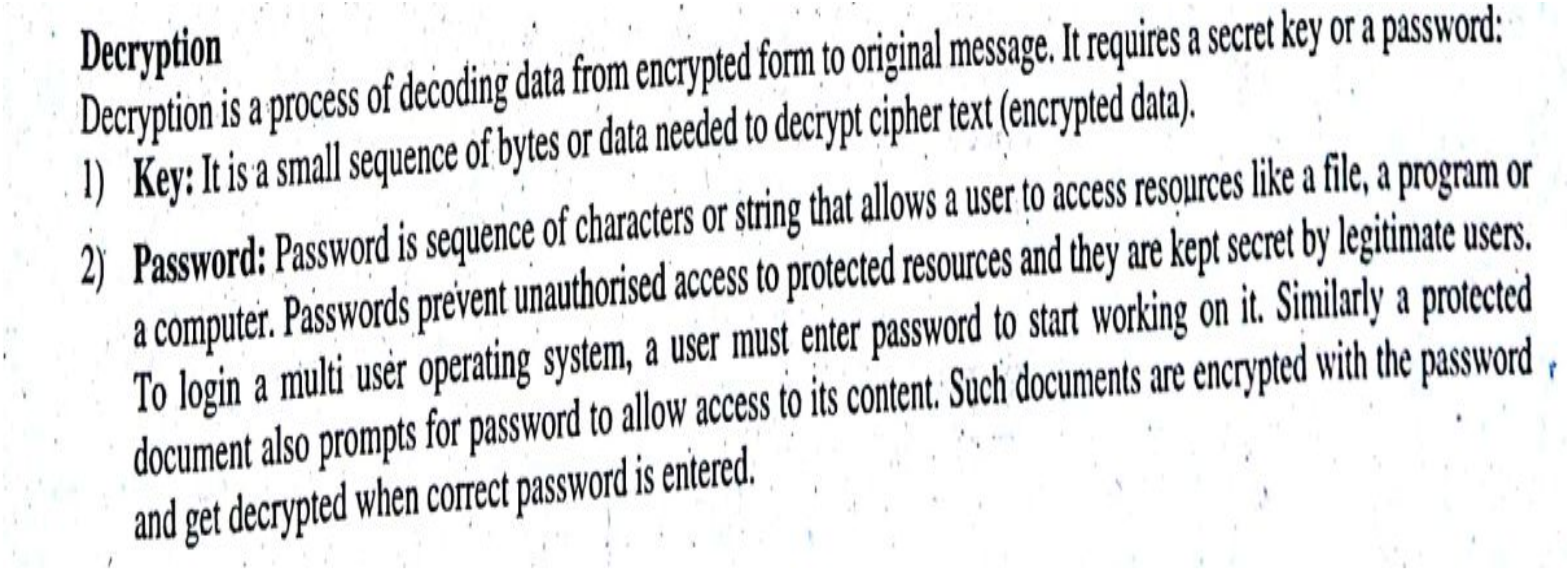
Ques 2) What is encryption and decryption?

Ans: Encryption

By the process of encryption it is possible to secure data and other resources that operate on the computer network. This specially involves extranets, intranets and Internet. It has become possible to transmit data like passwords and messages in the scrambled form. The authorised computer has the ability to unscramble the data and see it in the original form.

The process of encryption converts the digital data into the scrambled form with the help of keys or mathematical algorithms. After this, the scrambled code is transmitted. The receiver decodes the data it has received. One of the popular methods of encryption is the use of a pair of public and private keys that is unique for every individual.

For example, it is possible to scramble e-mail with the help of a unique public key. The sender knows this key. The recipient can unscramble the message using his/her secret private key only.



Decryption

Decryption is a process of decoding data from encrypted form to original message. It requires a secret key or a password:

1) **Key:** It is a small sequence of bytes or data needed to decrypt cipher text (encrypted data).

2) **Password:** Password is sequence of characters or string that allows a user to access resources like a file, a program or a computer. Passwords prevent unauthorised access to protected resources and they are kept secret by legitimate users. To login a multi user operating system, a user must enter password to start working on it. Similarly a protected document also prompts for password to allow access to its content. Such documents are encrypted with the password and get decrypted when correct password is entered.

Classical Encryption Techniques

- It is of two types, namely one is Symmetric Cipher Model and Substitution Techniques.

- **Key Points**

- Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption.

- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext.

- The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, and brute-force, which involves trying all possible keys.

- Traditional (precomputer) symmetric ciphers use substitution and/or transposition techniques. Substitution techniques map plaintext elements (characters, bits) into ciphertext elements. Transposition techniques systematically transpose the positions of plaintext elements.

- **Rotor machines** are sophisticated precomputer hardware devices that use substitution techniques.
- **Steganography** is a technique for hiding a secret message within a larger one in such a way that others cannot discern the presence or contents of the hidden message.
- An original message is known as the plaintext , while the coded message is called the ciphertext.
- The process of converting from plaintext to ciphertext is known as enciphering or encryption; restoring the plaintext from the ciphertext is deciphering or decryption.
- The many schemes used for encryption constitute the area of study known as cryptography. Such a scheme is known as a cryptographic system or a cipher. Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of cryptanalysis. Cryptanalysis is what the layperson calls "breaking the code." The areas of cryptography and cryptanalysis together are called cryptology.

- **Key**

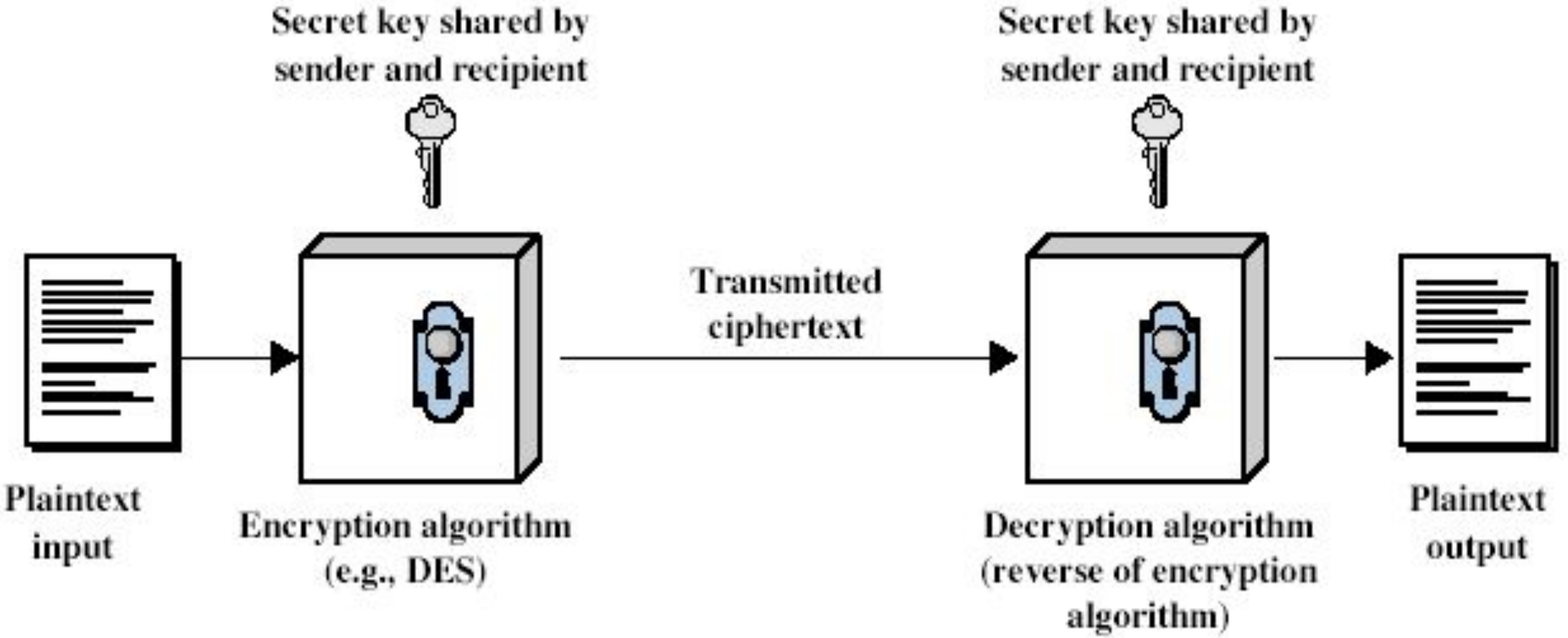
- A key in cryptology is similar to a key we use for locking and unlocking things in everyday life.
- In cryptography, keys are the bits and bytes used in the process of encryption and decryption.
- In this case, a key is a very large number that has special mathematical properties. Breaking into an encryption scheme depends on knowledge of the key or the ability to discover the key. The larger the key, the more difficult it is to discover

SYMMETRIC CIPHER MODEL

- A symmetric encryption scheme has five ingredients (see the given figure)

- **Plaintext**: This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm**: The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a 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 depend on the key.
- **Ciphertext:** This is the 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 ciphertexts.
- The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm**: This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Figure : Simplified Model of Conventional Encryption


Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

- There are two requirements for secure use of conventional encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more ciphertexts would be unable to decipher the ciphertext or figure out the key. This requirement is usually stated in a stronger form: The opponent should be unable to decrypt ciphertext or discover the key even if he or she is in possession of a number of ciphertexts together with the plaintext that produced each ciphertext.
2. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable

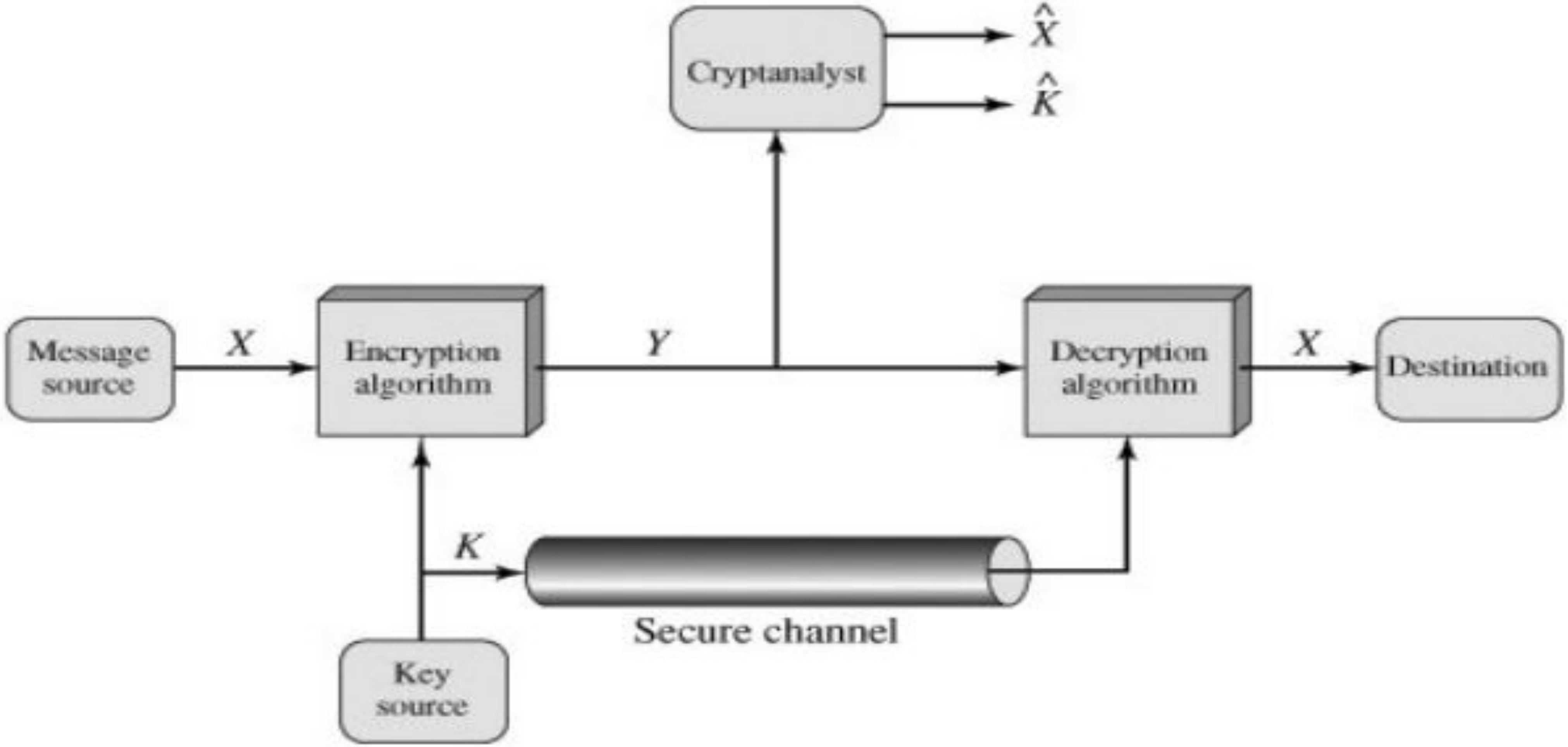
- We assume that it is impractical to decrypt a message on the basis of the ciphertext plus knowledge of the encryption/decryption algorithm.
- In other words, we do not need to keep the algorithm secret; we need to keep only the key secret. This feature of symmetric encryption is what makes it feasible for widespread use.
- The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of data encryption algorithms.
- These chips are widely available and incorporated into a number of products. With the use of symmetric encryption, the principal security problem is maintaining the secrecy of the key

- Let us take a closer look at the essential elements of a symmetric encryption scheme, using Figure 2.2. A source produces a message in plaintext, $X= [X_1, X_2, ...,X_M]$.

- The M elements of X are letters in some finite alphabet. Traditionally, the alphabet usually consisted of the 26 capital letters. Nowadays, the binary alphabet {0, 1} is typically used.

- For encryption, a key of the form $K= [K_1,K_2, ...,K_J]$ is generated. If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel. Alternatively, a third party could generate the key and securely deliver it to both source and destination.

Figure 2.2. Model of Conventional Cryptosystem



- With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext Y= [Y_1, Y_2, ..., Y_N]. We can write this as

- **Y= E $_{\mathbf{K}}$ (X)**
- This notation indicates that Y is produced by using encryption algorithm E as a function of the plaintext X, with the specific function determined by the value of the key K.
- The intended receiver, in possession of the key, is able to invert the transformation:
- **X = D $_{\mathbf{K}}$ (Y)**
- An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both X and K. It is assumed that the opponent knows the encryption (E) and decryption (D) algorithms.
- If the opponent is interested in only this particular message, then the focus of the effort is to recover X by generating a plaintext estimate
- Often, however, the opponent is interested in being able to read future $\hat{X}.$ messages as well, in which case an attempt is made to recover K by generating an estimate

- $Y = \mathrm{E}K(X)$ or $Y = \mathrm{E}(K, X)$
  $X = \mathrm{D}K(Y)$ or $X = \mathrm{D}(K, Y)$
  X = plaintext Y = ciphertext K = secret key
  E = encryption algorithm D = decryption algorithm

•Cryptography

•Cryptographic systems are characterized along three independent dimensions:

- **The type of operations used for transforming plaintext to ciphertext.** 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, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.
- **The 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.
- **The way in which the plaintext is processed.** A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

•Cryptanalysis

•Typically, the objective of attacking an encryption system is to recover the key in use rather then simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:

- **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
- **Brute-force attack :** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

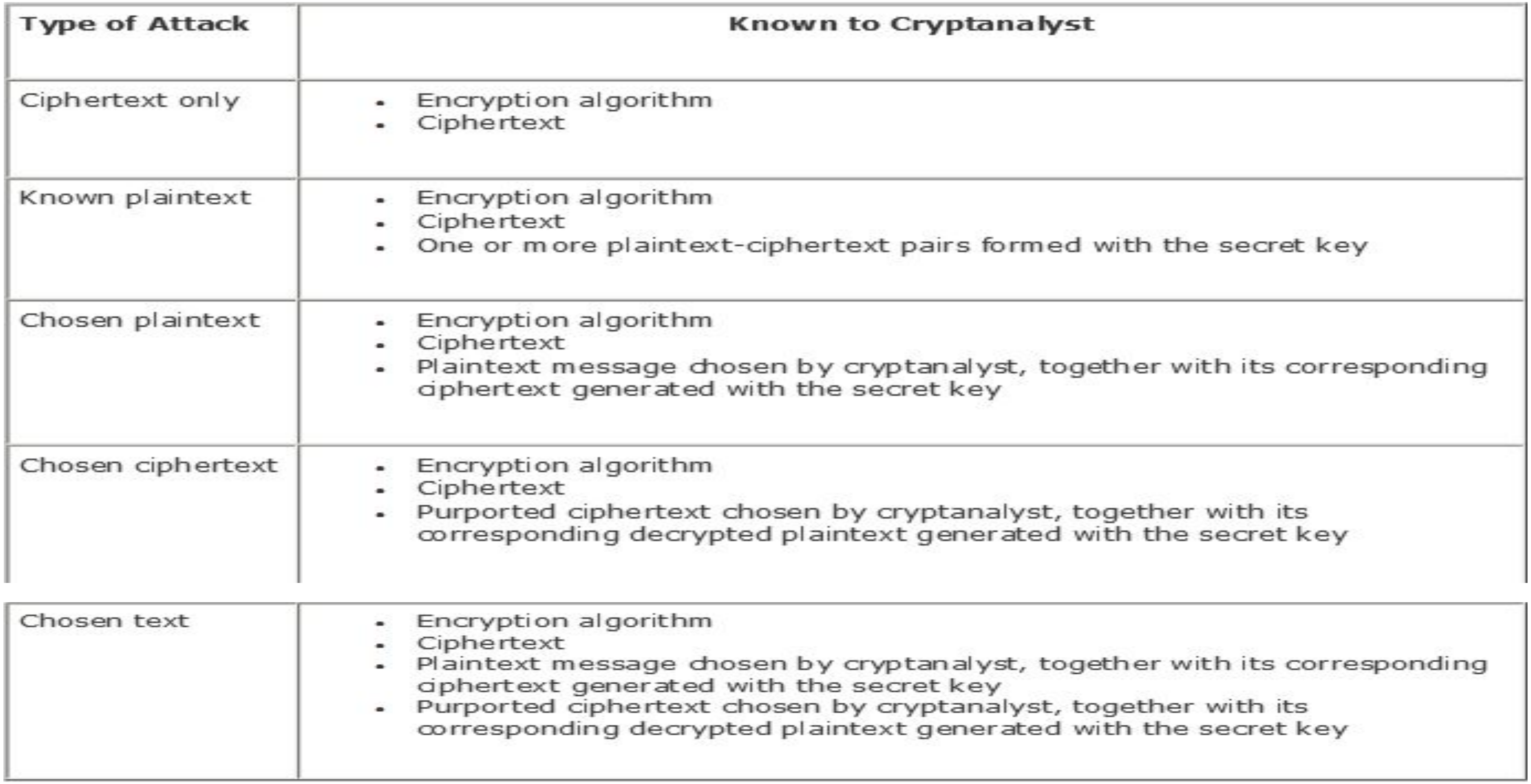
- We first consider cryptanalysis and then discuss brute-force attacks.

- Table 2.1 summarizes the various types of cryptanalytic attacks, based on the amount of information known to the cryptanalyst. The most difficult problem is presented when all that is available is the ciphertext only.

- In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute- force approach of trying all possible keys.

- If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plaintext that is concealed, such as English or French text, an EXE file, a Java source listing, an accounting file, and so on.

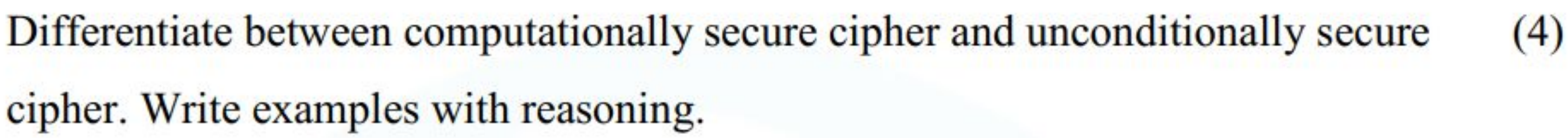
Table 2.1. Types of Attacks on Encrypted Messages



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

- The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of information to work with. In many cases, however, the analyst has more information.
- The analyst may be able to capture one or more plaintext messages as well as their encryptions. Or the analyst may know that certain plaintext patterns will appear in a message.
- For example, a file that is encoded in the Postscript format always begins with the same pattern, or there may be a standardized header or banner to an electronic funds transfer message, and so on.
- All these are examples of known plaintext. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed

- Closely related to the known-plaintext attack is what might be referred to as a probable-word attack. If the opponent is working with the encryption of some general prose message, he or she may have little knowledge of what is in the message.
- However, if the opponent is after some very specific information, then parts of the message may be known. For example, if an entire accounting file is being transmitted, the opponent may know the placement of certain key words in the header of the file. As another example, the source code for a program developed by Corporation X might include a copyright statement in some standardized position.
- If the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a chosen-plaintext attack is possible. An example of this strategy is differential cryptanalysis, explored in Chapter 3. In general, if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.



Differentiate between computationally secure cipher and unconditionally secure cipher. Write examples with reasoning. (4)

- Table 2.1 lists two other types of attack: chosen ciphertext and chosen text. These are less commonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

- Only relatively weak algorithms fail to withstand a ciphertext-only attack. Generally, an encryption algorithm is designed to withstand a known-plaintext attack.

- Two more definitions are worthy of note. An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext, simply because the required information is not there. With the exception of a scheme known as the one-time pad (described later in this chapter), there is no encryption algorithm that is unconditionally secure. Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

- The cost of breaking the cipher exceeds the value of the encrypted information.

- The time required to break the cipher exceeds the useful lifetime of the information

- An encryption scheme is said to be **computationally secure** if either of the foregoing two criteria are met. The rub is that it is very difficult to estimate the amount of effort required to cryptanalyze ciphertext successfully.

- All forms of cryptanalysis for symmetric encryption schemes are designed to exploit the fact that traces of structure or pattern in the plaintext may survive encryption and be discernible in the ciphertext. This will become clear as we examine various symmetric encryption schemes in this chapter. We will see in Part Two that cryptanalysis for public-key schemes proceeds from a fundamentally different premise, namely, that the mathematical properties of the pair of keys may make it possible for one of the two keys to be deduced from the other.

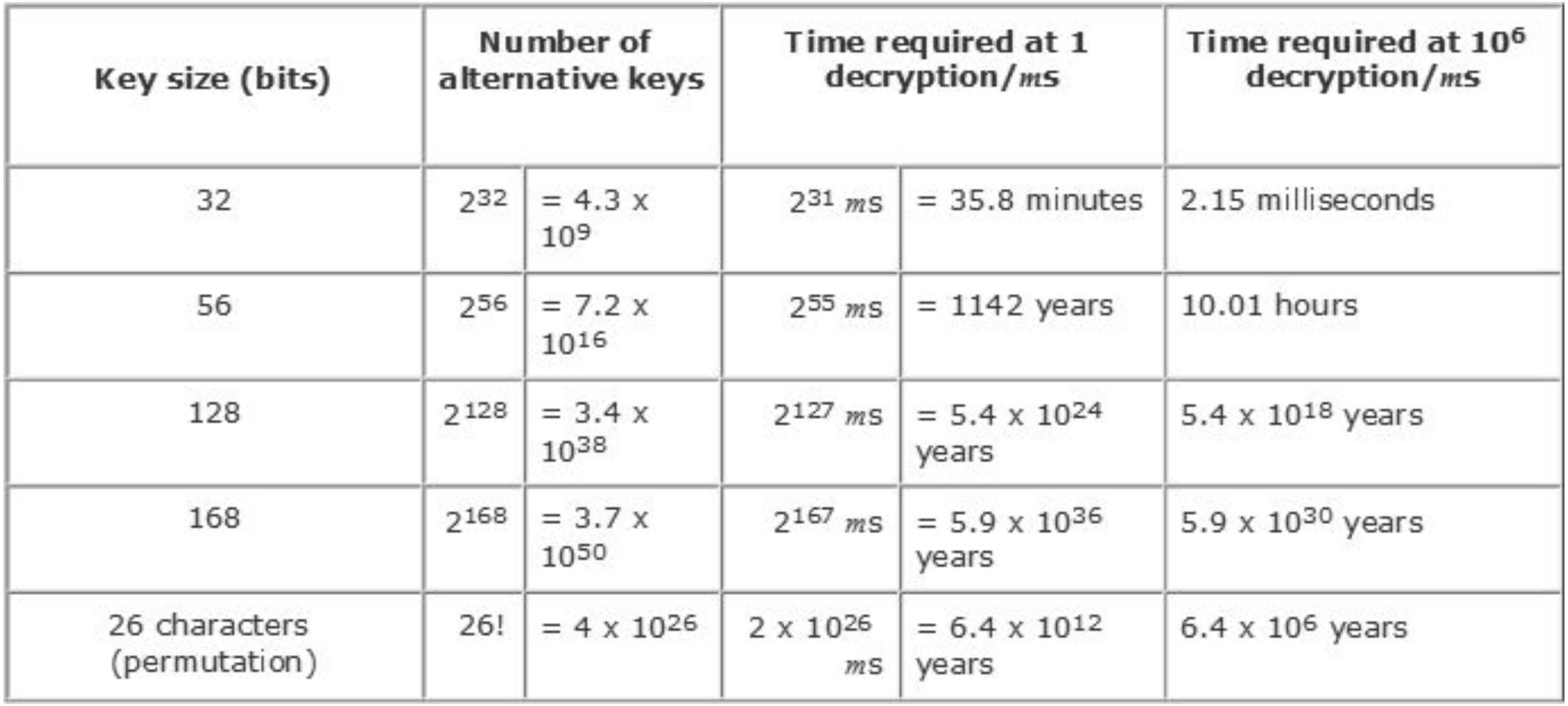
- **A brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.

- Table 2.2 shows how much time is involved for various key spaces. Results are shown for four binary key sizes. The 56-bit key size is used with the DES (Data Encryption Standard) algorithm, and the 168-bit key size is used for triple DES.

- The minimum key size specified for AES (Advanced Encryption Standard) is 128 bits. Results are also shown for what are called substitution codes that use a 26-character key (discussed later), in which all possible permutations of the 26 characters serve as keys.

- For each key size, the results are shown assuming that it takes 1 ms to perform a single decryption, which is a reasonable order of magnitude for today's machines. With the use of massively parallel organizations of microprocessors, it may be possible to achieve processing rates many orders of magnitude greater. The final column of Table 2.2 considers the results for a system that can process 1 million keys per microsecond. As you can see, at this performance level, DES can no longer be considered computationally secure

Table 2.2. Average Time Required for Exhaustive Key Search



Key size (bits)	Number of alternative keys		Time required at 1 decryption/*m*s		Time required at 10^6 decryption/*m*s
32	2^{32}	= 4.3 x 10^9	2^{31} *m*s	= 35.8 minutes	2.15 milliseconds
56	2^{56}	= 7.2 x 10^{16}	2^{55} *m*s	= 1142 years	10.01 hours
128	2^{128}	= 3.4 x 10^{38}	2^{127} *m*s	= 5.4 x 10^{24} years	5.4 x 10^{18} years
168	2^{168}	= 3.7 x 10^{50}	2^{167} *m*s	= 5.9 x 10^{36} years	5.9 x 10^{30} years
26 characters (permutation)	26!	= 4 x 10^{26}	2 x 10^{26} *m*s	= 6.4 x 10^{12} years	6.4 x 10^6 years

- **Cryptanalysis**

  Objective: to recover the plaintext of a ciphertext or, more typically, to recover the secretkey.

  □ Two general approaches:

  – brute-force attack

  – non-brute-force attack (cryptanalytic attack)

Brute-Force Attack

□ Try every key to decipher the ciphertext.

□ On average, need to try half of all possible keys

□ Time needed proportional to size of key space

Cryptanalytic Attacks

□ May be classified by how much information needed by the attacker:

✔ Ciphertext-only attack

✔ Known-plaintext attack

✔ Chosen-plaintext attack

✔ Chosen-ciphertext attack

✔ Chosen Text

- **More Definitions**

  ● **Unconditional security**

  ● no matter how much computer power is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext

● **Computational security**

● Cost of breaking the cipher exceeds the value of the encrypted information .

The time required to break the cipher exceeds the life time of the information

2.2. SUBSTITUTION TECHNIQUES

- In this section and the next, we examine a sampling of what might be called classical encryption techniques. A study of these techniques enables us to illustrate the basic approaches to symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated.

- The two basic building blocks of all encryption techniques are substitution and transposition. We examine these in the next two sections. Finally, we discuss a system that combines both substitution and transposition.

- A substitution technique is one in which 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 ciphertext bit patterns.

- *When letters are involved, the following conventions are used in this book. Plaintext is always in lowercase; ciphertext is in uppercase; key values are in italicized lowercase.*

- **Caesar Cipher**

- The earliest known use of a substitution cipher, and the simplest, was by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet. For example,

    - plain: meet me after the toga party

    - cipher: PHHW PH DIWHU WKH WRJD SDUWB

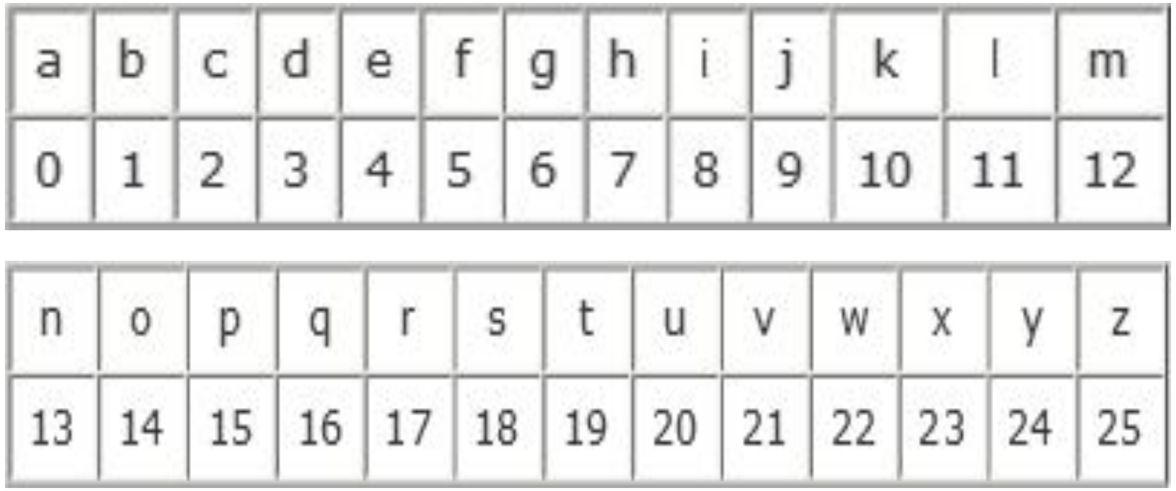
- Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

    - plain: 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

    - cipher: 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

- This cryptosystem is generally referred to as the **Shift Cipher**. The concept is to replace each alphabet by another alphabet which is ‘shifted’ by some fixed number between 0 and 25.
- For this type of scheme, both sender and receiver agree on a ‘secret shift number’ for shifting the alphabet. This number which is between 0 and 25 becomes the key of encryption.
- The name ‘Caesar Cipher’ is occasionally used to describe the Shift Cipher when the ‘shift of three’ is used.

- Let us assign a numerical equivalent to each letter:



a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Then the algorithm can be expressed as follows. For each plaintext letter p, substitute the ciphertext letter C:
- *We define a mod n to be the remainder when a is divided by n. For example, 11 mod 7 = 4.*

•$C = E(3, p) = (p + 3) \bmod 26$

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 1 to 25.

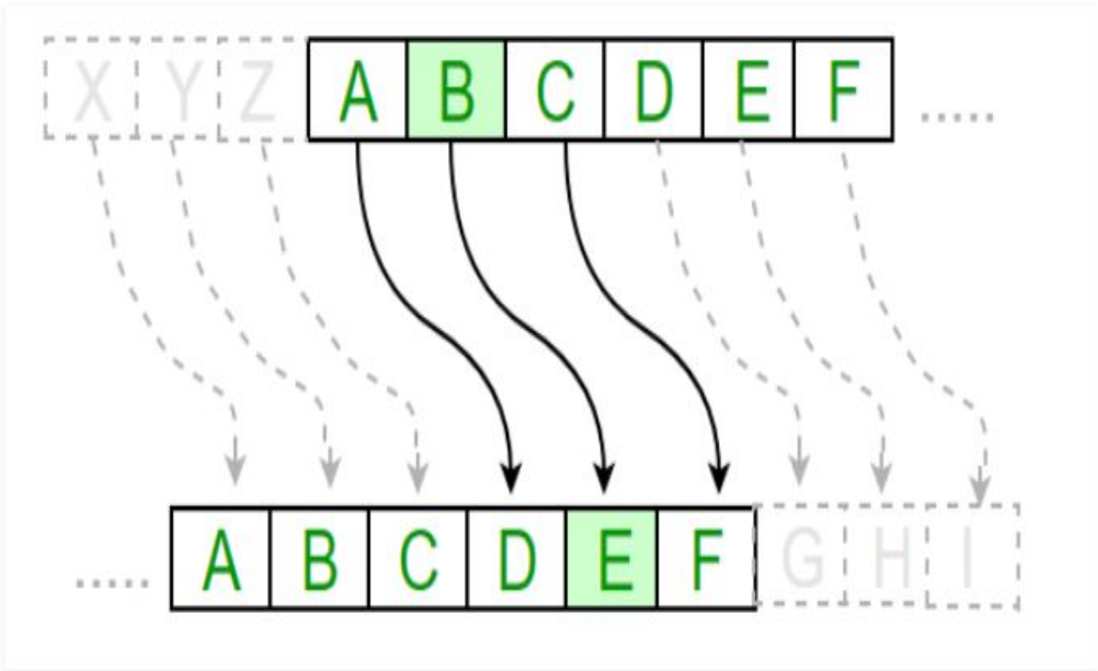
•The decryption algorithm is simply $p = D(k, C) = (C - k) \bmod 26$

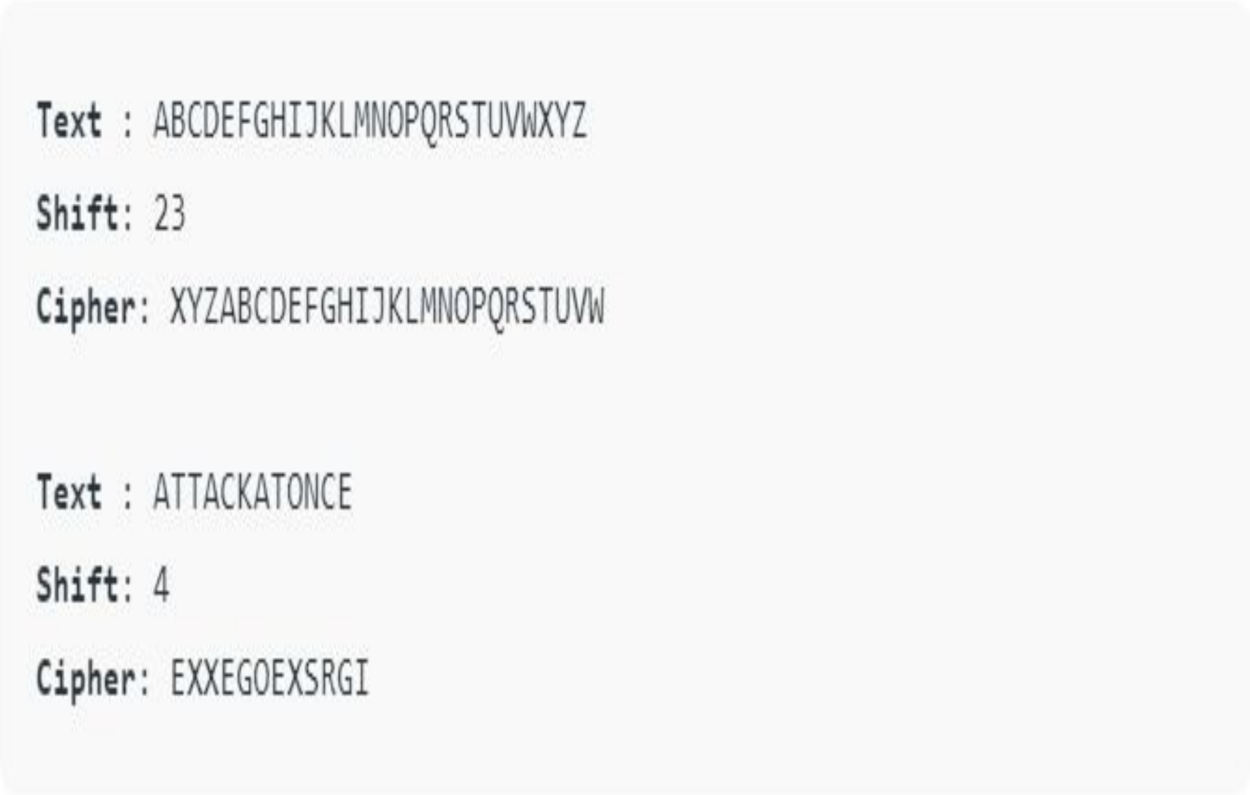
$$E_n(x) = (x+n) mod\ 26$$

(Encryption Phase with shift n)

$$D_n(x) = (x-n) mod\ 26$$

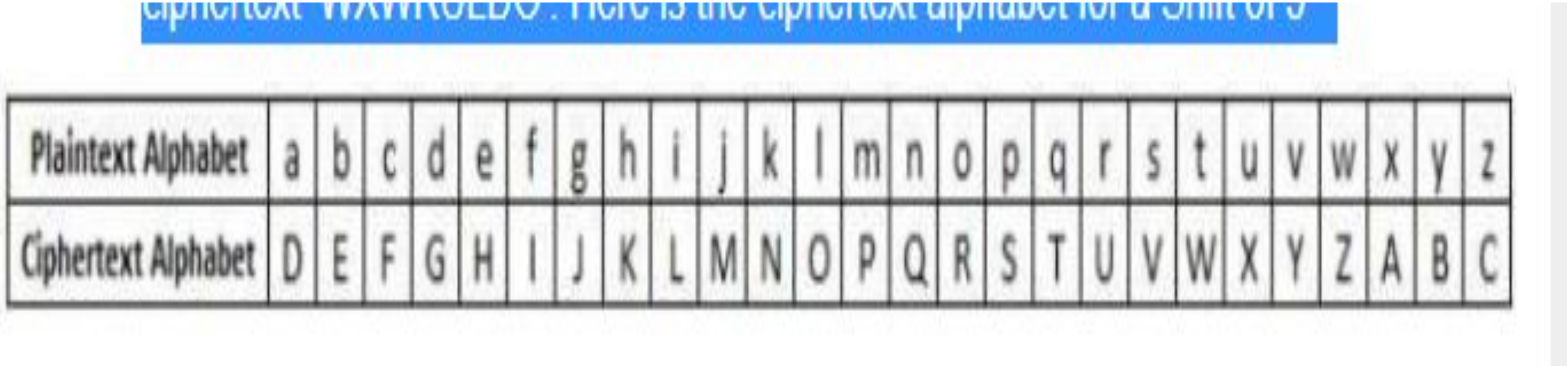
(Decryption Phase with shift n)





```
Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ

Shift: 23

Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW


Text : ATTACKATONCE

Shift: 4

Cipher: EXXEGOEXSRGI
```

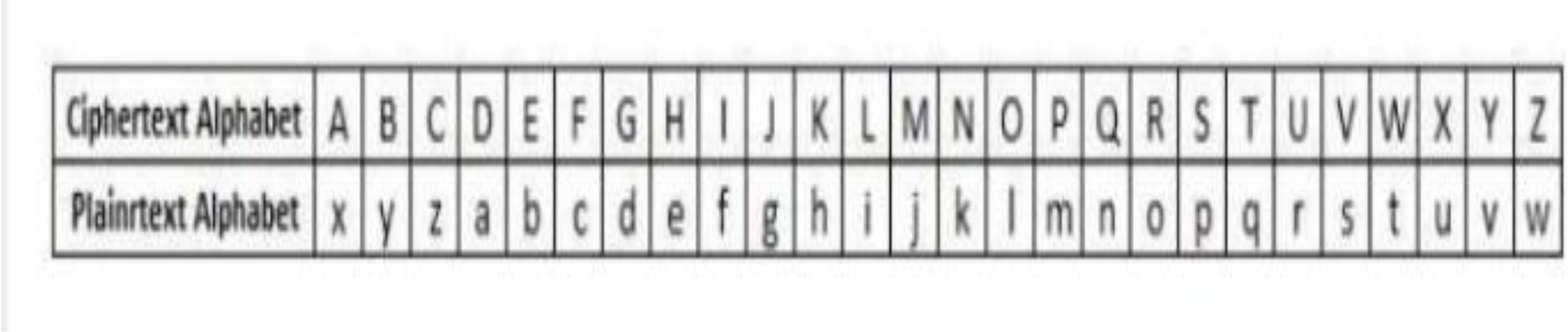
- In order to encrypt a plaintext letter, the sender positions the sliding ruler underneath the first set of plaintext letters and slides it to LEFT by the number of positions of the secret shift.
- The plaintext letter is then encrypted to the ciphertext letter on the sliding ruler underneath. The result of this process is depicted in the following illustration for an agreed shift of three positions. In this case, the plaintext ‘tutorial’ is encrypted to the ciphertext ‘WXWRULDO’. Here is the ciphertext alphabet for a Shift of 3 −



ciphertext ‘WXWRULDO’. Here is the ciphertext alphabet for a Shift of 3

Plaintext Alphabet	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 Alphabet	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

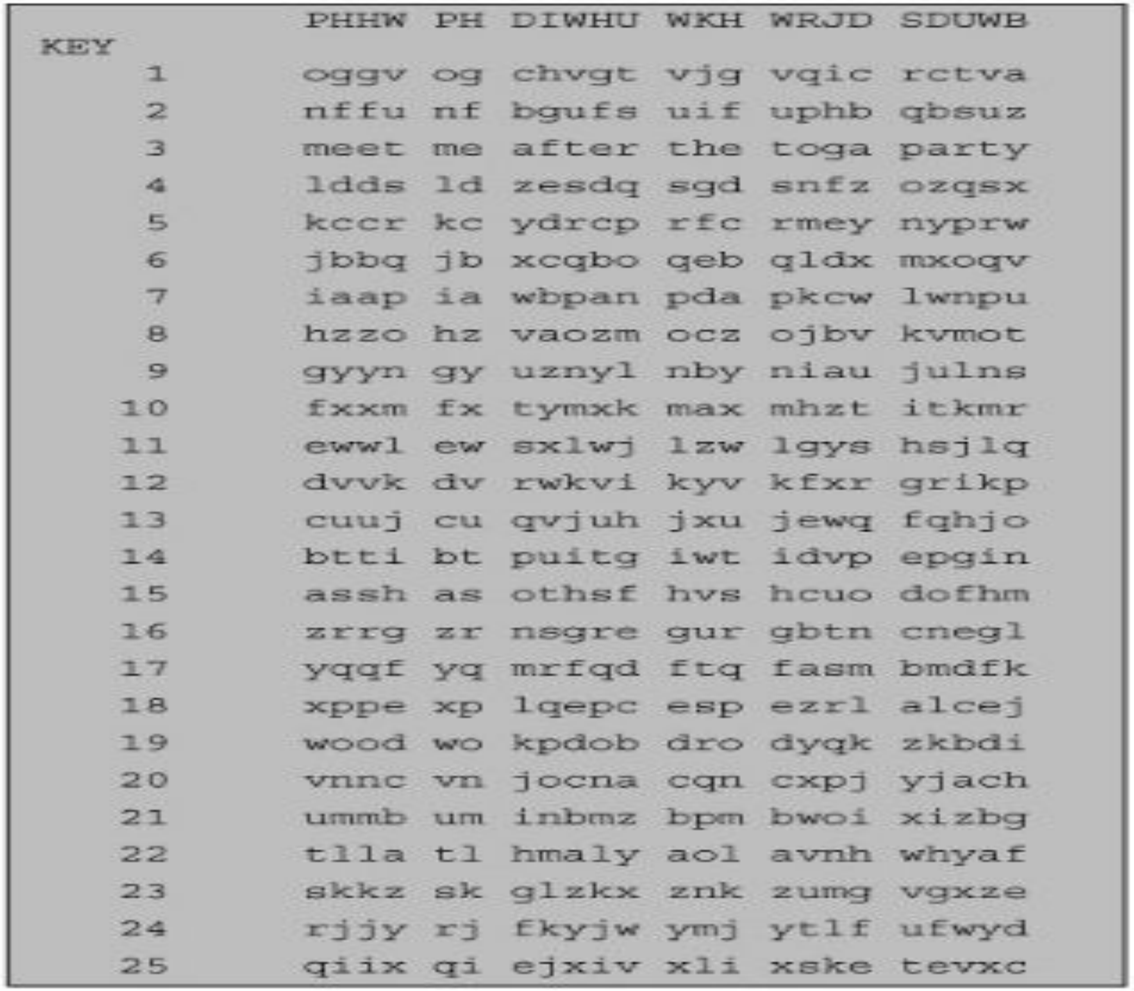
- On receiving the ciphertext, the receiver who also knows the secret shift, positions his sliding ruler underneath the ciphertext alphabet and slides it to RIGHT by the agreed shift number, 3 in this case.
- He then replaces the ciphertext letter by the plaintext letter on the sliding ruler underneath. Hence the ciphertext 'WXWRULDO' is decrypted to 'tutorial'. To decrypt a message encoded with a Shift of 3, generate the plaintext alphabet using a shift of '-3' as shown below −



Ciphertext Alphabet	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
Plainrtext Alphabet	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

- If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed:
- Simply try all the 25 possible keys. Figure 2.3 shows the results of applying this strategy to the example ciphertext. In this case, the plaintext leaps out as occupying the third line.

- **Figure 2.3. Brute-Force Cryptanalysis of Caesar Cipher**



	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
KEY						
1	oggv	og	chvgt	vjg	vqic	rctva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kccr	kc	ydrcp	rfc	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	mxoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaozm	ocz	ojbv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlq
12	dvvk	dv	rwkvi	kyv	kfxr	grikp
13	cuuj	cu	qvjuh	jxu	jewq	fqhjo
14	btti	bt	puitg	iwt	idvp	epgin
15	assh	as	othsf	hvs	hcuo	dofhm
16	zrrg	zr	nsgre	gur	gbtn	cnegl
17	yqqf	yq	mrfqd	ftq	fasm	bmdfk
18	xppe	xp	lqepc	esp	ezrl	alcej
19	wood	wo	kpdob	dro	dyqk	zkbdi
20	vnnc	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzkx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxc

- Three important characteristics of this problem enabled us to use a brute-force cryptanalysis:

1. The encryption and decryption algorithms are known.

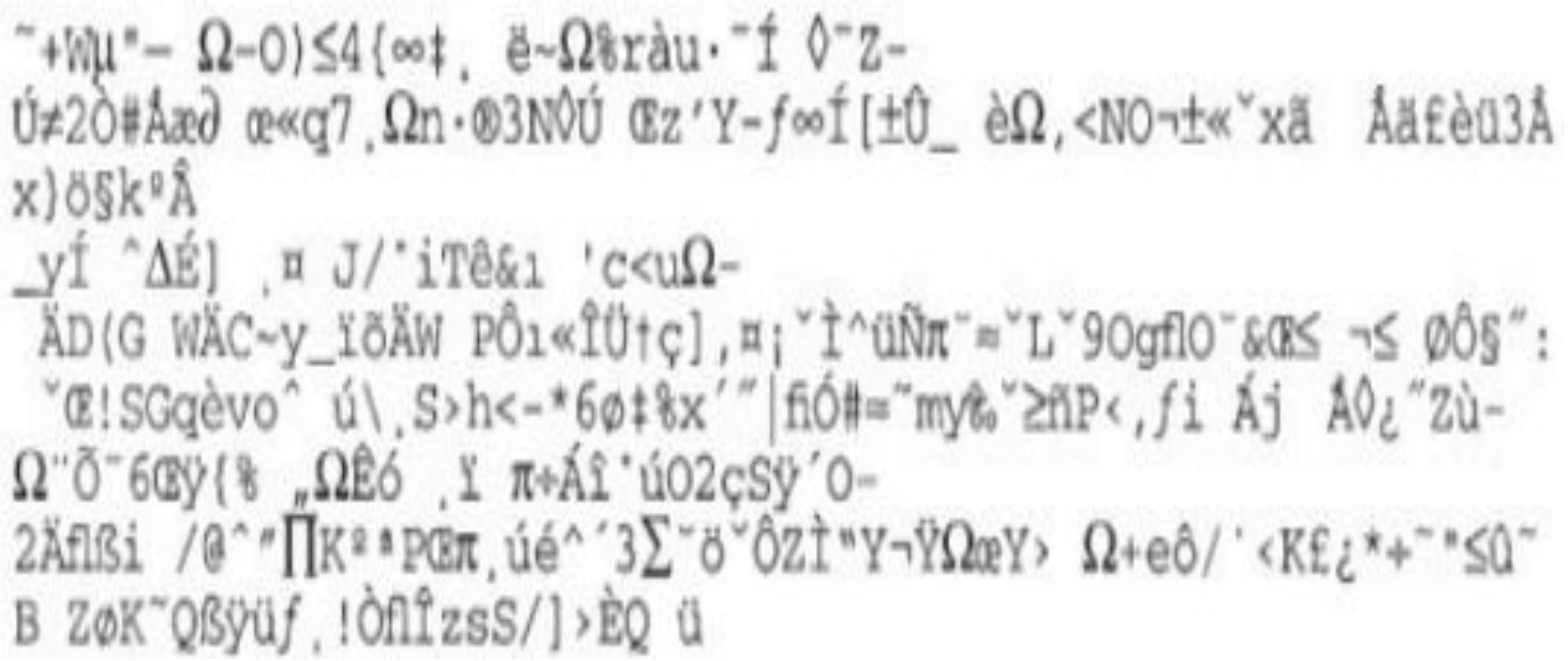
2. There are only 25 keys to try.

3. The language of the plaintext is known and easily recognizable.

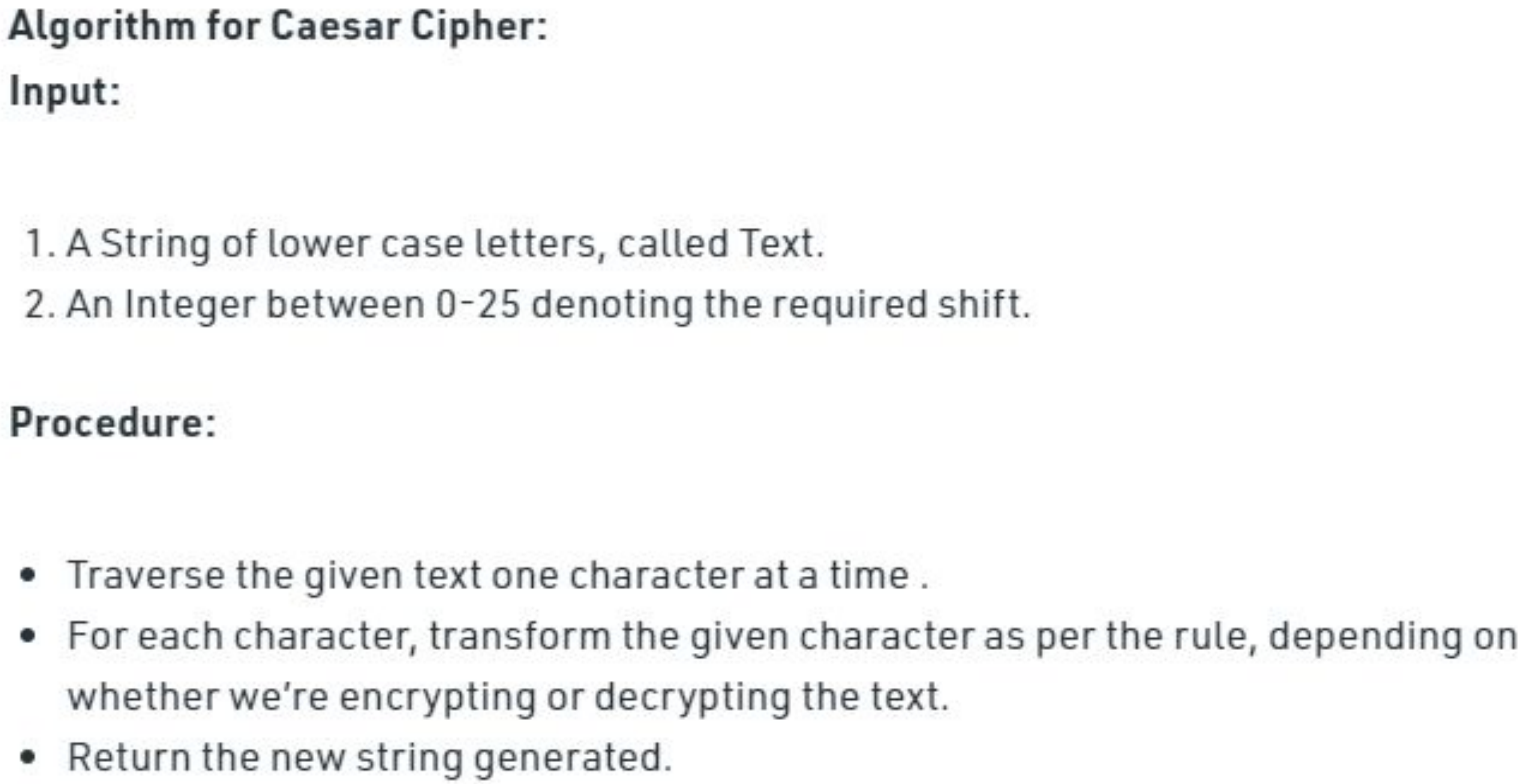
- In most networking situations, we can assume that the algorithms are known. What generally makes brute- force cryptanalysis impractical is the use of an algorithm that employs a large number of keys. For example, the triple DES algorithm, makes use of a 168-bit key, giving a key space of 2^{168} or greater than 3.7 x 10^{50} possible keys.

- The third characteristic is also significant. If the language of the plaintext is unknown, then plaintext output may not be recognizable. Furthermore, the input may be abbreviated or compressed in some fashion, again making recognition difficult. For example, Figure 2.4 shows a portion of a text file compressed using an algorithm called ZIP. If this file is then encrypted with a simple substitution cipher (expanded to include more than just 26 alphabetic characters), then the plaintext may not be recognized when it is uncovered in the brute-force cryptanalysis.

- **Figure 2.4. Sample of Compressed Text**



```
˜+Wµ"– Ω–O)≤4{∞‡¸ ë~Ω‰ràu·¯Í ◊¯Z–
Ú≠2Ò#Åæ∂ œ«q7¸Ωn·®3NÒÚ Œz′Y–ƒ∞Í[±Û_ èΩ,<NO¬±«˘xã  ÅäƒèÜ3Å
x)ö§kºÂ
_yÍ ˆ∆É] ¸¤ J/˙iTê&ı 'c<uΩ–
 ÄD(G WÄC~y_ÏõÄW PÔı«ÎÜ†ç],¤¡˘Ì^üÑπ˘=˘L˘9Ogﬂ0¯&Œ≤ ¬≤ ØÔ§″:
 ˘Œ!SGqèvoˆ ú\¸S›h<–*6ø‡%x′″|ﬁÓ#=˜my‰˘≥ñP‹,ƒi Áj  Å◊¿″Zù–
Ω¨Õ¯6Œÿ{% „ΩÊó ¸Ÿ π+ÁÎ˙úO2çSÿ´O–
2Äﬂßi /@ˆ″∏Kº*PŒπ¸úé^´3∑˜ö˘ÔZÌ"Y¬ŸΩœY› Ω+eô/˙‹K£¿*+¯"≤Û˜
B ZøK˜Qßÿüƒ¸!ÒﬂÎzsS/]›ÈQ ü
```



Algorithm for Caesar Cipher:

Input:

1. A String of lower case letters, called Text.
2. An Integer between 0-25 denoting the required shift.

Procedure:

- Traverse the given text one character at a time .
- For each character, transform the given character as per the rule, depending on whether we're encrypting or decrypting the text.
- Return the new string generated.

Monoalphabetic Ciphers

- With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. Recall the assignment for the Caesar cipher:
  - plain: 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
  - cipher: 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
- If, instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 x 10^{26} possible keys. This is 10 orders of magnitude greater than the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a monoalphabetic substitution cipher, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.
- There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext (e.g., non compressed English text), then the analyst can exploit the regularities of the language. To see how such a cryptanalysis might proceed, we give a partial example here that is adapted from one in [SINK66]. The ciphertext to be solved is
- UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ

  VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX

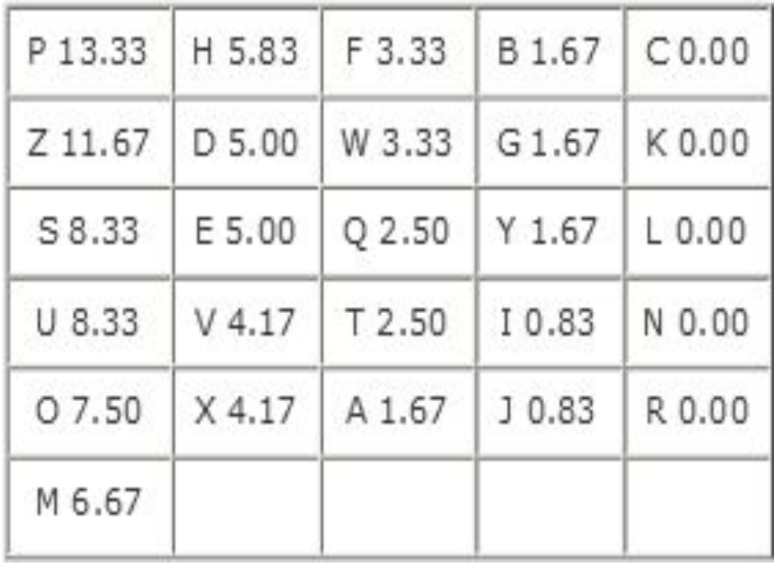
- Before proceeding, we define the term *permutation*. A **permutation** of a finite set of elements *S*is an ordered sequence of all the elements of *S*, with each element appearing exactly once. For example, if *S* = {a, b, c}, there are six permutations of *S*:

  abc, acb, bac, bca, cab, cba

- In general, there are *n*! permutations of a set of *n* elements, because the first element can be chosen in one of *n* ways, the second in *n* - 1 ways, the third in *n* – 2 ways, and so on

- Monoalphabetic cipher is a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process. For example, if ‘A’ is encrypted as ‘D’, for any number of occurrence in that plaintext, ‘A’ will always get encrypted to ‘D’.

- As a first step, the relative frequency of the letters can be determined and compared to a standard frequency distribution for English, such as is shown in Figure 2.5 (based on [LEWA00]).
- If the message were long enough, this technique alone might be sufficient, but because this is a relatively short message, we cannot expect an exact match. In any case, the relative frequencies of the letters in the ciphertext (in percentages) are as follows:



P 13.33	H 5.83	F 3.33	B 1.67	C 0.00
Z 11.67	D 5.00	W 3.33	G 1.67	K 0.00
S 8.33	E 5.00	Q 2.50	Y 1.67	L 0.00
U 8.33	V 4.17	T 2.50	I 0.83	N 0.00
O 7.50	X 4.17	A 1.67	J 0.83	R 0.00
M 6.67				

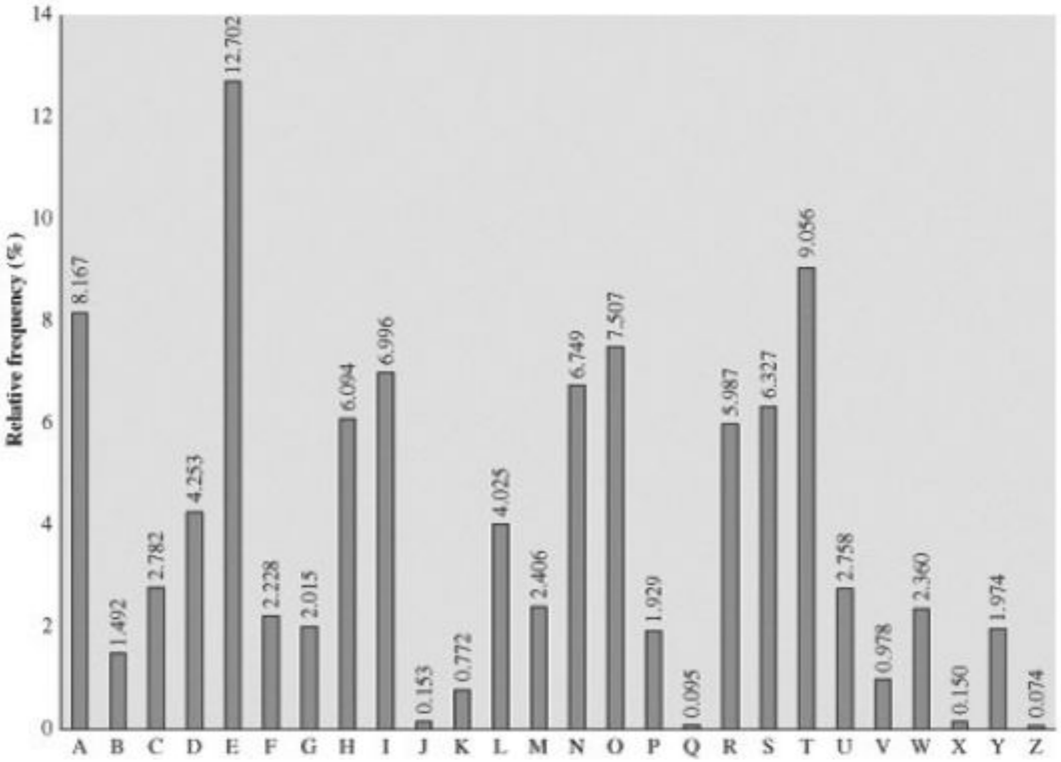


Figure 2.5. Relative Frequency of Letters in English Text

- Comparing this breakdown with Figure 2.5, it seems likely that cipher letters P and Z are the equivalents of plain letters e and t, but it is not certain which is which. The letters S,U,O,M,andHareallofrelatively
high frequency and probably correspond to plain letters from the set {a, h, i, n, o, r, s}.The letters with the lowest frequencies (namely, A, B, G, Y, I, J) are likely included in the set {b, j, k, q, v, x, z}.

- There are a number of ways to proceed at this point. We could make some tentative assignments and start to fill in the plaintext to see if it looks like a reasonable "skeleton" of a message. A more systematic approach is to look for other regularities. For example, certain words may be known to be in the text. Or we could look for repeating sequences of cipher letters and try to deduce their plaintext equivalents.

- A powerful tool is to look at the frequency of two-letter combinations, known as digrams. A table similar to Figure 2.5 could be drawn up showing the relative frequency of digrams. The most common such digram is th. In our ciphertext, the most common digram is ZW, which appears three times. So we make the correspondence of Z with t and W with h.

- Then, by our earlier hypothesis, we can equate P with e. Now notice that the sequence ZWP appears in the ciphertext, and we can translate that sequence as "the." This is the most frequent trigram (three-letter combination) in English, which seems to indicate that we are on the right track.
- Next, notice the sequence ZWSZ in the first line. We do not know that these four letters form a complete word, but if they do, it is of the form th_t. If so, S equates with a.
- So far, then, we
- UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
- t a e e te a that e e aa
- VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
- e t ta t ha e ee a e th t a
  EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
- e ee tate thet

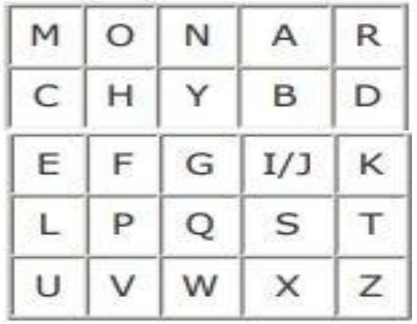
- Only four letters have been identified, but already we have quite a bit of the message. Continued analysis of frequencies plus trial and error should easily yield a solution from this point. The complete plaintext, with spaces added between words, follows:
  - *it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow*

- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet. A countermeasure is to provide multiple substitutes, known as homophones, for a single letter. For example, the letter e could be assigned a number of different cipher symbols, such as 16, 74, 35, and 21, with each homophone used in rotation, or randomly. If the number of symbols assigned to each letter is proportional to the relative frequency of that letter, then single-letter frequency information is completely obliterated.
- The great mathematician Carl Friedrich Gauss believed that he had devised an unbreakable cipher using homophones. However, even with homophones, each element of plaintext affects only one element of ciphertext, and multiple-letter patterns (e.g., digram frequencies) still survive in the ciphertext, making cryptanalysis relatively straightforward.
-

Two principal methods are used in substitution ciphers to lessen the extent to which the structure of the plaintext survives in the ciphertext: One approach is to encrypt multiple letters of plaintext, and the other is to use multiple cipher alphabets. We briefly examine each.

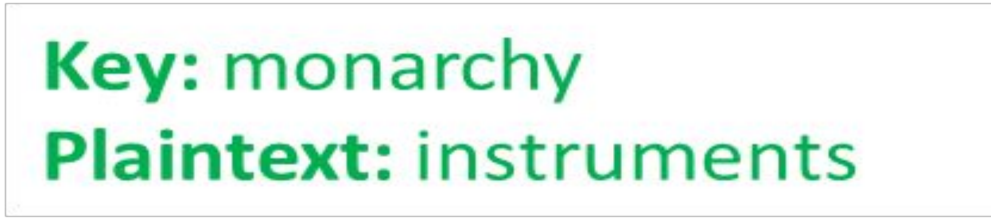
Playfair Cipher

- The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams.
- The Playfair algorithm is based on the use of a 5 x 5 matrix of letters constructed using a keyword. Here is an example, solved by Lord Peter Wimsey in Dorothy Sayers's Have His Carcase:



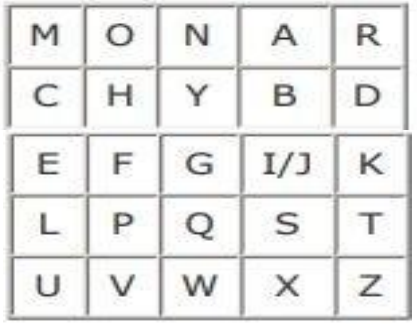
M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- In playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.
- Encryption Technique
- For the encryption process let us consider the following example:



Key: monarchy

Plaintext: instruments



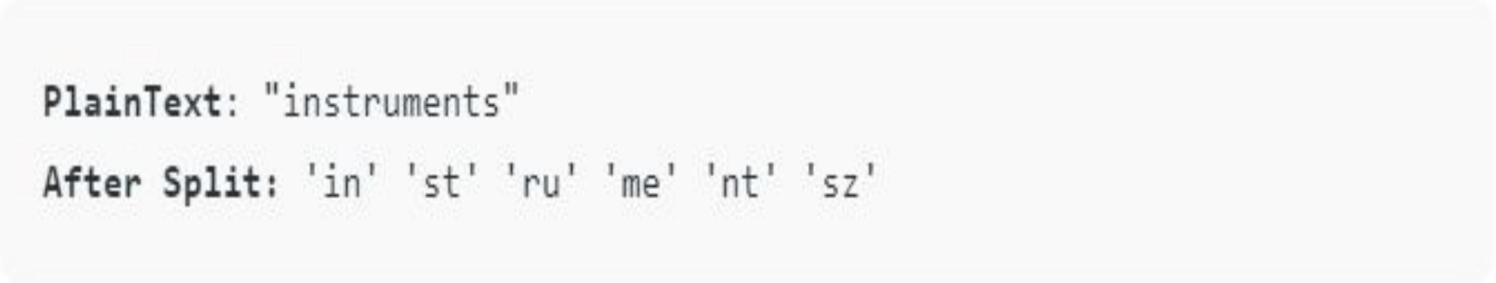
M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- **The Playfair Cipher Encryption Algorithm:** The Algorithm consists of 2 steps:

1. **Generate the key Square(5×5):**
   1. The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.

   2. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

- **2.Algorithm to encrypt the plain text:** The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.
  For example:



```
PlainText: "instruments"

After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'
```

1. Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.

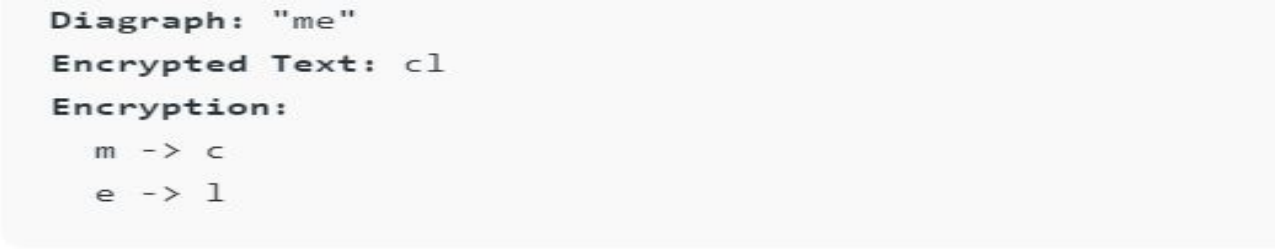
- **Plain Text:** “hello”
- **After Split:** ‘he’ ‘lx’ ‘lo’
- Here **‘x’** is the bogus letter.

2. If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter

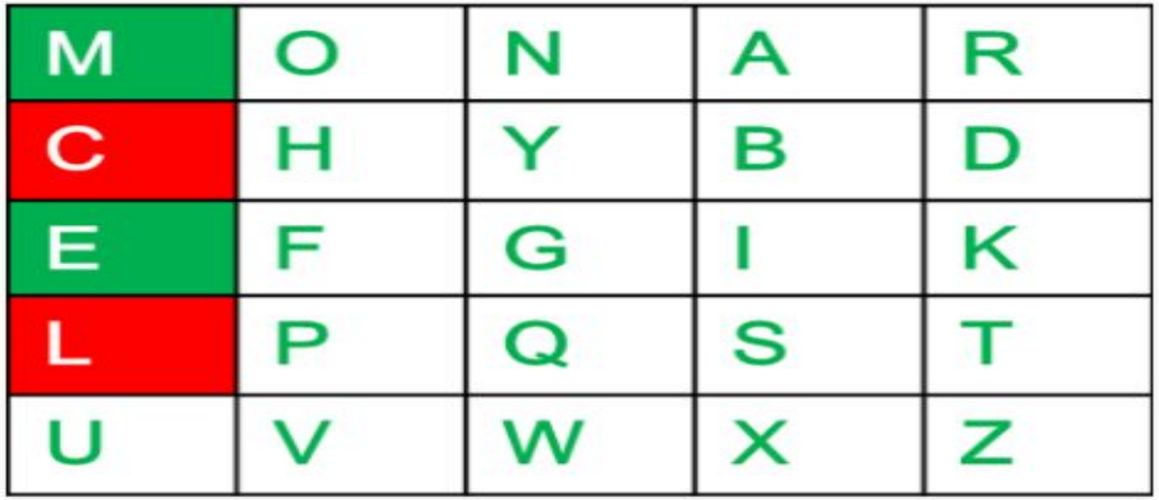
- **Plain Text:** “helloe”
- **AfterSplit:** ‘he’ ‘lx’ ‘lo’ ‘ez’
- Here **‘z’** is the bogus letter.

if both letters are the same (or only one letter is left), add an "X" after the first letter. Encrypt the new pair and continue. Some variants of Playfair use "Q" instead of "X", but any letter, itself uncommon as a repeated pair, will do.

- **Rules for Encryption:**
- **If both the letters are in the same column**: Take the letter below each one (going back to the top if at the bottom).
  For example:

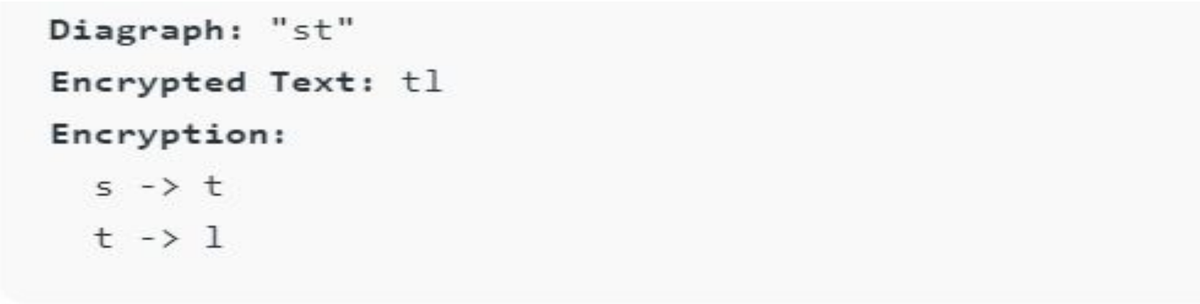


```
Diagraph: "me"
Encrypted Text: cl
Encryption:
  m -> c
  e -> l
```

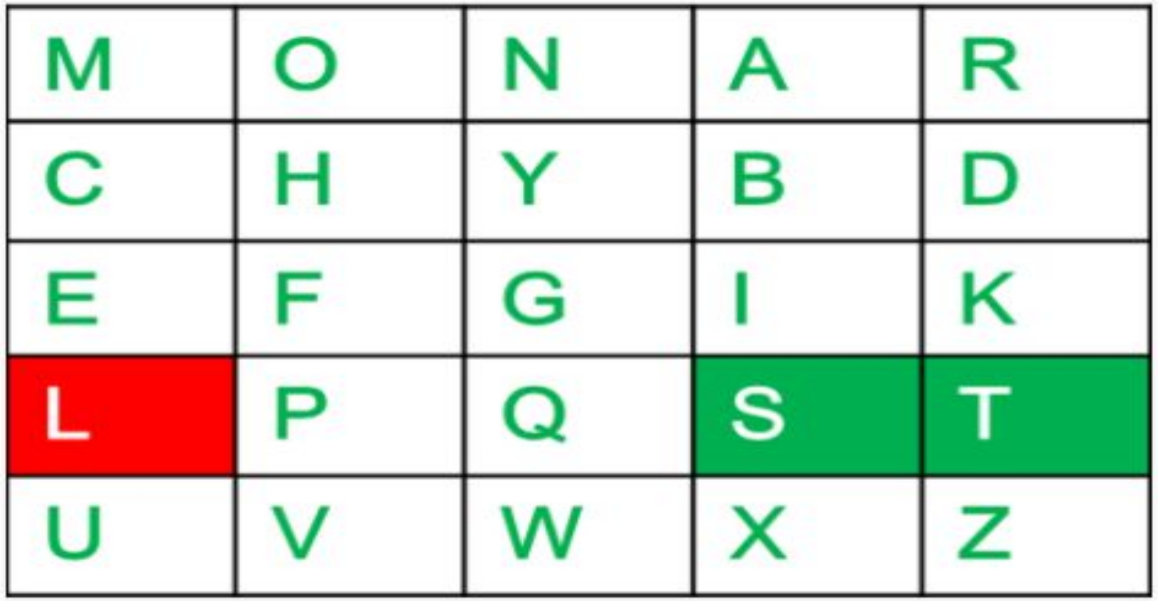


M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **If both the letters are in the same row**: Take the letter to the right of each one (going back to the leftmost if at the rightmost position).
  For example:

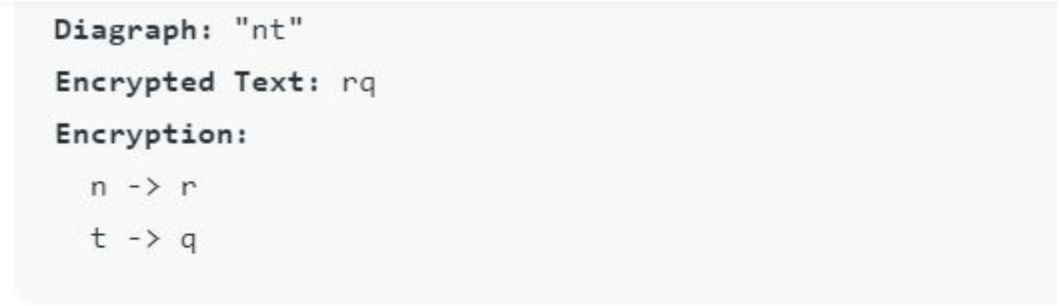


```
Diagraph: "st"
Encrypted Text: tl
Encryption:
  s -> t
  t -> l
```



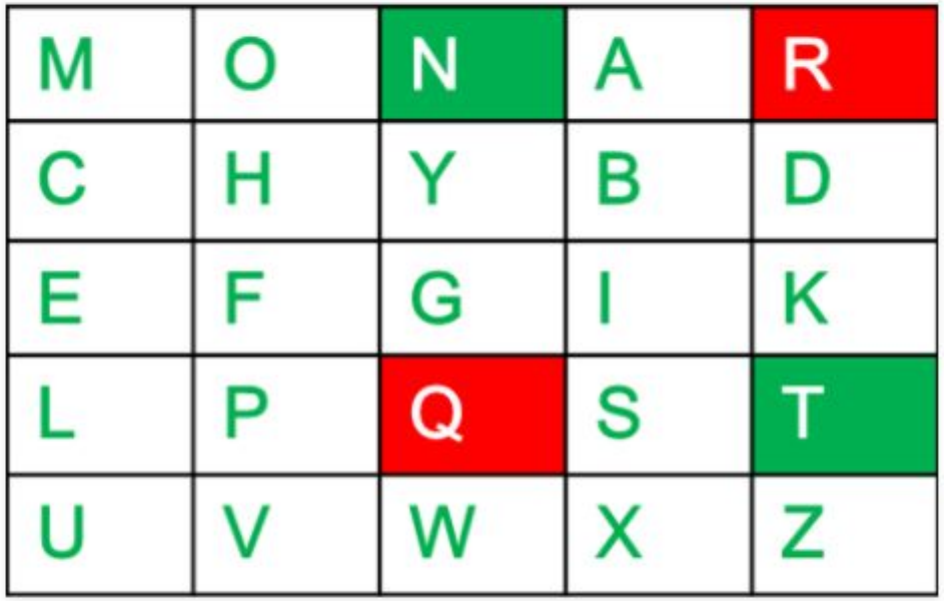
M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

- **If neither of the above rules is true**: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.
  For example:

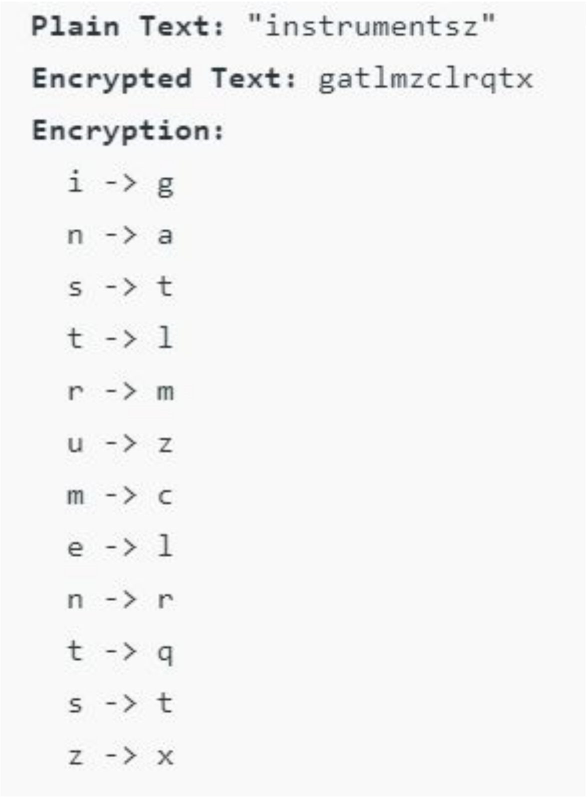


```
Diagraph: "nt"

Encrypted Text: rq

Encryption:

  n -> r

  t -> q
```

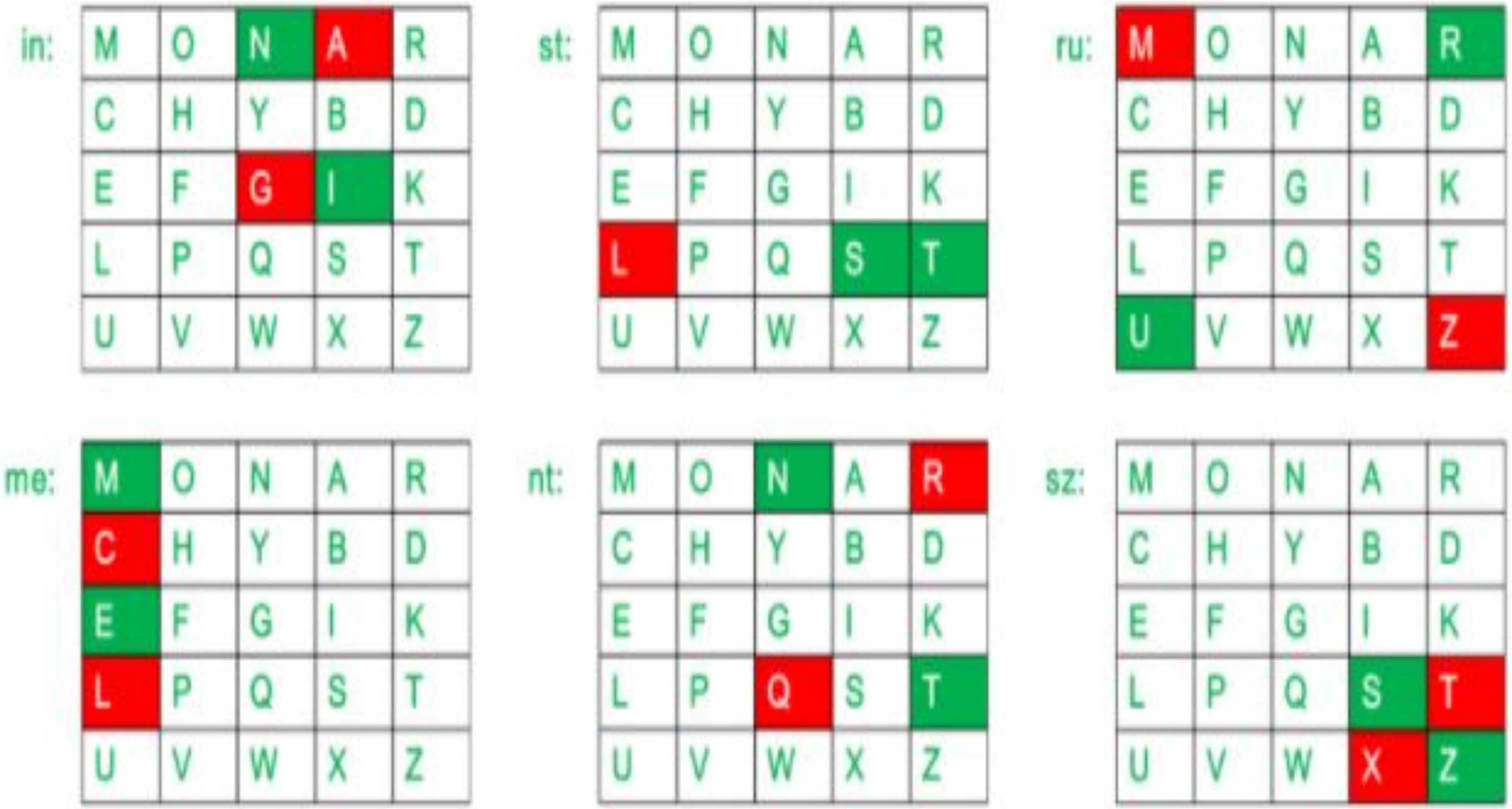
•



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



```
Plain Text: "instrumentsz"
Encrypted Text: gatlmzclrqtx
Encryption:
  i -> g
  n -> a
  s -> t
  t -> l
  r -> m
  u -> z
  m -> c
  e -> l
  n -> r
  t -> q
  s -> t
  z -> x
```



in:

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

st:

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

ru:

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

me:

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

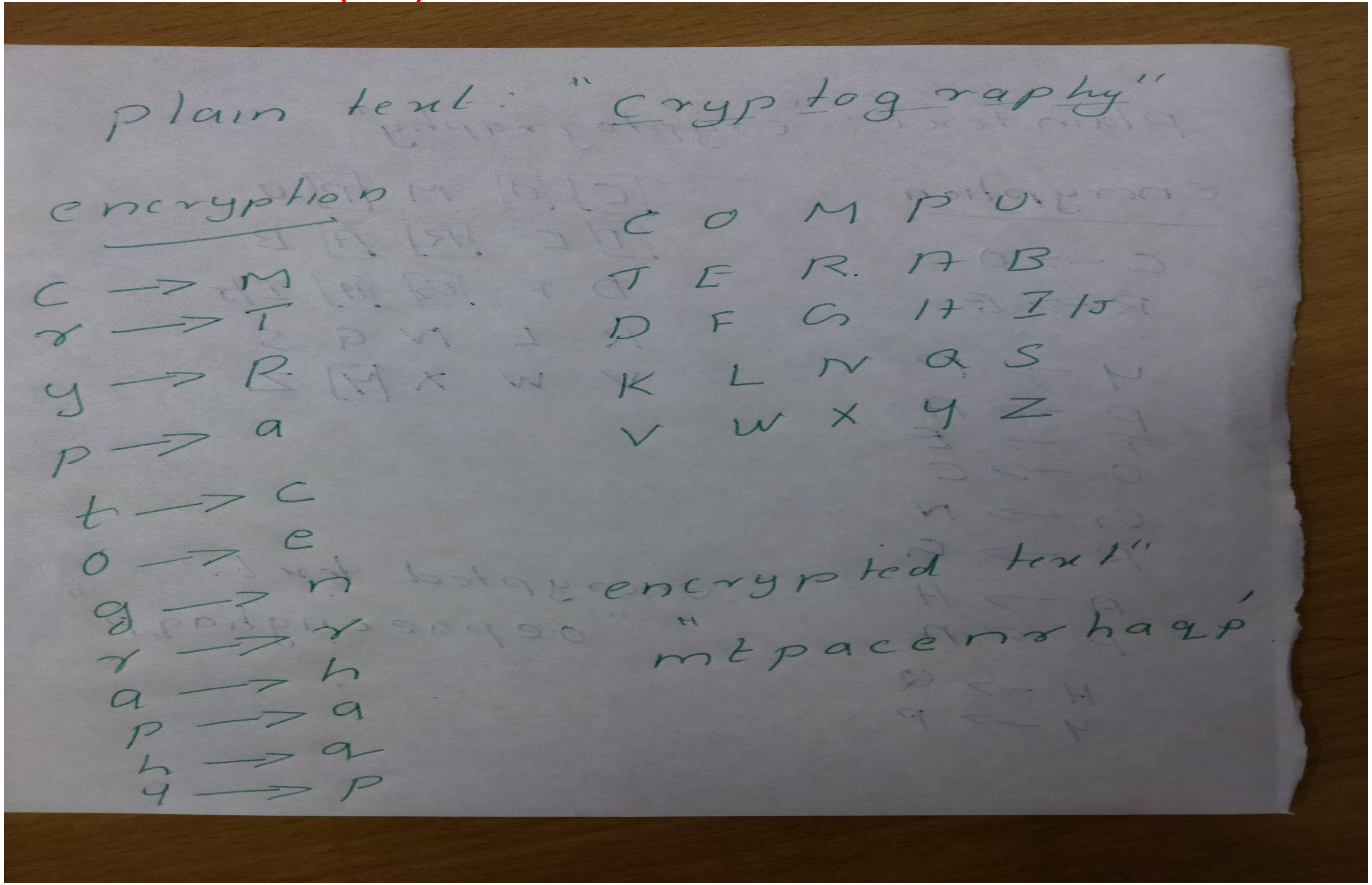
nt:

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

sz:

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

Q1.Use Playfair Cipher with key COMPUTER to encrypt the message “CRYPTOGRAPHY”. (ktu)



plain text: "cryptography"

encryption

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

c → M
r → T
y → R
p → a
t → c
o → e
g → n
r → r
a → h
p → a
h → q
y → p

encrypted text"

"mtpacenrhaqp'

- In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time, according to the following rules:

1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo x on.
2. 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. For example, ar is encrypted as RM.
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. For example, mu is encrypted as CM.
4. 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. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

- The Playfair cipher is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 x 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipher was for a long time
-

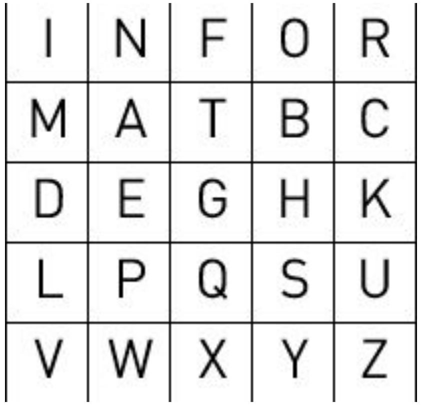
considered unbreakable. It was used as the standard field system by the British Army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

- Despite this level of confidence in its security, the Playfair cipher is relatively easy to break because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

- One way of revealing the effectiveness of the Playfair and other ciphers is shown in Figure 2.6, based on [SIMM93]. The line labeled plaintext plots the frequency distribution of the more than 70,000 alphabetic characters in the Encyclopaedia Brittanica article on cryptology.[5] This is also the frequency distribution of any monoalphabetic substitution cipher. The plot was developed in the following way: The number of occurrences of each letter in the text was counted and divided by the number of occurrences of the letter e (the most frequently used letter). As a result, e has a relative frequency of 1, t of about 0.76, and so on. The points on the horizontal axis correspond to the letters in order of decreasing frequency.

- Q1.Suppose we have a plain text ‘**Balloon**’ we need to convert it to cipher text.
  First lets choose our secret key and make key table

1. Lets choose secret key as ‘INFORMATION’
2. The key table will be



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

3. From left to right starting from the top left box we inserted the alphabets according to our secret key INFORMATION. We did not insert the duplicate alphabets again e.g. after T of information I,O,N is not inserted since it is already in the table. Then the missing alphabets are inserted accordingly. J is ommitted.

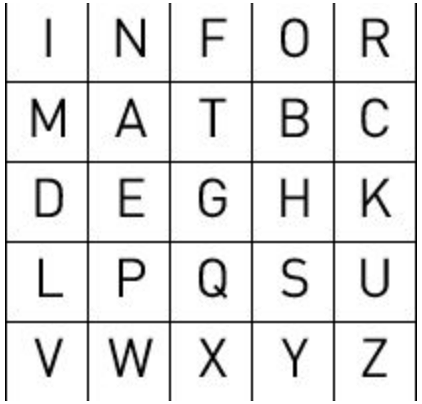
- 4. Now lets make the plain text pair wise BA | LL | OO | N
  we have duplicate alphabets in a pair so we insert X in it
  BA | LX | LO | ON.
- 5. For **BA** they are in the same row so it will become **CT**
  BA→ CT
- 6. For LX they are diagonally so L will get Q and X will get V meaning the rectangles opposite corners.
  LX→ QV
- 7. LO are also diagonally so
  LO→SI
- 8. ON is row wise so it will be RF
- **ON→RF**
- Finally the cipher text of BALLOON is CTQVSIRF

Q2.Lets take another example ‘**KZTQACISW**’ as plain text

- we will use the same key table

1. Pair wise KZ | TQ | AC | IS | WZ
   notice we added Z at the end to make it even
2. KZ is column wise so K->U and Z->R (rotates)
   KZ→UR
3. TQ is column
   TQ→GX
4. AC is row
   AC→TM
5. IS is diagonal
   IS→OL
6. WZ is row
   WZ→XV

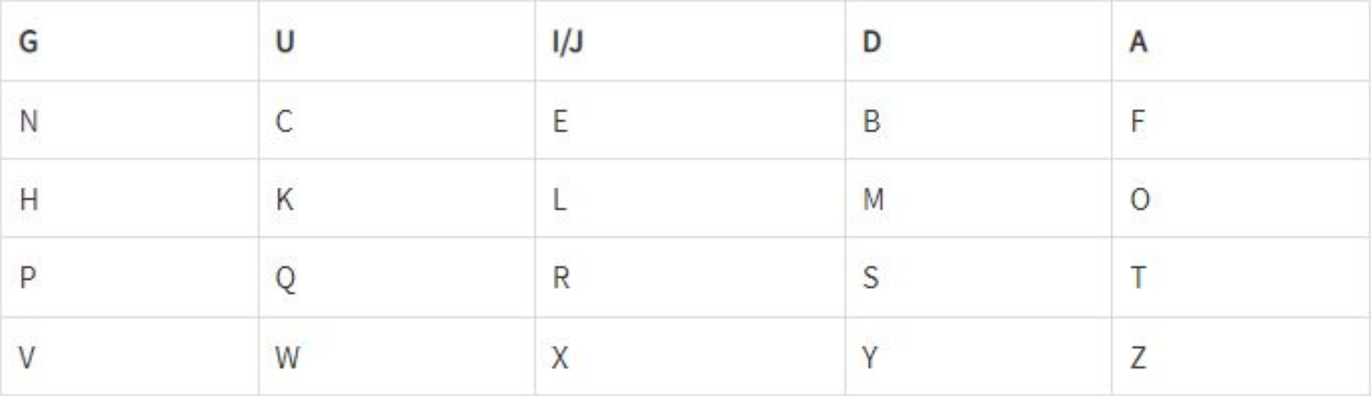
- **KZTQACISW → URGXTMOLXV**



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

Q3.Encrypt "This is the final exam" with Playfair cipher using key "Guidance". Explain the steps involved.

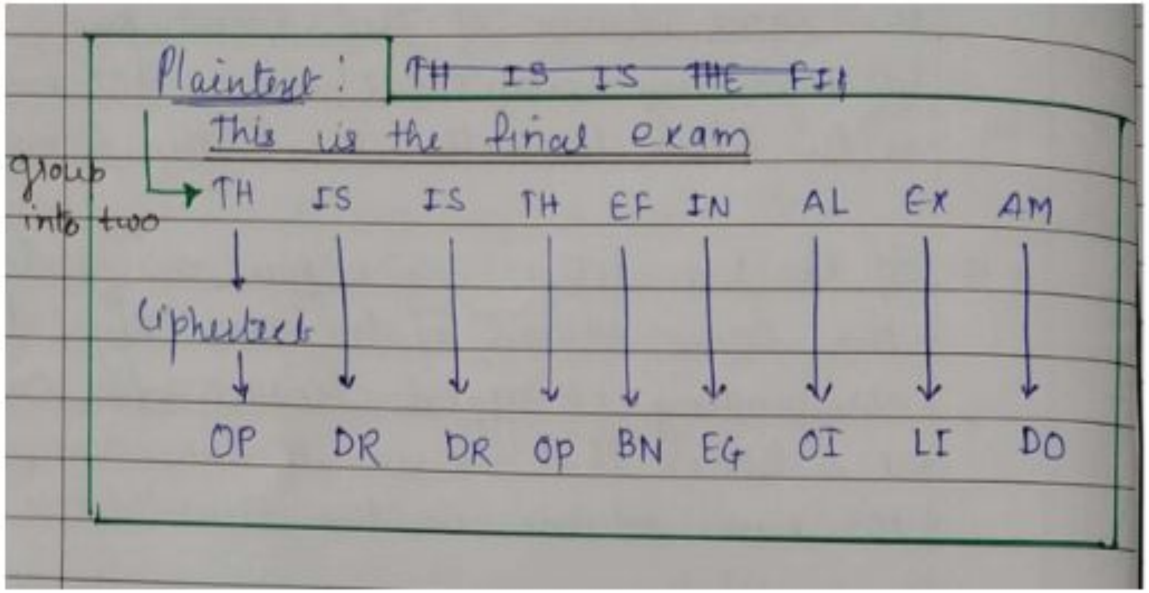
- The secret key in playfair cipher is made of 25 alphabet letters arranged in a 5 X 5 matrix in which I & J are considered the same when encrypting
  - Different arrangements of the letters in the matrix can create many different secret keys



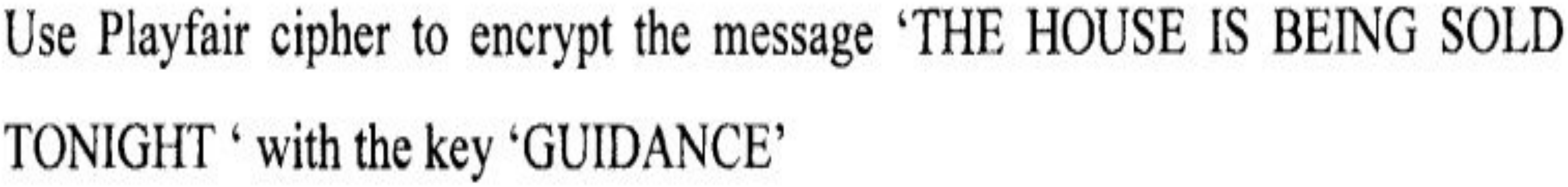
G	**U**	**I/J**	**D**	**A**
N	C	E	B	F
H	K	L	M	O
P	Q	R	S	T
V	W	X	Y	Z

- The cipher uses three rules of encryption.
- If the two letters in a pair are located in the same row f the secret key , the corresponding encrypted character for each letter is the next letter to the right in the same row.
- If the two letters in a pair are located in the same column of the secret key, the corresponding encrypted characters for each letter is the letter beneath in the same column.
- If the two letters in a pair are hot located in the same column or row of the secret key, the corresponding encrypted characters for each letter is a letter that is in its own row but in the same columns as the other letter.

- plaintext: " This is the final exam "
- cipher : th is th ef in al ex am
- Ciphertent by using all rules
- 'Th' = OP ..... by rule 3
- 'is' = DR ...... by rule 3
- 'ef = BN ..... by rule 2
- 'in' = FG ..... by rule 3
- 'al' = OI ..... by rule 3
- 'ex' = LI ...... by rule 2
- 'am' = DO ..... by rule 3



Questions



Use Playfair cipher to encrypt the message ‘THE HOUSE IS BEING SOLD TONIGHT ‘ with the key ‘GUIDANCE’

• Polyalphabetic Ciphers

•Another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is polyalphabetic substitution cipher. All these techniques have the following features in common:

1. A set of related monoalphabetic substitution rules is used.

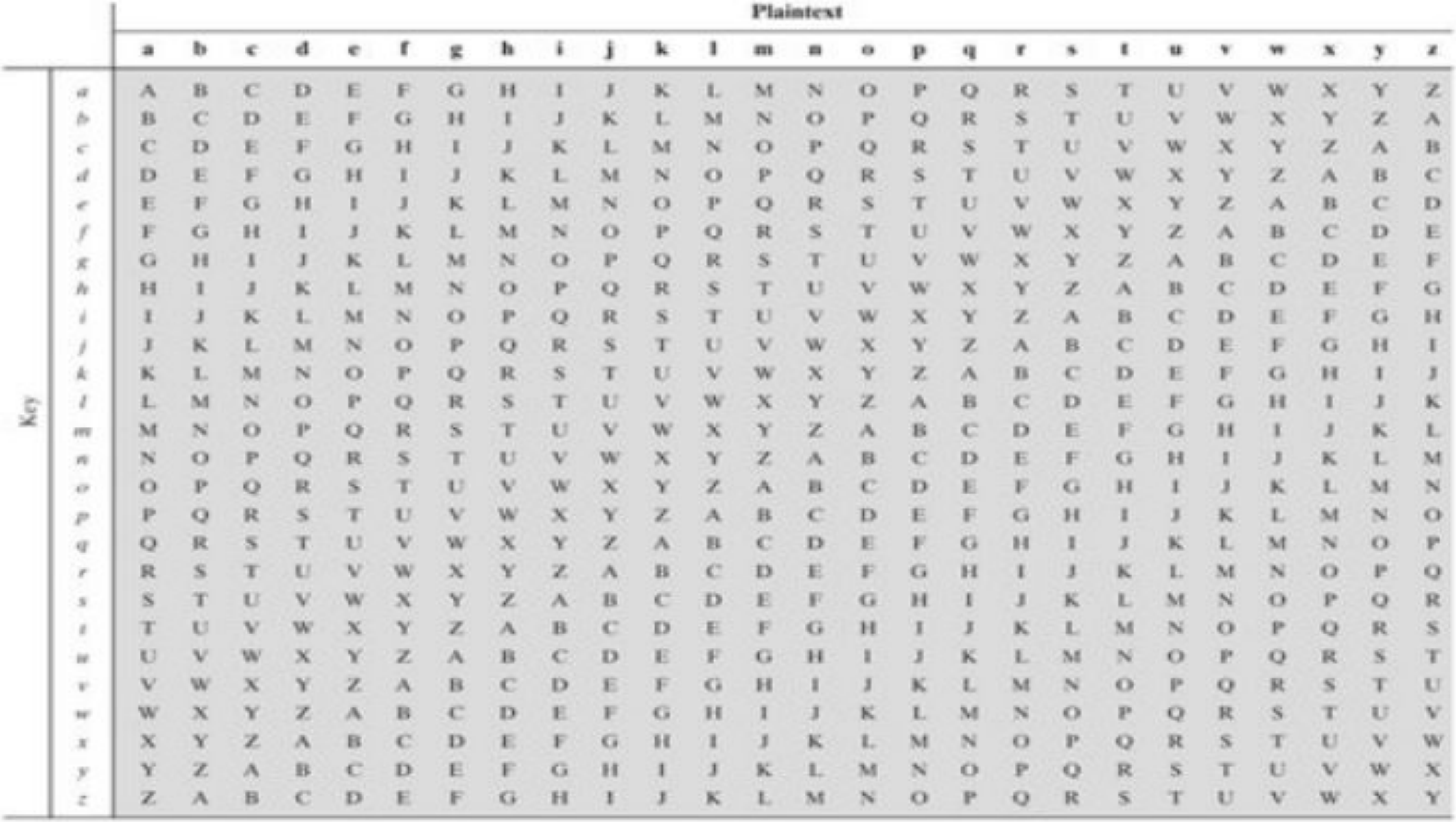
2.A key determines which particular rule is chosen for a given transformation.

•The best known, and one of the simplest, such algorithm is referred to as the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25.

•Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value d.

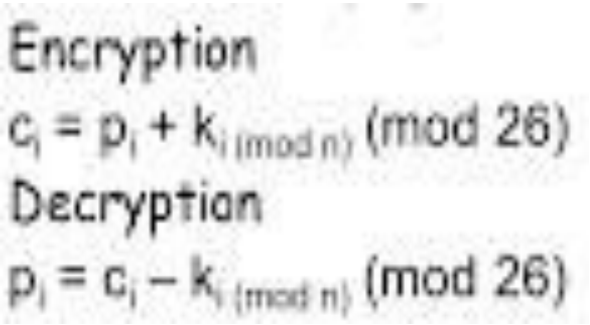
•To aid in understanding the scheme and to aid in its use, a matrix known as the Vigenère tableau is constructed (Table 2.3). Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to its left. A normal alphabet for the plaintext runs across the top. The process of encryption is simple: Given a key letter x and a plaintext letter y, the ciphertext letter is at the intersection of the row labeled x and the column labeled y; in this case the ciphertext is V.

- **Table 2.3. The Modern Vigenère Tableau**



		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
Key	*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
	s	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

- **Vigenère Cipher**
  - simplest polyalphabetic substitution cipher
  - effectively multiple caesar ciphers
  - key is multiple letters long K = k1 k2 ... kd
  - ith letter specifies ith alphabet to use
  - use each alphabet in turn
  - repeat from start after „d“ letters in message
  - decryption simply works in reverse



Encryption

$c_i = p_i + k_{i \,(\mathrm{mod}\, n)} \pmod{26}$

Decryption

$p_i = c_i - k_{i \,(\mathrm{mod}\, n)} \pmod{26}$

Example of 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: deceptivedeceptivedeceptive
- plaintext: wearediscoveredsaveyourself
- ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ
- reverse process is for decryption

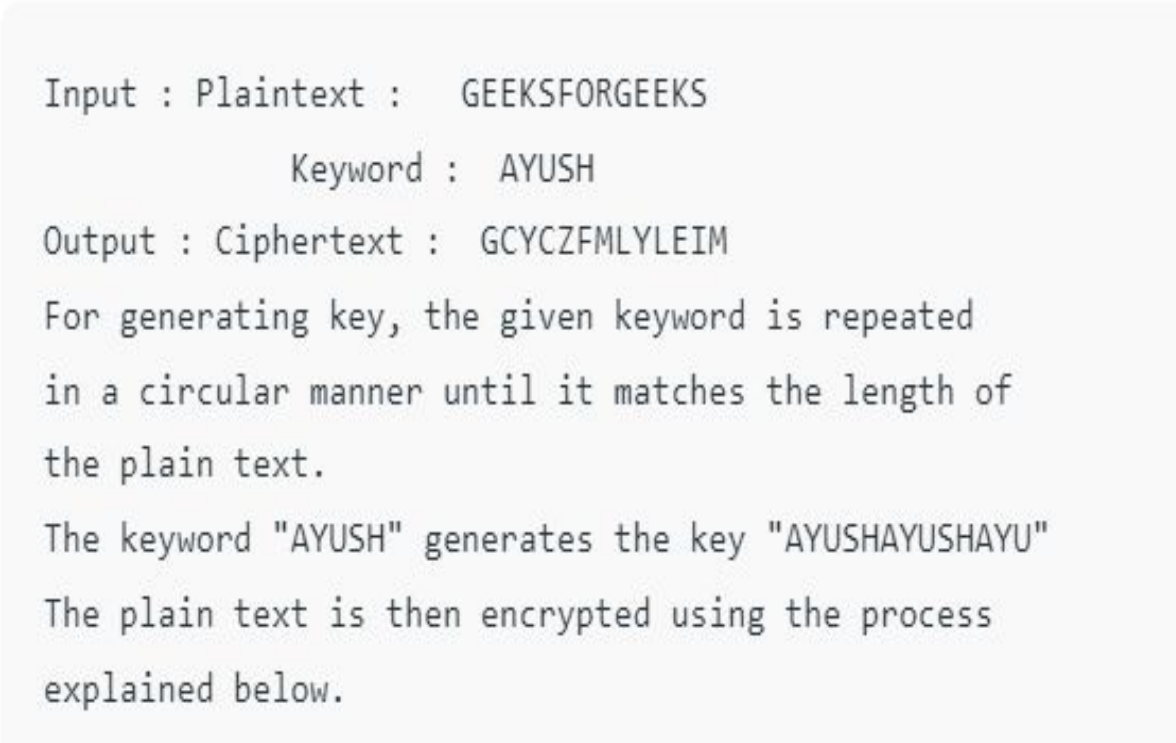
Q1.Encrypt the message "she is listening music" using Vigenere cipher with 'PASCAL' as the secret key

- Message = ‘SHE IS LISTENING MUSIC’

Secret key = ‘PASCAL’

Divide the message into blocks each with 5 letters. Repeat the secret key along with it. Therefore,
SHEIS LISTEN INGMU SIC = Plaintext
PASCA LPASC ALPAS CAL = Secret Key code
Create the Vignere table with rows and columns from ‘a’ to ‘z’.
Now to encrypt take the first letter of secret key code as column and plaintext letter as row. Find the alphabet in the cell which both row and column meets. Similarly repeat the step until all letters are encrypted.

Example:

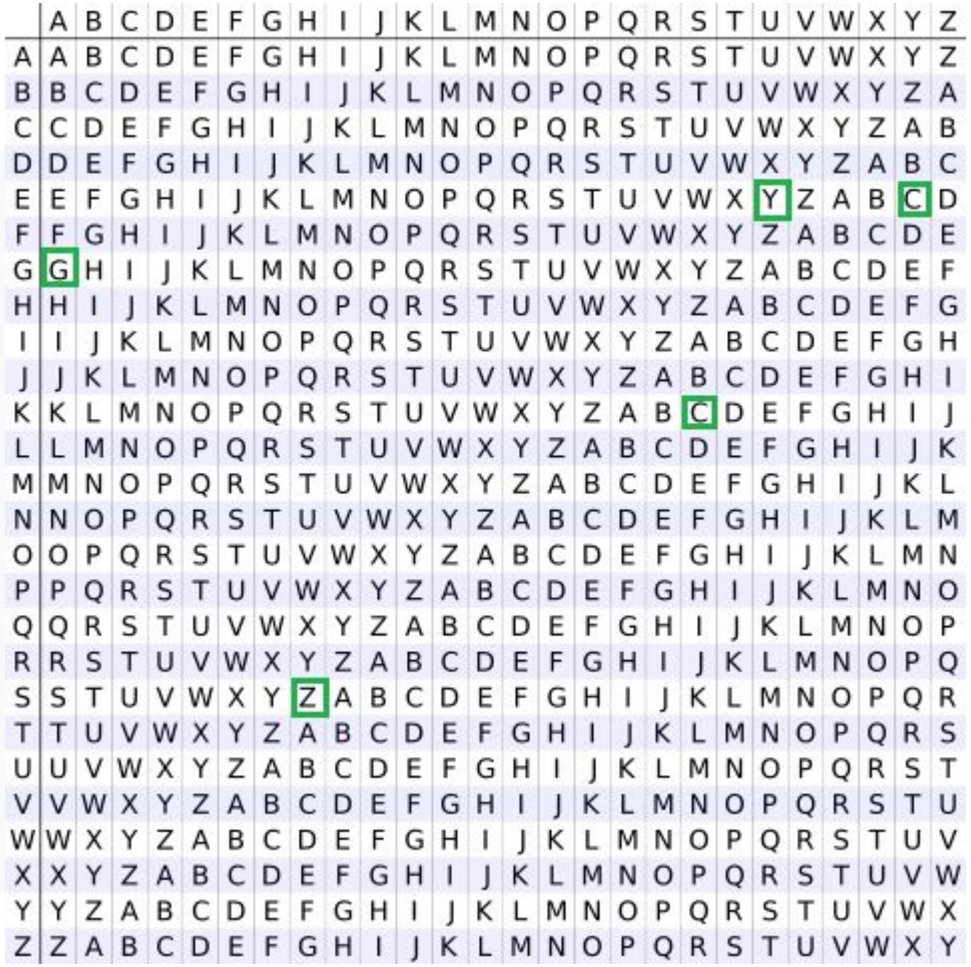


```
Input : Plaintext :   GEEKSFORGEEKS
             Keyword :  AYUSH
Output : Ciphertext :  GCYCZFMLYLEIM
```

For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.



	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
S	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

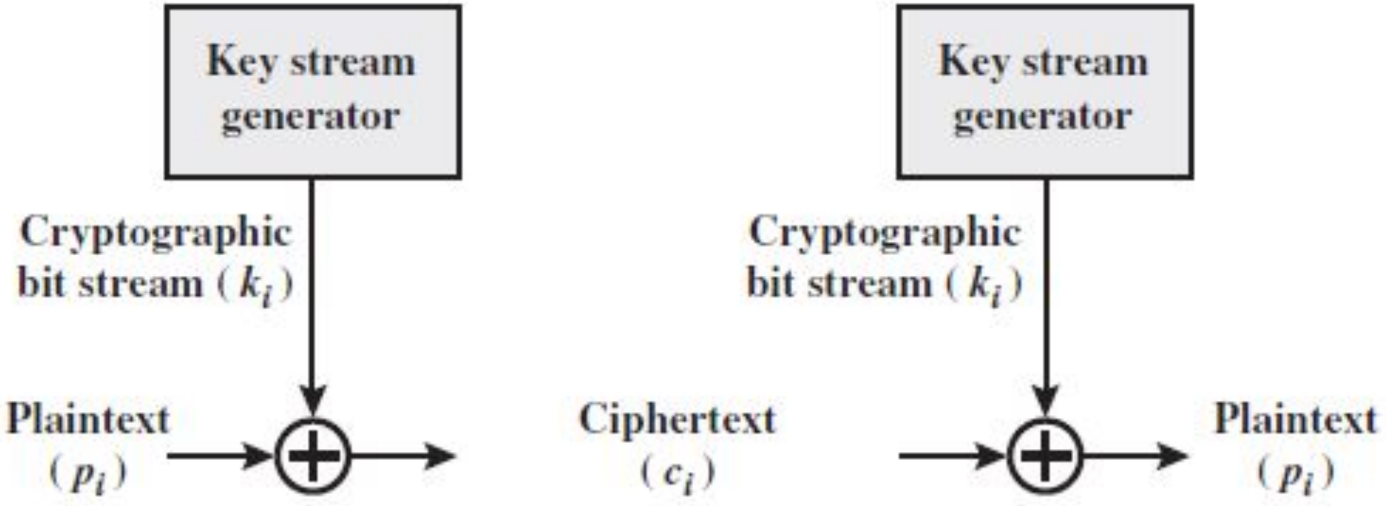
Encrypt the message **“the house is being sold tonight”** using Vigenere cipher with key “*dollars*”. Ignore the space between words. Decrypt the message to get the plain text.

- **Variants of Vigenere Cipher**
- There are two special cases of Vigenere cipher –

- The keyword length is same as plaintect message. This case is called **Vernam Cipher**. It is more secure than typical Vigenere cipher.
- Vigenere cipher becomes a cryptosystem with perfect secrecy, which is called **One-time pad**

Vernam Cipher

- *Encryption:* $c_i = p_i \oplus k_i$
- Decryption: $p_i = c_i \oplus k_i$



One-Time Pad

- An improvement to the Vernam cipher
- Using a random key that is as long as the message, so that the key need not be repeated.
- In addition, 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.
- Such a scheme, known as a **one-time pad, is unbreakable.**

• Hill Cipher

•Another interesting multi letter cipher is the Hill cipher, developed by the mathematician Lester Hill in 1929. The encryption algorithm takes m successive plaintext letters and substitutes for them m ciphertext letters. The substitution is determined by m linear equations in which each character is assigned a numerical value (a= 0, b = 1 ... z = 25). For m = 3, the system can be described as follows:

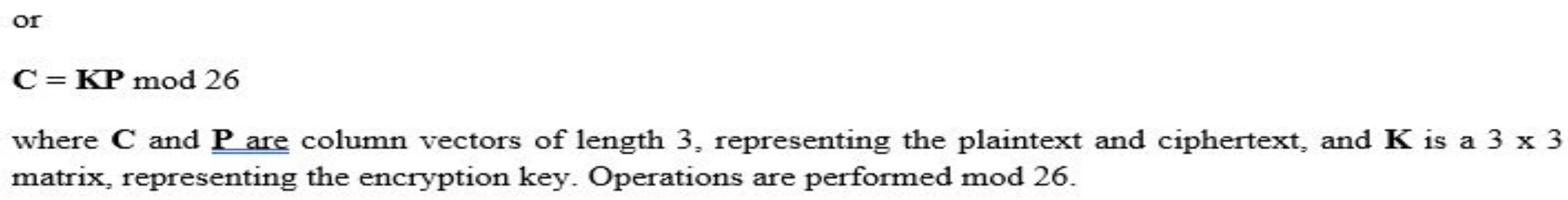
$$c_1 = (k_{11p1} + k_{12p2} + k_{13p3}) \bmod 26$$

$$c_2 = (k_{21p1} + k_{22p2} + k_{23p3}) \bmod 26$$

$$c_3 = (k_{31p1} + k_{32p2} + k3_{3p3}) \bmod 26$$

•This can be expressed in term of column vectors and matrices:

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



or

$\mathbf{C} = \mathbf{KP} \bmod 26$

where **C** and **P** are column vectors of length 3, representing the plaintext and ciphertext, and **K** is a 3 x 3 matrix, representing the encryption key. Operations are performed mod 26.

- For example, consider the plaintext "paymoremoney" and use the encryption key

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

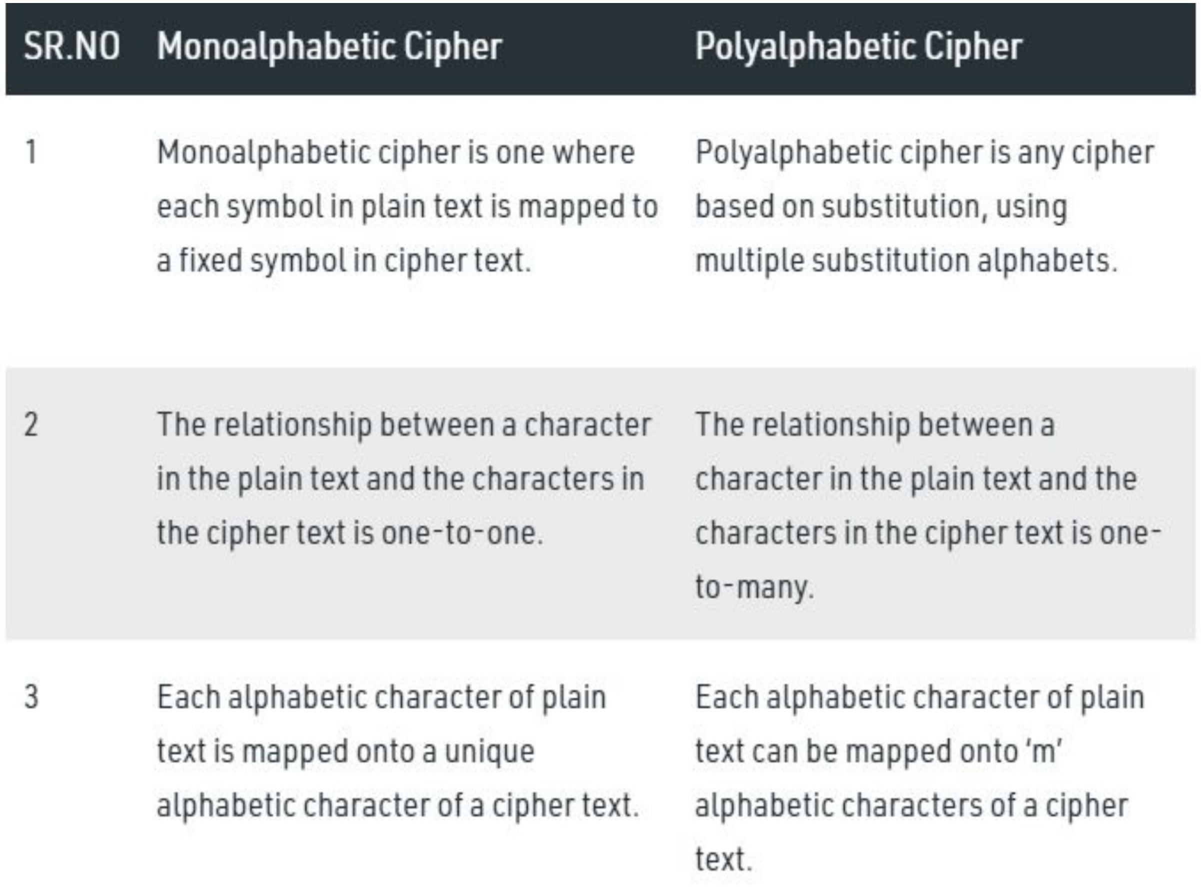
- The first three letters of the plaintext are represented by the vector

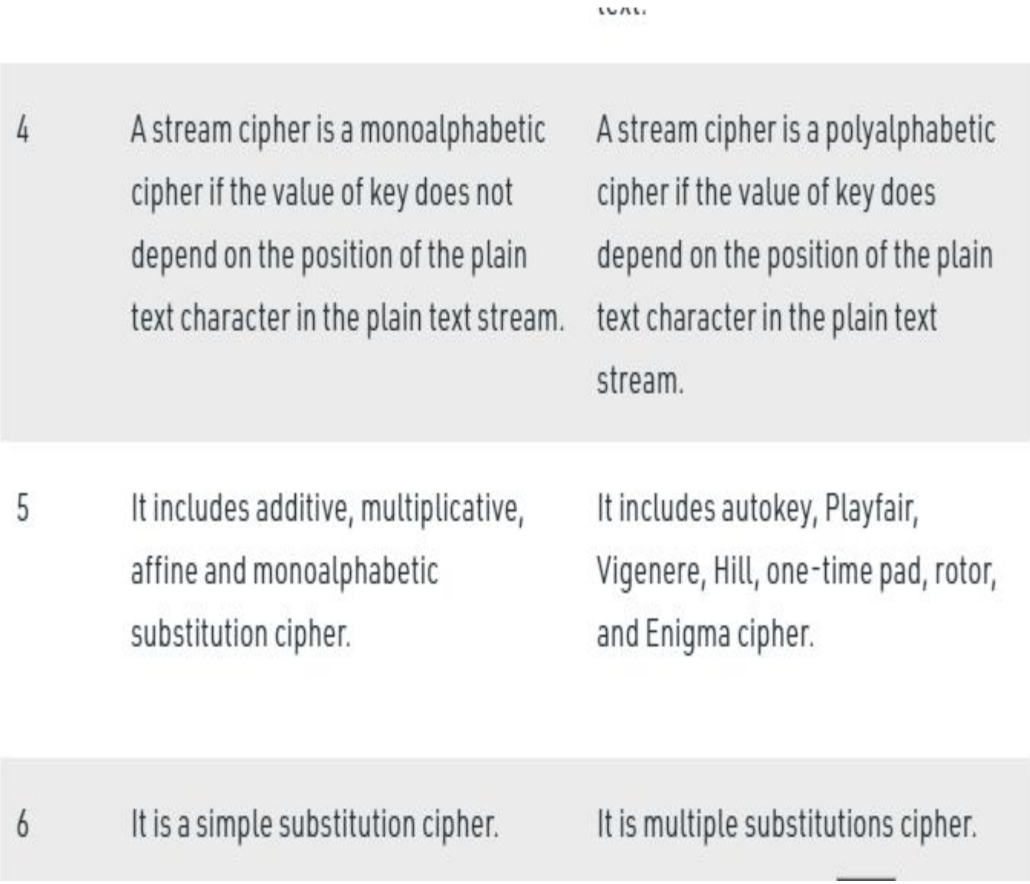
$$\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix}. \text{Then } \mathbf{K}\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS. Continuing in this fashion,}$$

- the ciphertext for the entire plaintext is LNSHDLEWMTRW.

Differiante between monoalphabetic and polyalphabetic ciphers with example (5)

Difference Between Monoalphabetic Cipher and Polyalphabetic Cipher :





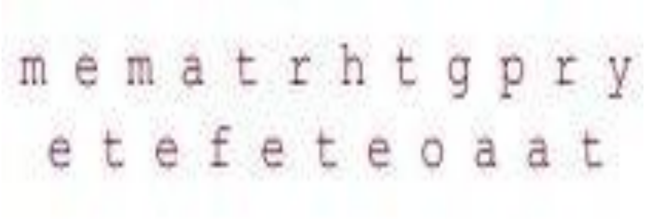
SR.NO	Monoalphabetic Cipher	Polyalphabetic Cipher
1	Monoalphabetic cipher is one where each symbol in plain text is mapped to a fixed symbol in cipher text.	Polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets.
2	The relationship between a character in the plain text and the characters in the cipher text is one-to-one.	The relationship between a character in the plain text and the characters in the cipher text is one-to-many.
3	Each alphabetic character of plain text is mapped onto a unique alphabetic character of a cipher text.	Each alphabetic character of plain text can be mapped onto 'm' alphabetic characters of a cipher text.
4	A stream cipher is a monoalphabetic cipher if the value of key does not depend on the position of the plain text character in the plain text stream.	A stream cipher is a polyalphabetic cipher if the value of key does depend on the position of the plain text character in the plain text stream.
5	It includes additive, multiplicative, affine and monoalphabetic substitution cipher.	It includes autokey, Playfair, Vigenere, Hill, one-time pad, rotor, and Enigma cipher.
6	It is a simple substitution cipher.	It is multiple substitutions cipher.

TRANSPOSITION TECHNIQUES

- All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher.
- These techniques hide the message by rearranging the letter order without *altering the actual letters*.

● The ciphertext has the same frequency distribution as the original plaintext

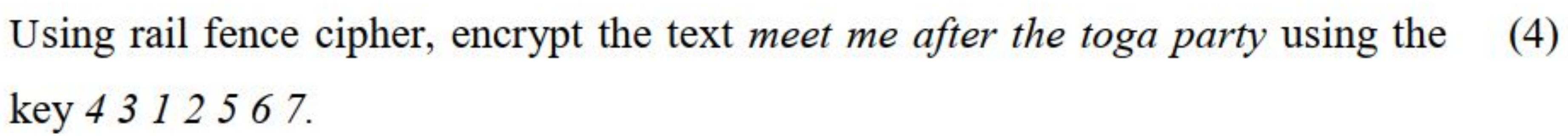
The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows. For example, to encipher the message "meet me after the toga party" with a rail fence of depth 2, we write the following:



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

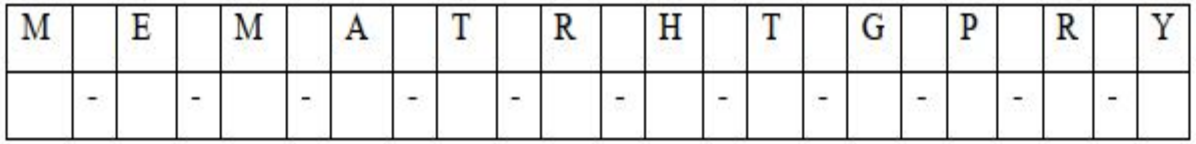
The encrypted message is

MEMATRHTGPRYETEFETEOAAT

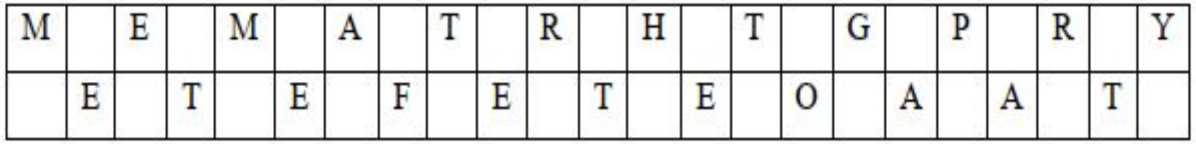


Using rail fence cipher, encrypt the text *meet me after the toga party* using the key *4 3 1 2 5 6 7*. (4)

- For Decryption:
  - ● Count the number of characters and make columns
  - ● And make number of rows = depth



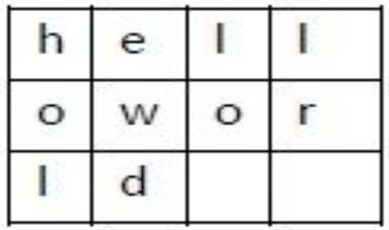
M		E		M		A		T		R		H		T		G		P		R		Y
	-		-		-		-		-		-		-		-		-		-		-	



M		E		M		A		T		R		H		T		G		P		R		Y
	E		T		E		F		E		T		E		O		A		A		T	

- Diagonally read the characters to obtain the plaintext
  - ● This sort of encryption would be trivial to cryptanalyst

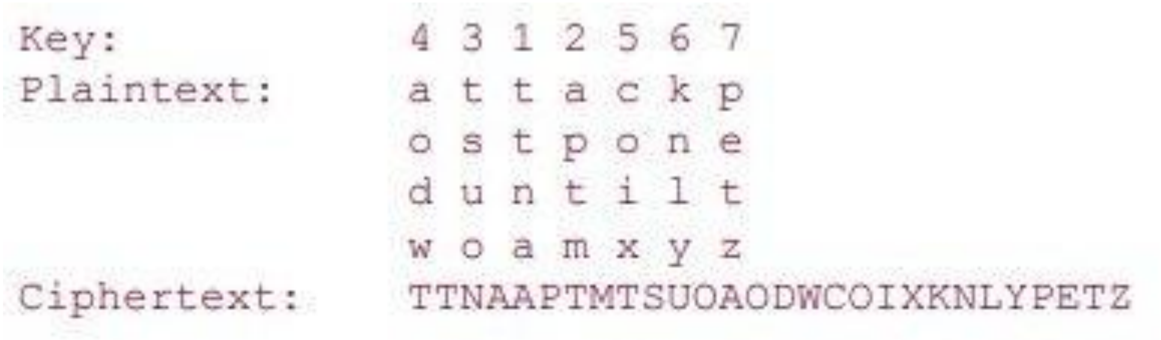
- Example
- A simple example for a transposition cipher is **columnar transposition cipher** where each character in the plain text is written horizontally with specified alphabet width. The cipher is written vertically, which creates an entirely different cipher text.
- Consider the plain text **hello world**, and let us apply the simple columnar transposition technique as shown below



h	e	l	l
o	w	o	r
l	d		

- The plain text characters are placed horizontally and the cipher text is created with vertical format as **: holewdlo lr.** Now, the receiver has to use the same table to decrypt the cipher text to plain text.

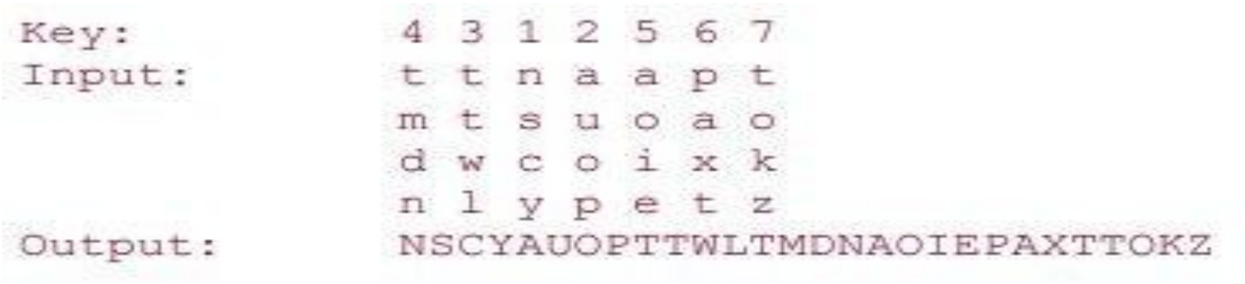
- This sort of thing would be trivial to cryptanalyze. A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm. For example,



```
Key:          4 3 1 2 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
```

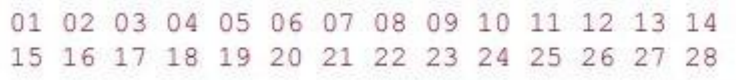
- A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext. For the type of columnar transposition just shown, cryptanalysis is fairly straightforward and involves laying out the ciphertext in a matrix and playing around with column positions. Digram and trigram frequency tables can be useful.

- The transposition cipher can be made significantly more secure by performing more than one stage of transposition. The result is a more complex permutation that is not easily reconstructed. Thus, if the foregoing message is reencrypted using the same algorithm,



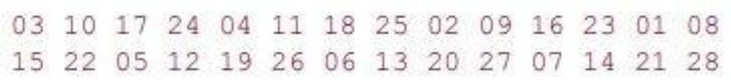
```
Key:        4 3 1 2 5 6 7
Input:      t t n a a p t
            m t s u o a o
            d w c o i x k
            n l y p e t z
Output:     NSCYAUOPTTWLTMDNAOIEPAXTTOKZ
```

- To visualize the result of this double transposition, designate the letters in the original plaintext message by the numbers designating their position. Thus, with 28 letters in the message, the original sequence of letters is



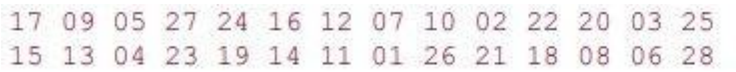
```
01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28
```

- After the first transposition we have



```
03 10 17 24 04 11 18 25 02 09 16 23 01 08
15 22 05 12 19 26 06 13 20 27 07 14 21 28
```

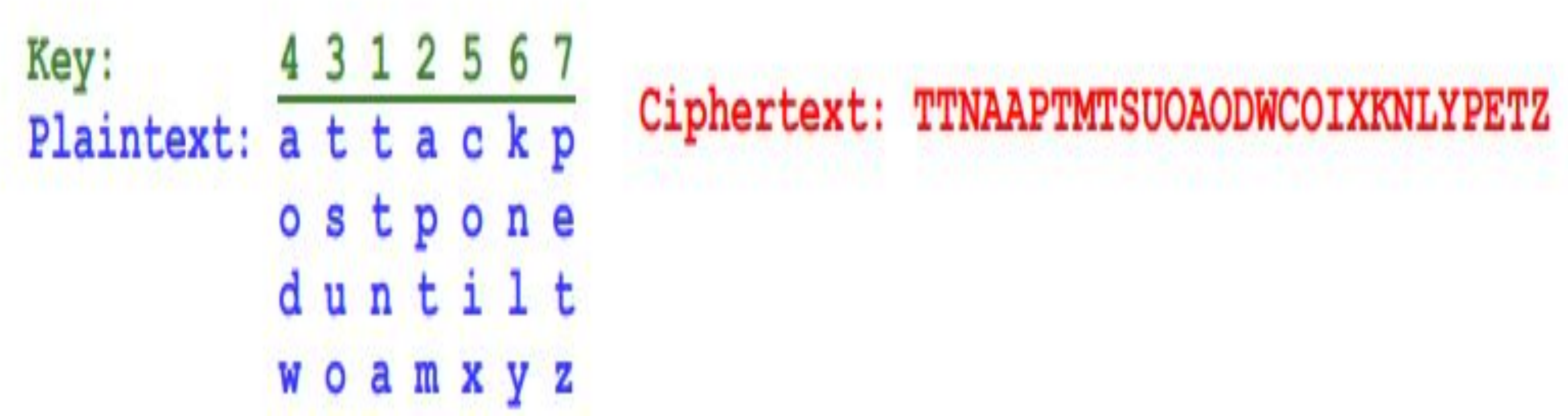
- which has a somewhat regular structure. But after the second transposition, we have



```
17 09 05 27 24 16 12 07 10 02 22 20 03 25
15 13 04 23 19 14 11 01 26 21 18 08 06 28
```

- This is a much less structured permutation and is much more difficult to cryptanalyze.

Example: “attack postponed until two am” with key 4312567: first read the column marked by 1, then the one marked by 2, etc.



Key:	4	3	1	2	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

ROTOR MACHINES

- The example just given suggests that multiple stages of encryption can produce an algorithm that is significantly more difficult to cryptanalyze. This is as true of substitution ciphers as it is of transposition ciphers. Before the introduction of DES, the most important application of the principle of multiple stages of encryption was a class of systems known as rotor machines.

- The basic principle of the rotor machine is illustrated in Figure 2.7. The machine consists of a set of independently rotating cylinders through which electrical pulses can flow. Each cylinder has 26 input pins and 26 output pins, with internal wiring that connects each input pin to a unique output pin. For simplicity, only three of the internal connections in each cylinder are shown.

- If we associate each input and output pin with a letter of the alphabet, then a single cylinder defines a monoalphabetic substitution. For example, in Figure 2.7, if an operator depresses the key for the letter A, an electric signal is applied to the first pin of the first cylinder and flows through the internal connection to the twenty-fifth output pin.

Hagelin Rotor Machine

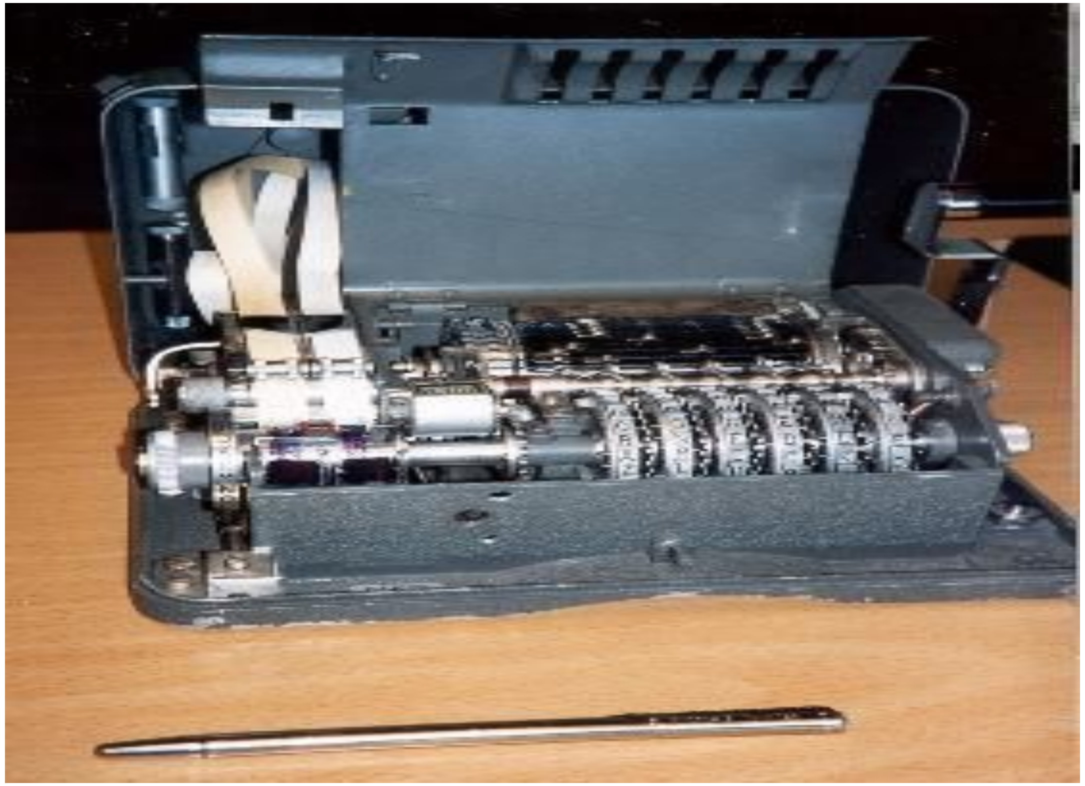
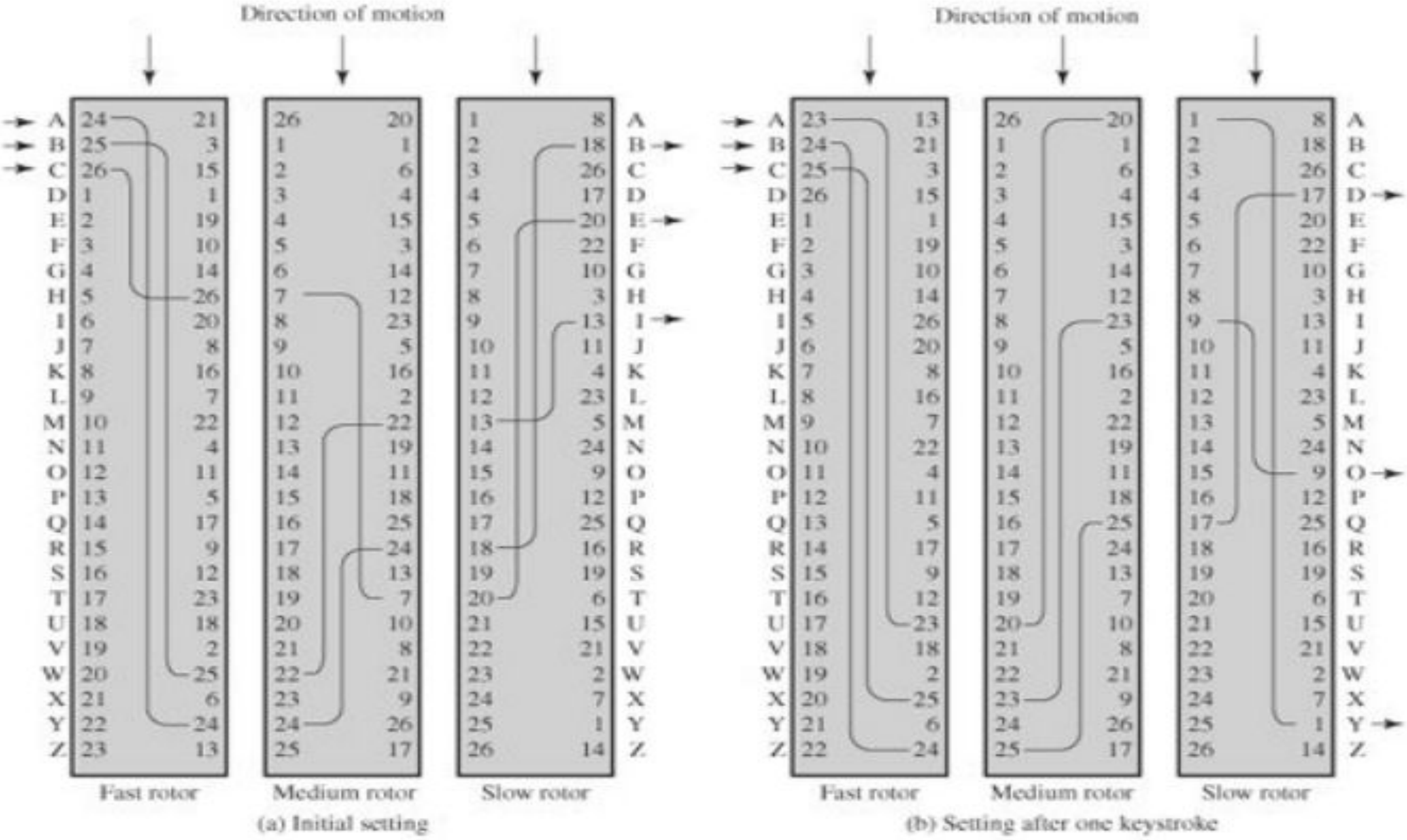


Figure 2.7. Three-Rotor Machine with Wiring Represented by Numbered Contacts



(a) Initial setting

(b) Setting after one keystroke

- Consider a machine with a single cylinder. After each input key is depressed, the cylinder rotates one position, so that the internal connections are shifted accordingly. Thus, a different monoalphabetic substitution cipher is defined. After 26 letters of plaintext, the cylinder would be back to the initial position. Thus, we have a polyalphabetic substitution algorithm with a period of 26.

- A single-cylinder system is trivial and does not present a formidable cryptanalytic task. The power of the rotor machine is in the use of multiple cylinders, in which the output pins of one cylinder are connected to the input pins of the next. Figure 2.7 shows a three-cylinder system. The left half of the figure shows a position in which the input from the operator to the first pin (plaintext letter a) is routed through the three cylinders to appear at the output of the second pin (ciphertext letter B).

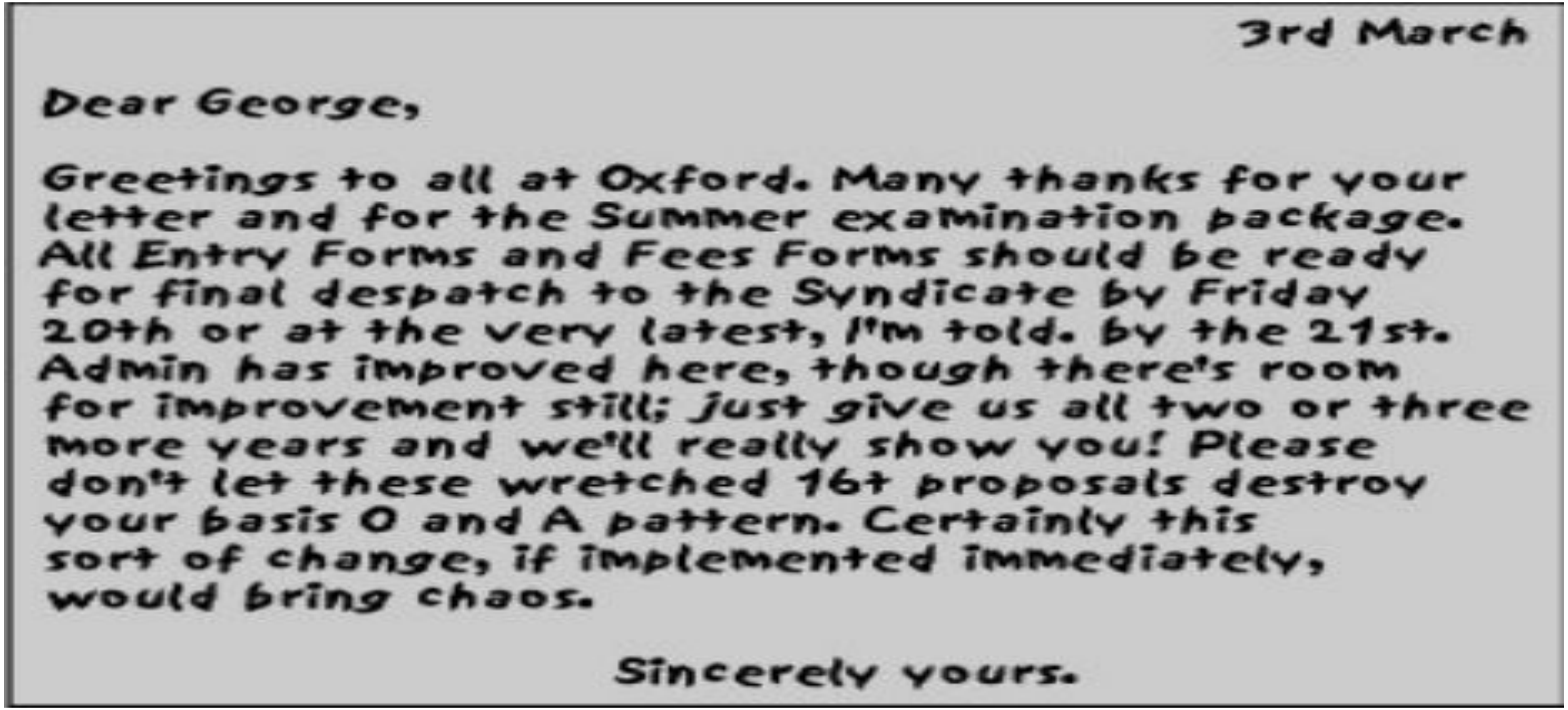
- With multiple cylinders, the one closest to the operator input rotates one pin position with each keystroke. The right half of Figure 2.7 shows the system's configuration after a single keystroke. For every complete rotation of the inner cylinder, the middle cylinder rotates one pin position. Finally, for every complete rotation of the middle cylinder, the outer cylinder rotates one pin position. This is the same type of operation seen with an odometer. The result is that there are 26 x 26 x 26 = 17,576 different substitution alphabets used before the system repeats. The addition of fourth and fifth rotors results in periods of 456,976 and 11,881,376 letters, respectively. As David Kahn eloquently put it, referring to a five-rotor machine

- A period of that length thwarts any practical possibility of a straightforward solution on the basis of letter frequency. This general solution would need about 50 letters per cipher alphabet, meaning that all five rotors would have to go through their combined cycle 50 times. The ciphertext would have to be as long as all the speeches made on the floor of the Senate and the House of Representatives in three successive sessions of Congress. No cryptanalyst is likely to bag that kind of trophy in his lifetime; even diplomats, who can be as verbose as politicians, rarely scale those heights of loquacity.
- The significance of the rotor machine today is that it points the way to the most widely used cipher ever: the Data Encryption Standard (DES).

STEGANOGRAPHY

- We conclude with a discussion of a technique that is, strictly speaking, not encryption, namely, steganography. A plaintext message may be hidden in one of two ways. The methods of steganography conceal the existence of the message, whereas the methods of cryptography render the message unintelligible to outsiders by various transformations of the text.
- A simple form of steganography, but one that is time-consuming to construct, is one in which an arrangement of words or letters within an apparently innocuous 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. Figure 2.8 shows an example in which a subset of the words of the overall message is used to convey the hidden message.

Figure 2.8. A Puzzle for Inspector Morse



3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told. by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basis O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

•Various other techniques have been used historically; some examples are the following

- **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.

•Although these techniques may seem archaic, they have contemporary equivalents. [WAYN93] proposes hiding a message by using the least significant bits of frames on a CD. For example, the Kodak Photo CD format's maximum resolution is 2048 by 3072 pixels, with each pixel containing 24 bits of RGB color information. 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.

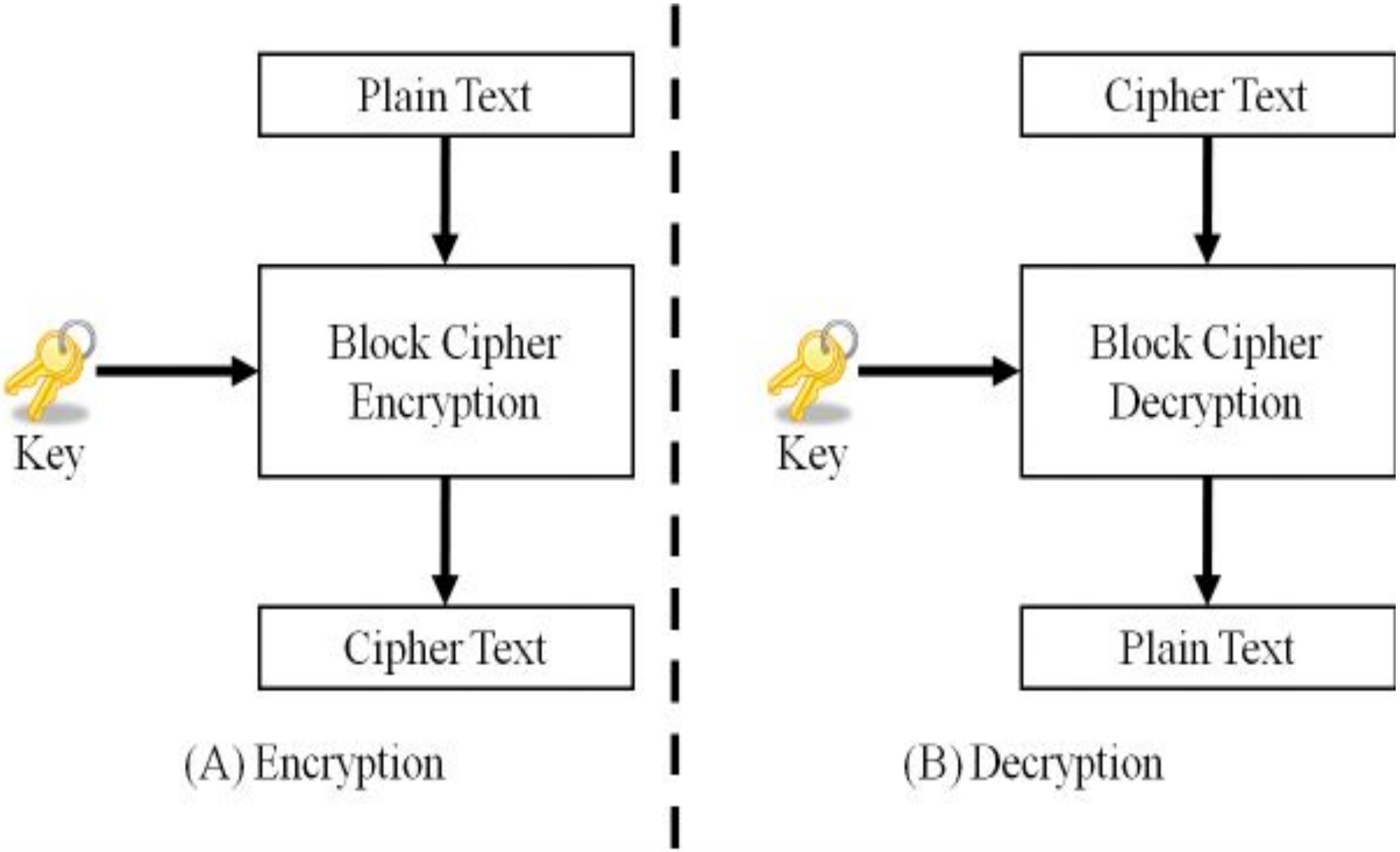
•Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information, although using some scheme like that proposed in the preceding paragraph may make it more effective. Also, once the system is discovered, it becomes virtually worthless. This problem, too, can be overcome if the insertion method depends on some sort of key (e.g., see Problem 2.11). Alternatively, a message can be first encrypted and then hidden using steganography.

•The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered. Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

Block Ciphers and the Data Encryption Standard

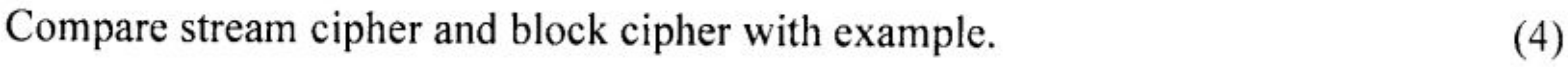
•Key Points

- A **block cipher** is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Many block ciphers have a Feistel structure. Such a structure consists of a number of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.
- The Data Encryption Standard (DES) has been the most widely used encryption algorithm until recently. It exhibits the classic Feistel structure. DES uses a 64-bit block and a 56-bit key.
- Two important methods of cryptanalysis are differential cryptanalysis and linear cryptanalysis. DES has been shown to be highly resistant to these two types of attack.



(A) Encryption

(B) Decryption



Compare stream cipher and block cipher with example. (4)

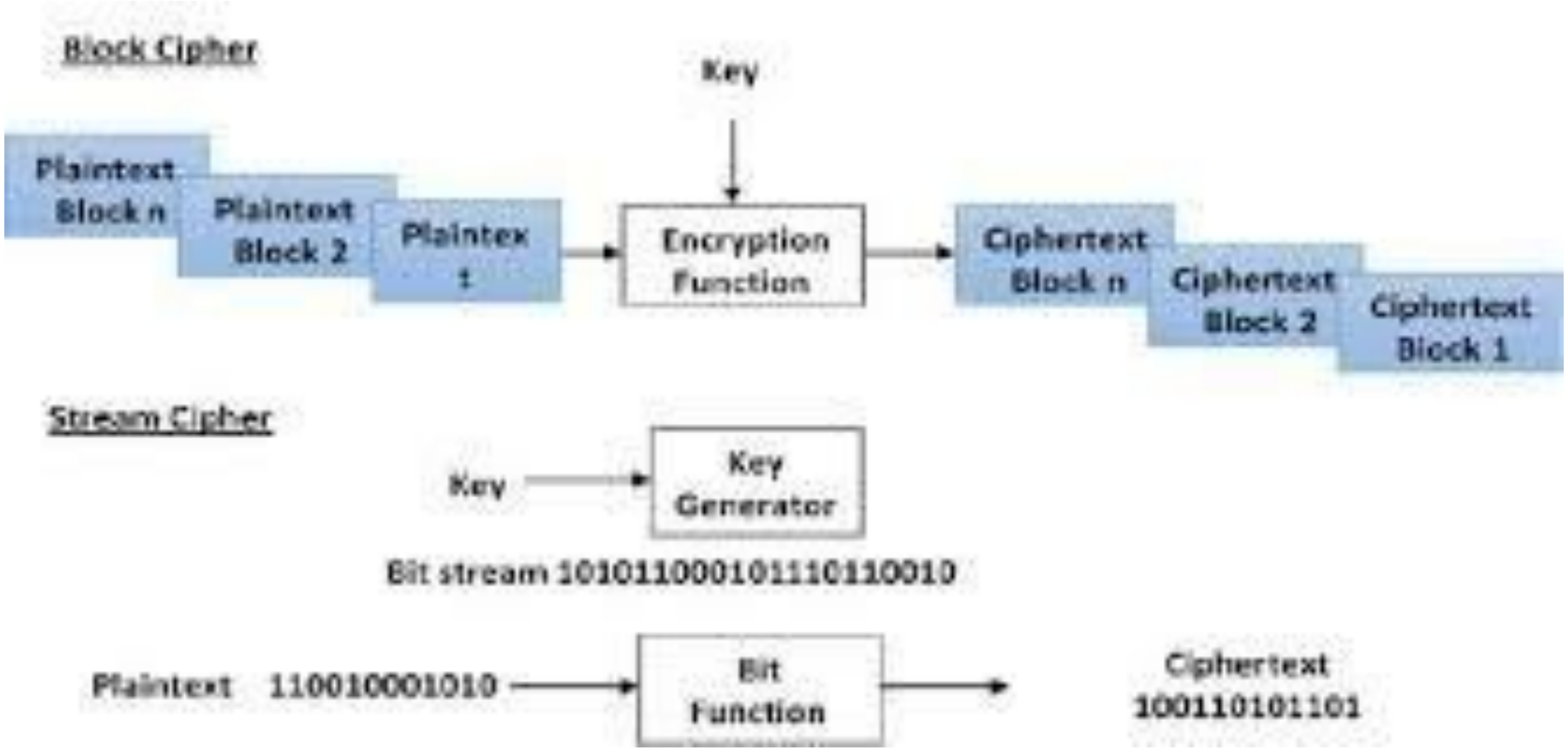
BLOCK CIPHER PRINCIPLES

- Most symmetric block encryption algorithms in current use are based on a structure referred to as a Feistel block cipher [FEIS73]. For that reason, it is important to examine the design principles of the Feistel cipher. We begin with a comparison of stream ciphers and block ciphers. Then we discuss the motivation for the Feistel block cipher structure. Finally, we discuss some of its implications.

•Stream Ciphers and Block Ciphers

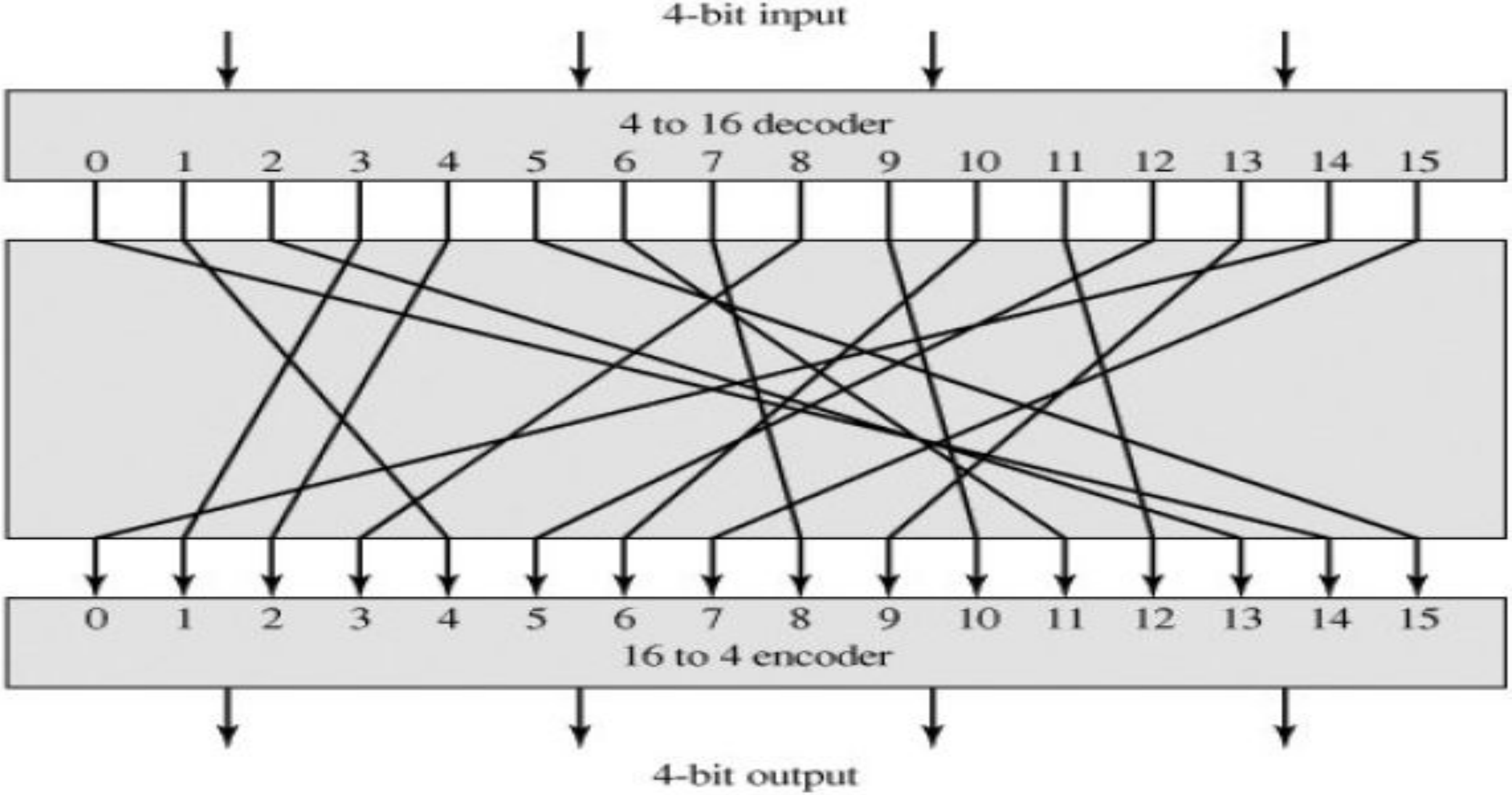
- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher. A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used.

- Far more effort has gone into analyzing block ciphers. In general, they seem applicable to a broader range of applications than stream ciphers. The vast majority of network-based symmetric cryptographic applications make use of block ciphers. Accordingly, the concern in this chapter, and in our discussions throughout the book of symmetric encryption, will focus on block ciphers.


Block Cipher
Key
Plaintext Block n
Plaintext Block 2
Plaintex t
Encryption Function
Ciphertext Block n
Ciphertext Block 2
Ciphertext Block 1
Stream Cipher
Key
Key Generator
Bit stream 10101100010111011 0010
Plaintext 110010001010
Bit Function
Ciphertext 100110101101

- Examples of Block Ciphers
- **Data Encryption Standard** (DES),
- **Triple DES** (3DES or TDEA),
- **Advanced Encryption Standard** (AES),
- **International Data Encryption Algorithm** (IDEA),
- **Blowfish**,
- **Twofish**, and
- **RC5**

Figure 3.1. General *n*-bit-*n*-bit Block Substitution (shown with *n* = 4)



- But there is a practical problem with the ideal block cipher. If a small block size, such as $n = 4$, is used, then the system is equivalent to a classical substitution cipher.
- Such systems, as we have seen, are vulnerable to a statistical analysis of the plaintext. This weakness is not inherent in the use of a substitution cipher but rather results from the use of a small block size.
- If n is sufficiently large and an arbitrary reversible substitution between plaintext and ciphertext is allowed, then the statistical characteristics of the source plaintext are masked to such an extent that this type of cryptanalysis is infeasible

The Feistel Cipher

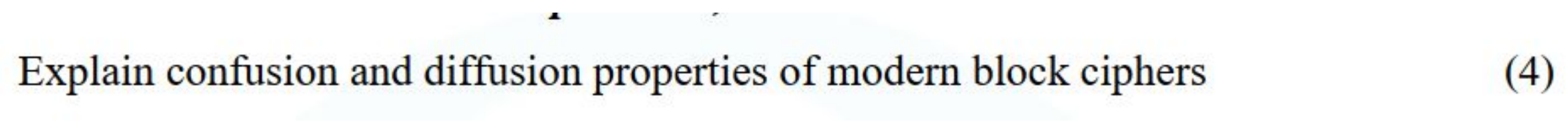
- Feistel Cipher is not a specific scheme of block cipher.
- It is a design model from which many different block ciphers are derived. DES is just one example of a Feistel Cipher.
- A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

The Feistel Cipher

- Feistel proposed that we can approximate the ideal block cipher by utilizing the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers. The essence of the approach is to develop a block cipher with a key length of k bits and a block length of n bits, allowing a total of $2k$ possible transformations, rather than the $2n!$ transformations available with the ideal block cipher.

- In particular, Feistel proposed the use of a cipher that alternates substitutions and permutations. In fact, this is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates *confusion* and *diffusion* functions [SHAN49]. We look next at these concepts of diffusion and confusion and then present the Feistel cipher. But first, it is worth commenting on this remarkable fact:

- The Feistel cipher structure, which dates back over a quarter century and which, in turn, is based on Shannon's proposal of 1945, is the structure used by many significant symmetric block ciphers currently in use.



Explain confusion and diffusion properties of modern block ciphers (4)

- **Diffusion and Confusion**

- The terms *diffusion* and *confusion* were introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system Shannon's concern was to thwart cryptanalysis based on statistical analysis.

- **In diffusion, the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext**. This is achieved by having each plaintext digit affect the value of many ciphertext digits; generally this is equivalent to having each ciphertext digit be affected by many plaintext digits. An example of diffusion is to encrypt a message $M = m1, m2, m3,...$ of characters with an averaging operation:

- The mechanism of diffusion seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to thwart attempts to deduce the key. On the other hand, **confusion** seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
- Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key. This is achieved by the use of a complex substitution algorithm. In contrast, a simple linear substitution function would add little confusion.

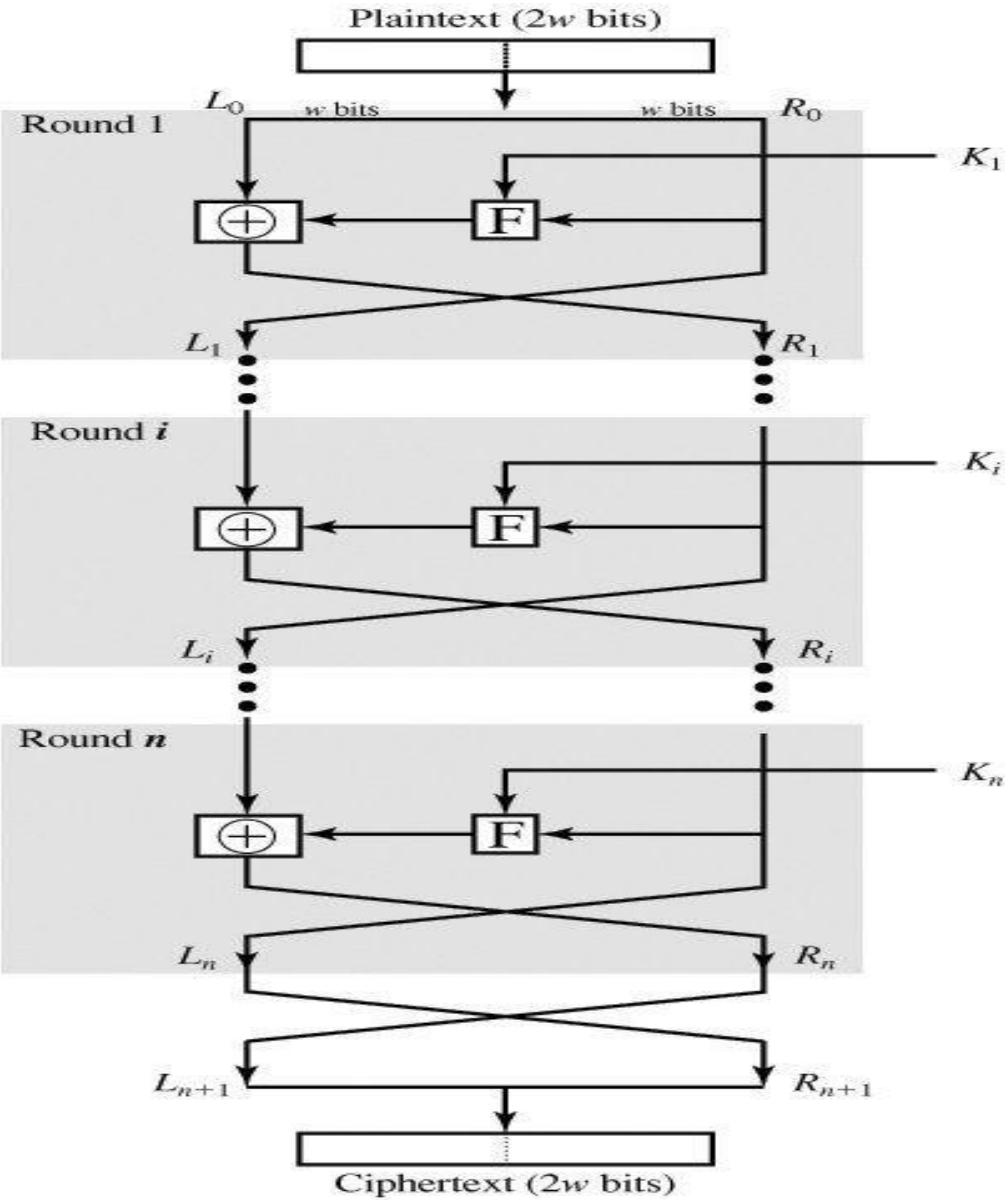
- **Confusion** fabricates a complex relation between the **cipher text** and **encryption key** by implementing a complex **substitution** algorithm.
- Whereas, the **diffusion** fabricates a complex relation between **plain text** and **cipher text** by implementing more complex **permutation** algorithm.

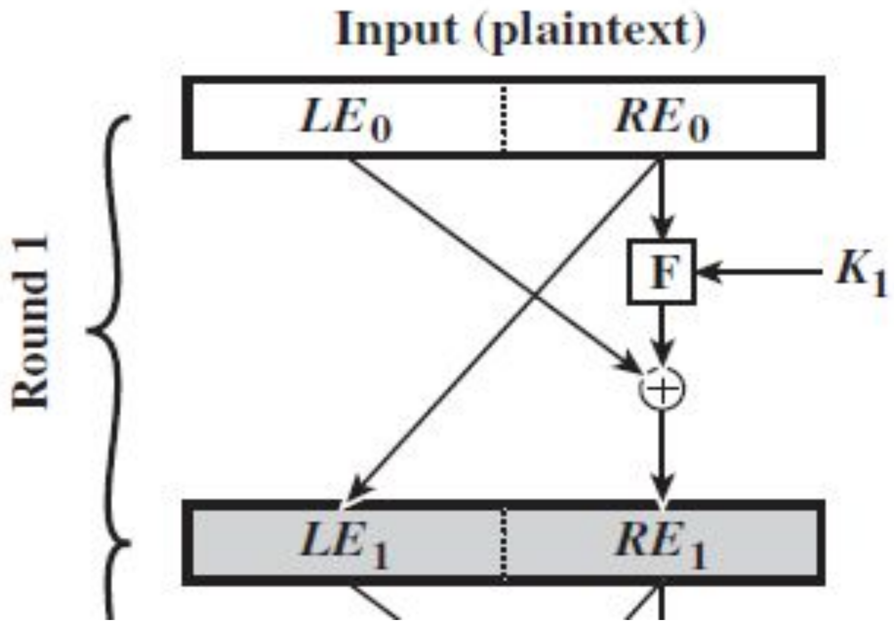
Feistel Cipher Structure

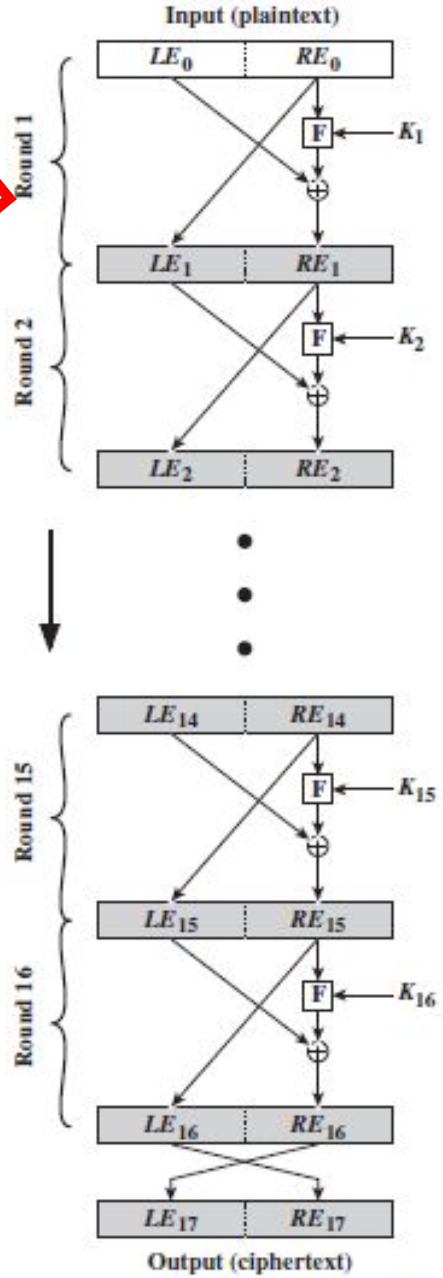
- Figure 3.2 depicts the structure proposed by Feistel. The inputs to the encryption algorithm are a plaintext block of length 2*w* bits and a key *K*.
- The plaintext block is divided into two halves, *L*0 and *R*0. The two halves of the data pass through *n* rounds of processing and then combine to produce the ciphertext block.
- Each round *i* has as inputs *Li-1* and *Ri-1*, derived from the previous round, as well as a subkey *Ki*, derived from the overall *K*. In general, the subkeys *Ki* are different from *K* and from each other.

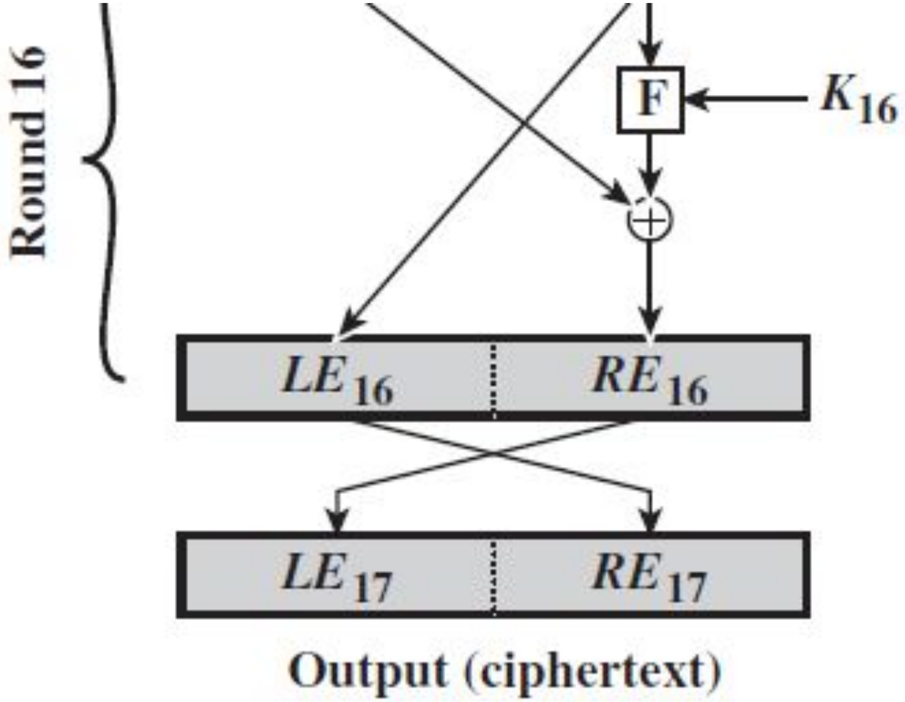
Figure 3.2. Classical Feistel Network

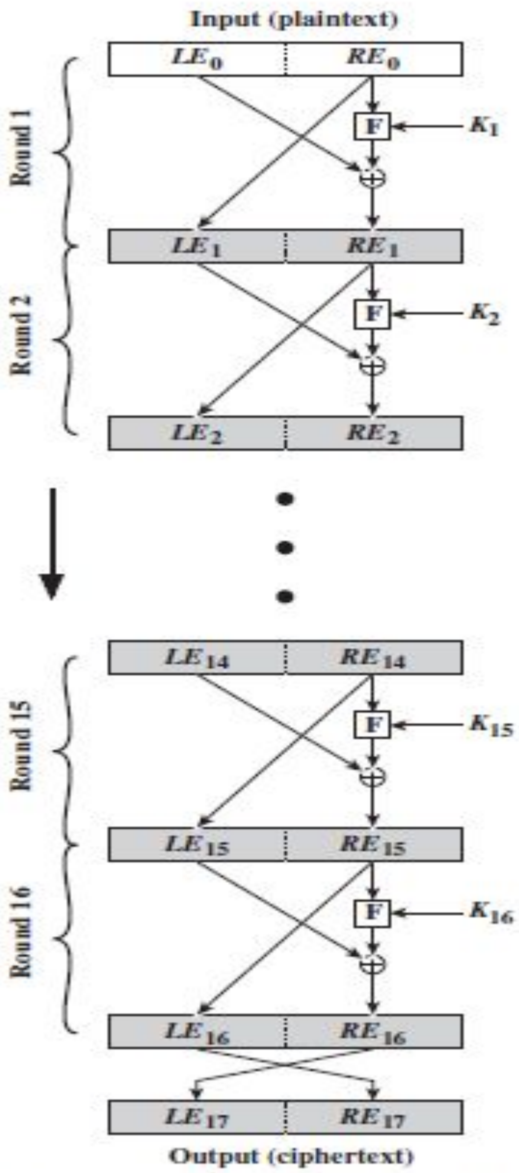
- The encryption process uses
- the Feistel structure consisting
- multiple rounds of processing
- of the plaintext, each round
- consisting of a “substitution”
- step followed by a permutation
- step.




Input (plaintext)
LE_0
RE_0
Round 1
F
K_1
LE_1
RE_1


Input (plaintext)
LE_0
RE_0
Round 1
F
K_1
LE_1
RE_1
Round 2
F
K_2
LE_2
RE_2
LE_{14}
RE_{14}
Round 15
F
K_{15}
LE_{15}
RE_{15}
Round 16
F
K_{16}
LE_{16}
RE_{16}
LE_{17}
RE_{17}
Output (ciphertext)


Round 16
F
K_{16}
LE_{16}
RE_{16}
LE_{17}
RE_{17}
Output (ciphertext)


Input (plaintext)
LE_0
RE_0
Round 1
F
K_1
LE_1
RE_1
Round 2
F
K_2
LE_2
RE_2
LE_{14}
RE_{14}
Round 15
F
K_{15}
LE_{15}
RE_{15}
Round 16
F
K_{16}
LE_{16}
RE_{16}
LE_{17}
RE_{17}
Output (ciphertext)

- All rounds have the same structure. A **substitution** is performed on the left half of the data. This is done by applying a *round function* F to the right half of the data and then taking the exclusive-OR of the output of that function and the left half of the data. The round function has the same general structure for each round but is parameterized by the round subkey *Ki*.

- Following this substitution, a **permutation** is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the substitution-permutation network (SPN) proposed by Shannon.

•The exact realization of a Feistel network depends on the choice of the following parameters and design features:

- **Block size:** Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm. The greater security is achieved by greater diffusion Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design. However, the new AES uses a 128-bit block size.

- **Key size:** Larger key size means greater security but may decrease encryption/decryption speed. The greater security is achieved by greater resistance to brute-force attacks and greater confusion. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
- **Number of rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.

- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

- **Round function:** Again, greater complexity generally means greater resistance to cryptanalysis. There are two other considerations in the design of a Feistel cipher:

- **Fast software encryption/decryption:** In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation. Accordingly, the speed of execution of the algorithm becomes a concern.

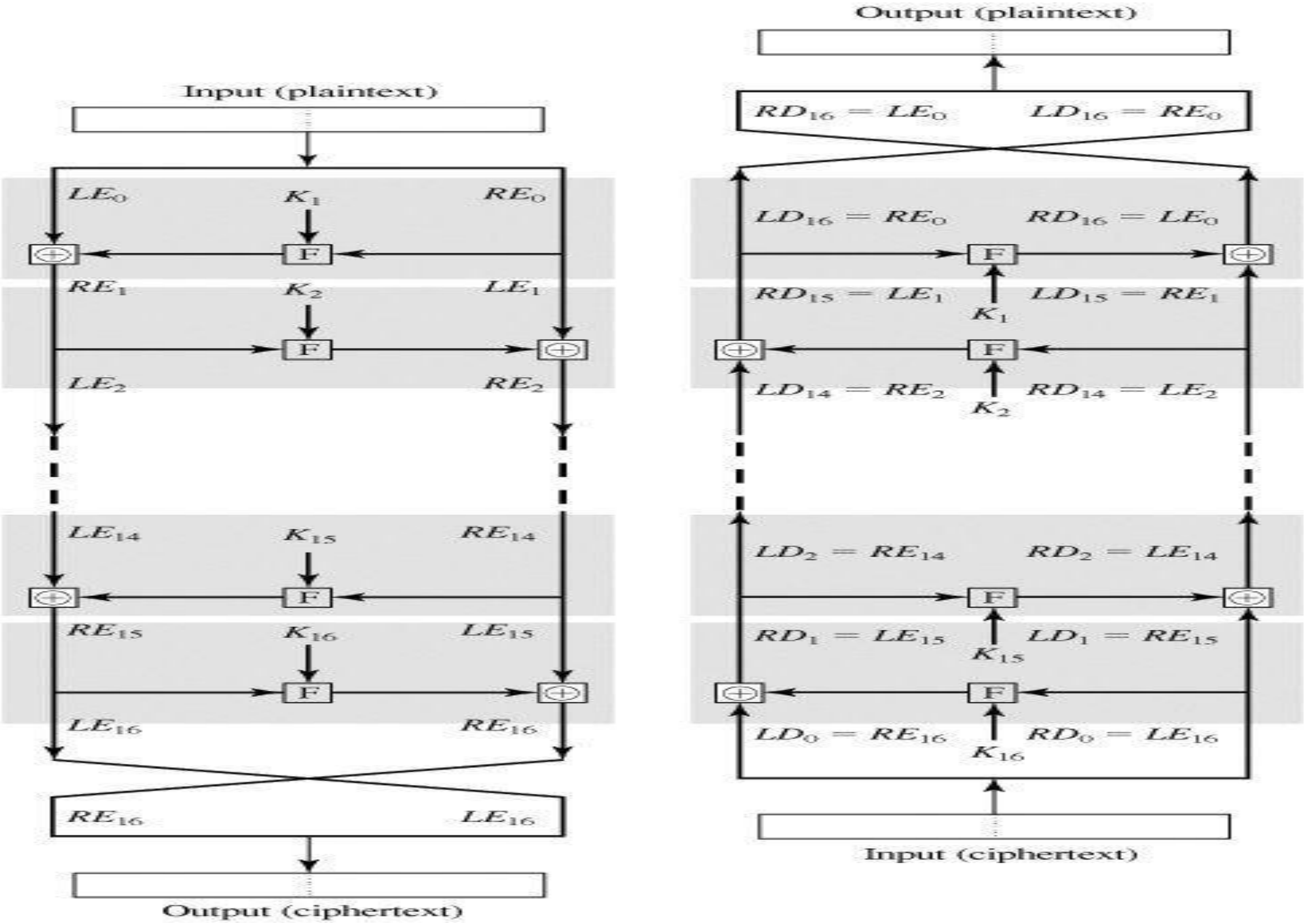
- **Ease of analysis:** Although we would like to make our algorithm as difficult as possible to cryptanalyze, there is great benefit in making the algorithm easy to analyze. That is, if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength. DES, for example, does not have an easily analyzed functionality.

Feistel Decryption Algorithm

- The process of decryption with a Feistel cipher is essentially the same as the encryption process. The rule is as follows: Use the ciphertext as input to the algorithm, but use the subkeys *Ki* in reverse order.
- That is, use *Kn* in the first round, *Kn-1* in the second round, and so on until *K*1 is used in the last round. This is a nice feature because it means we need not implement two different algorithms, one for encryption and one for decryption.
- To see that the same algorithm with a reversed key order produces the correct result, consider Figure 3.3, which shows the encryption process going down the left-hand side and the decryption process going up the right-hand side for a 16-round algorithm (the result would be the same for any number of rounds).

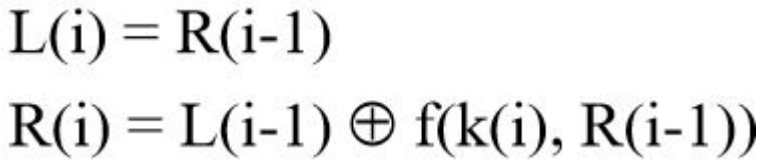
- For clarity, we use the notation LE*i* and RE*i* for data traveling through the encryption algorithm and *LDi* and RD*i* for data traveling through the decryption algorithm.
- The diagram indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped.
- To put this another way, let the output of the *i*th encryption round be LE*i*||RE*i* (*Li* concatenated with *Ri*). Then the corresponding input to the (16 *i*) th decryption round is RE*i*||LE*i* or, equivalently, RD16-*i*||*LD*16-*i*.

Figure 3.3. Feistel Encryption and Decryption



- After the last iteration of the encryption process, the two halves of the output are swapped, so that the ciphertext is RE16||LE16. The output of that round is the ciphertext. Now take that ciphertext and use it as input to the same algorithm. The input to the first round is RE16||LE16, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.

- Now we would like to show that the output of the first round of the decryption process is equal to a 32- bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that



$$L(i) = R(i-1)$$

$$R(i) = L(i-1) \oplus f(k(i), R(i-1))$$

- $LE_{16} = RE_{15}$
- $RE_{16} = LE_{15}$ x $F(RE_{15}, K_{16})$
- On the decryption side,

$LD_1 = RD_0 = LE_{16} = RE_{15}$

$RD_1 = LD_0$ x $F(RD_0, K_{16})$

- = RE_{16} x $F(RE_{15}, K_{16})$
- = $[LE_{15}$ x $F(RE_{15}, K_{16})]$ x $F(RE_{15}, K_{16})$

- The XOR has the following properties:
- $[A$ x $B]$ x $C = A$ x $[B$ x $C]$
- D x $D = 0$

- E x $0 = E$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $LE_{15}||RE_{15}$, which is the 32-bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the *i*th iteration of the encryption algorithm,

$LE_i = RE_{i-1}$
$RE_i = LE_{i-1}$ x $F(RE_{i-1}, K_i)$
Rearranging terms,
$RE_{i-1} = LE_i$
$LE_{i-1} = RE_i$ x $F(RE_{i-1}, K_{i2} = RE_i$ x $F(LE_i, K_i)$

- Thus, we have described the inputs to the *i*th iteration as a function of the outputs, and these equations confirm the assignments shown in the right-hand side of Figure 3.3.
- Finally, we see that the output of the last round of the decryption process is $RE_0 \| LE_0$. A 32-bit swap recovers the original plaintext, demonstrating the validity of the Feistel decryption process.
- Note that the derivation does not require that F be a reversible function. To see this, take a limiting case in which F produces a constant output (e.g., all ones) regardless of the values of its two arguments. The equations still hold

THE DATA ENCRYPTION STANDARD

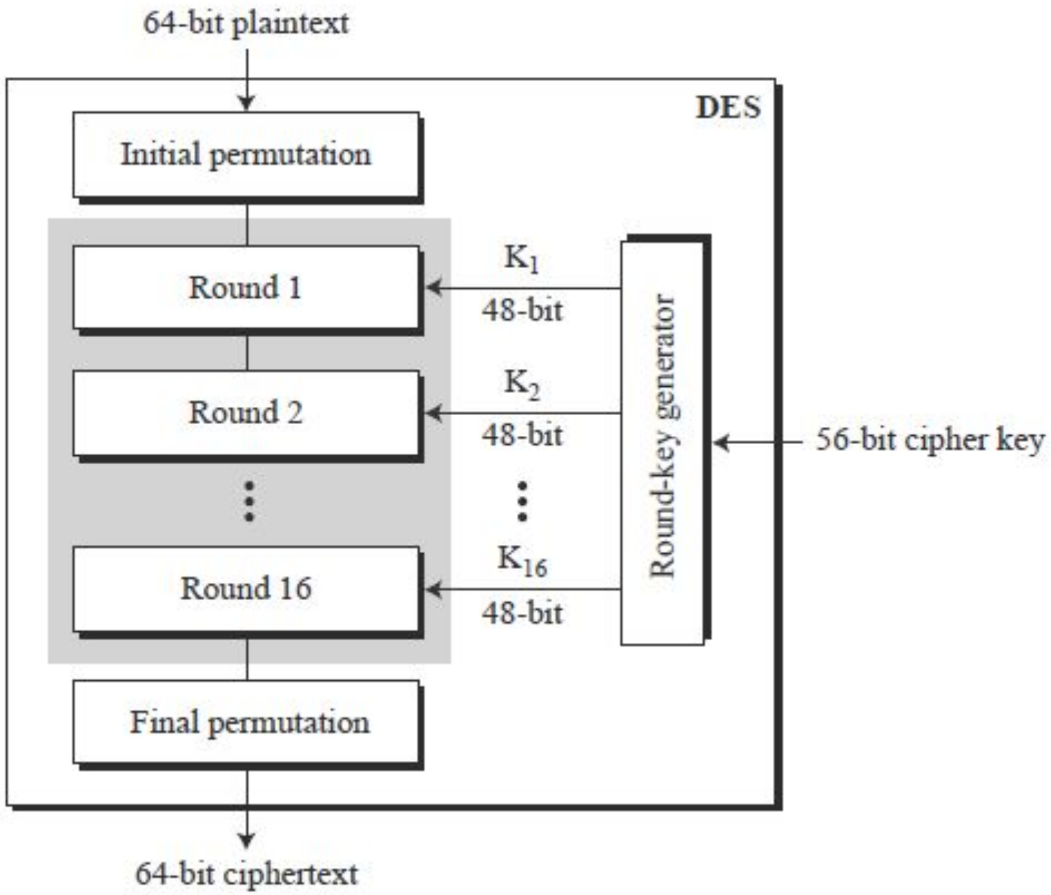
- The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46).
- The algorithm itself is referred to as the Data Encryption Algorithm (DEA).
- For DES, data are encrypted in 64-bit blocks using a 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.
- The DES enjoys widespread use. It has also been the subject of much controversy concerning how secure the DES is. To appreciate the nature of the controversy, let us quickly review the history of the DES.

DES History

- IBM developed Lucifer cipher
  - by team led by Feistel in late 60's
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others
- in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES

- DES is based on the two fundamental attributes of cryptography: substitution (also called as confusion) and transposition (also called as diffusion). DES consists of 16 steps, each of which is called as a round. Each round performs the steps of substitution and transposition
- DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −

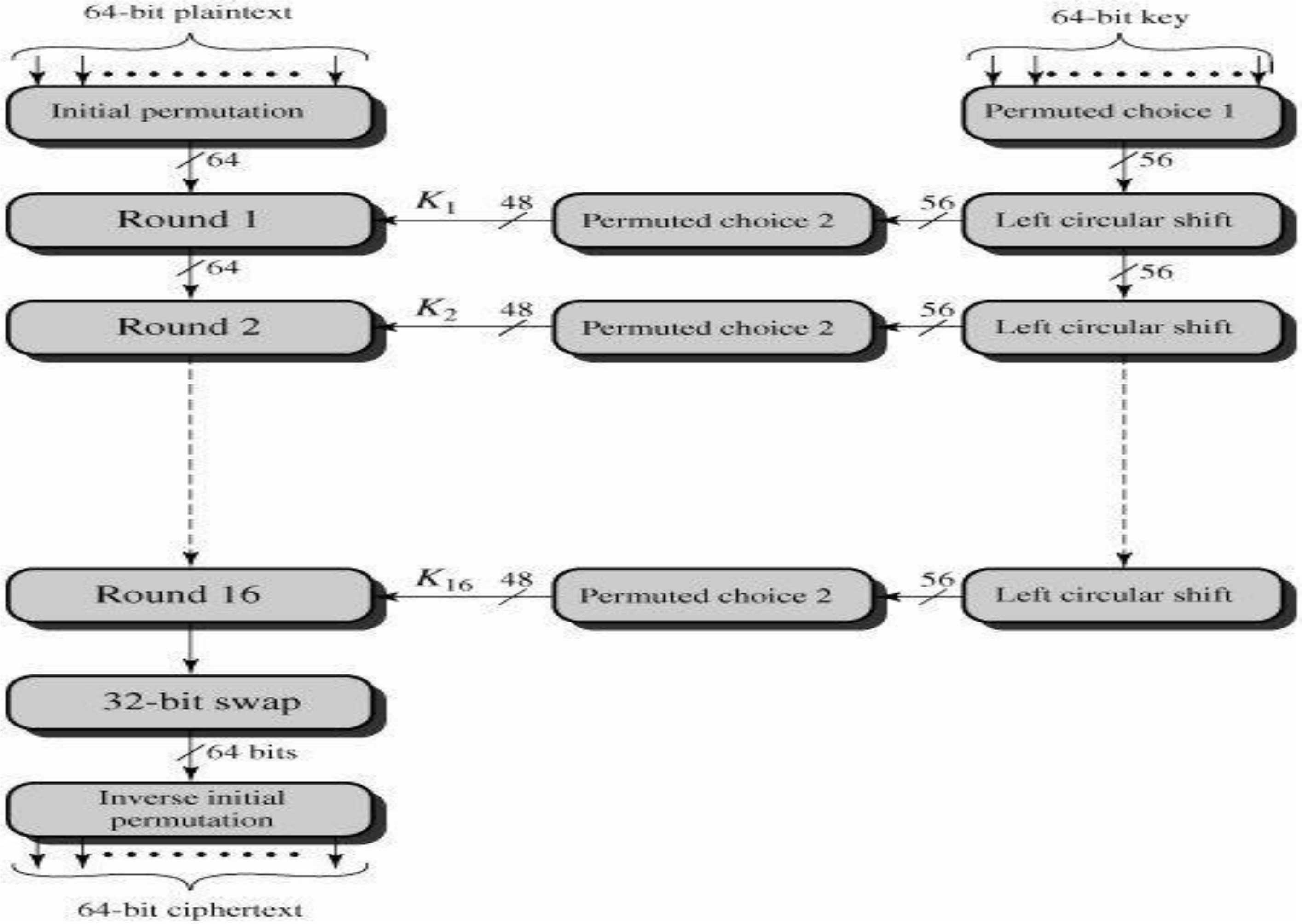
General structure of DES



DES Encryption

- The overall scheme for DES encryption is illustrated in Figure 3.4. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

Figure 3.4. General Depiction of DES Encryption Algorithm



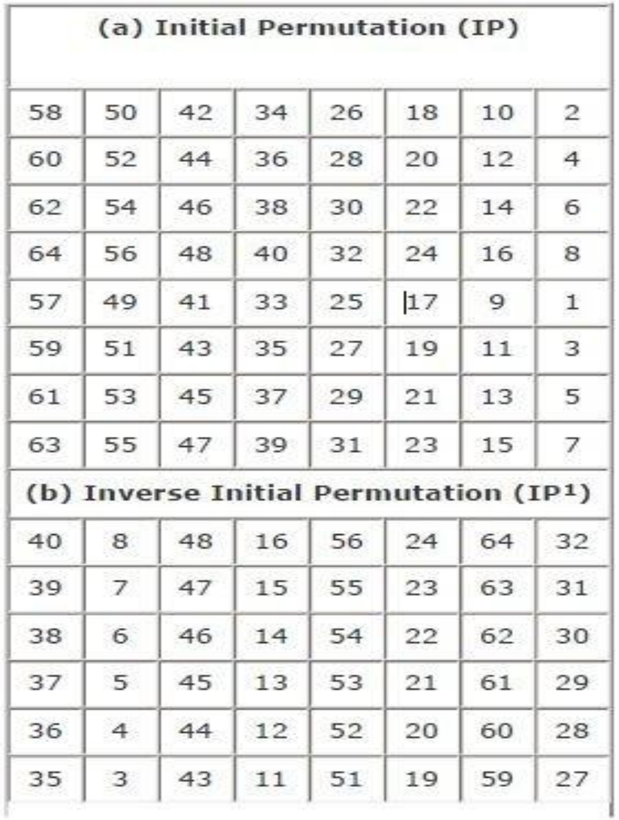
- Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases.
- First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*. This is followed by a phase consisting of 16 rounds 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 a permutation (IP-1) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext. With the exception of the initial and final permutations, DES has the exact structure of a Feistel cipher, as shown in Figure 3.2.

- The right-hand portion of Figure 3.4 shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function. Then, for each of the 16 rounds, a *subkey* (*Ki*) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

•Initial Permutation

- The initial permutation and its inverse are defined by tables, as shown in Tables 3.2a and 3.2b, respectively. The tables are to be interpreted as follows. The input to a table consists of 64 bits numbered from 1 to 64.
- The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits.

Table 3.2. Permutation Tables for DES

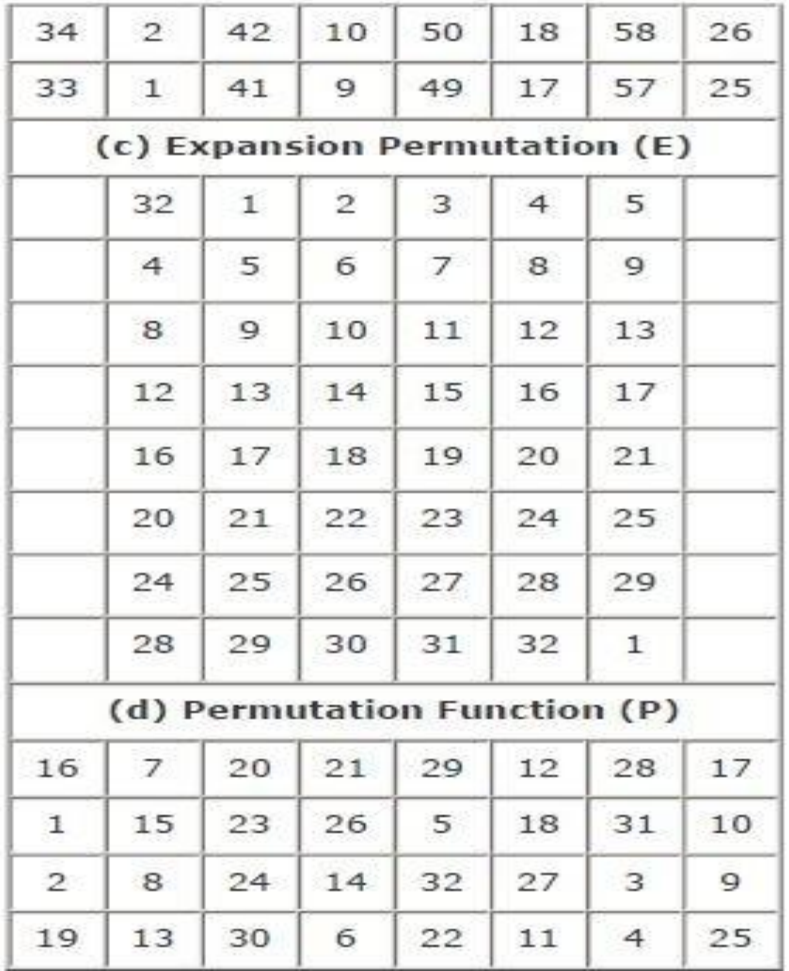


(a) Initial Permutation (IP)

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

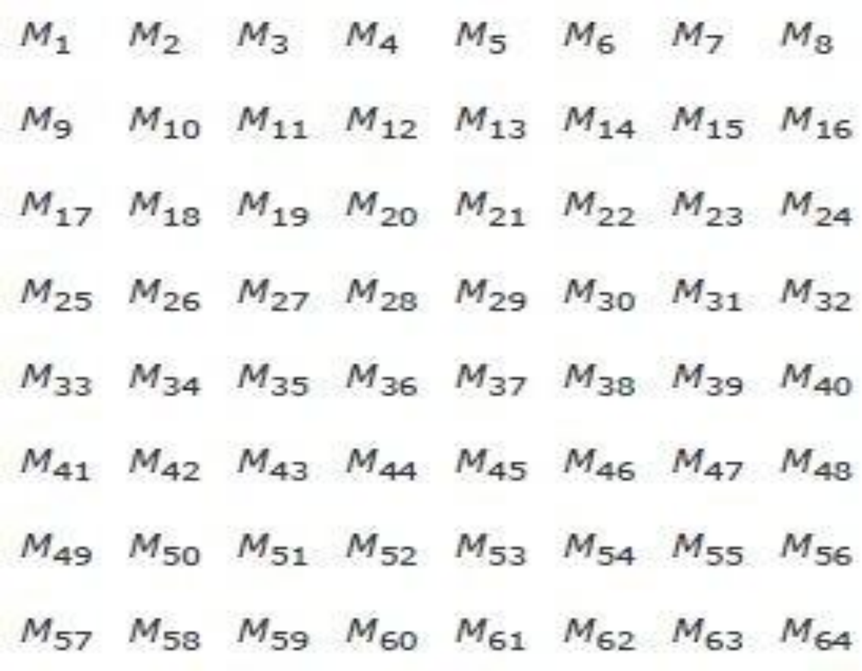
(c) Expansion Permutation (E)

	32	1	2	3	4	5	
	4	5	6	7	8	9	
	8	9	10	11	12	13	
	12	13	14	15	16	17	
	16	17	18	19	20	21	
	20	21	22	23	24	25	
	24	25	26	27	28	29	
	28	29	30	31	32	1	

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

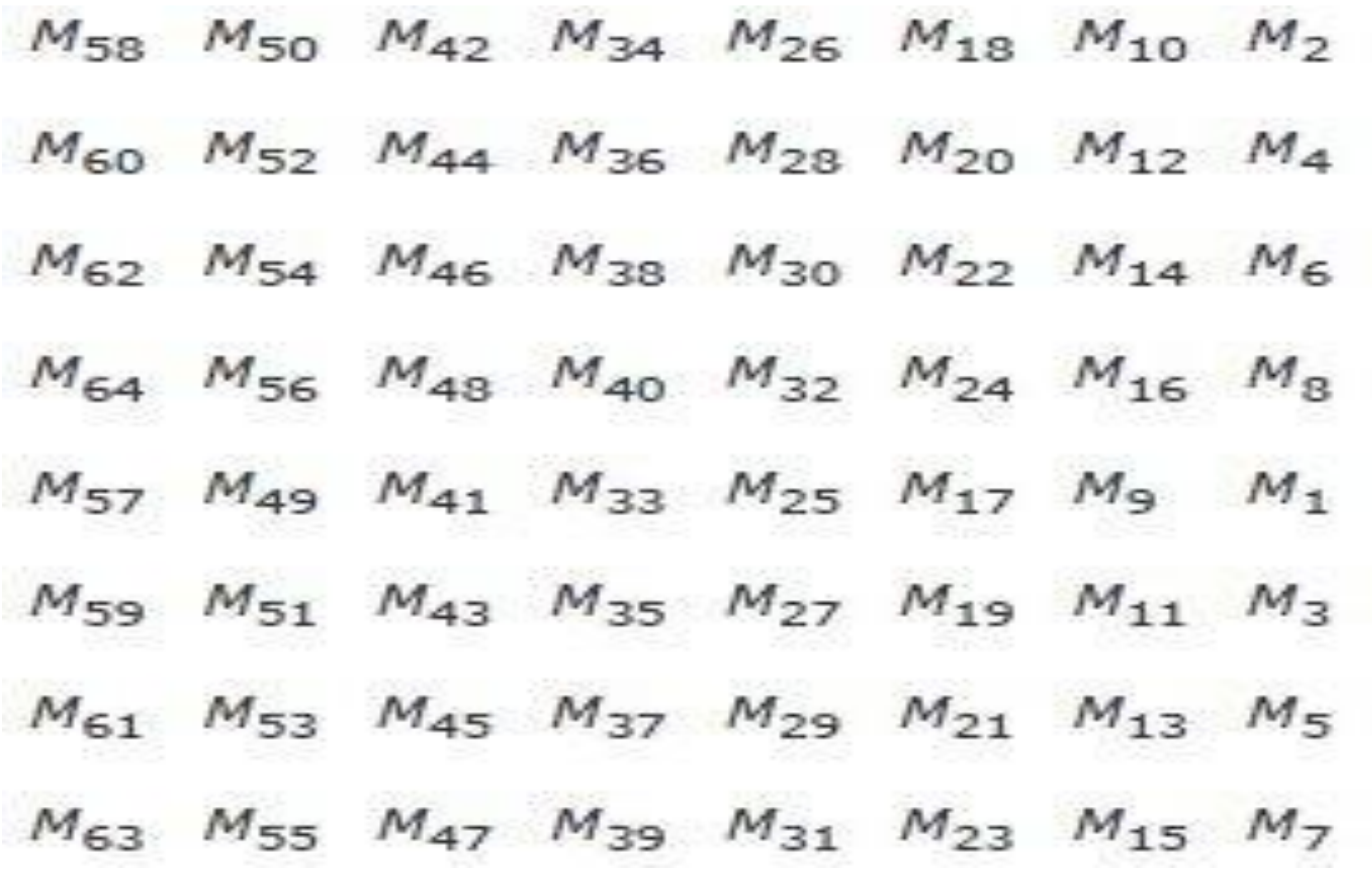
- To see that these two permutation functions are indeed the inverse of each other, consider the following 64-bit input *M*:



M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

where *Mi* is a binary digit. Then the permutation *X* = IP(*M*) is as follows:

- If we then take the inverse permutation Y = IP-1(X) = IP-1(IP(M)), it can be seen that the original ordering of the bits is restored



M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_{2}
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_{4}
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_{6}
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_{8}
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_{9}	M_{1}
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_{3}
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_{5}
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_{7}

DES ROUND STRUCTURE

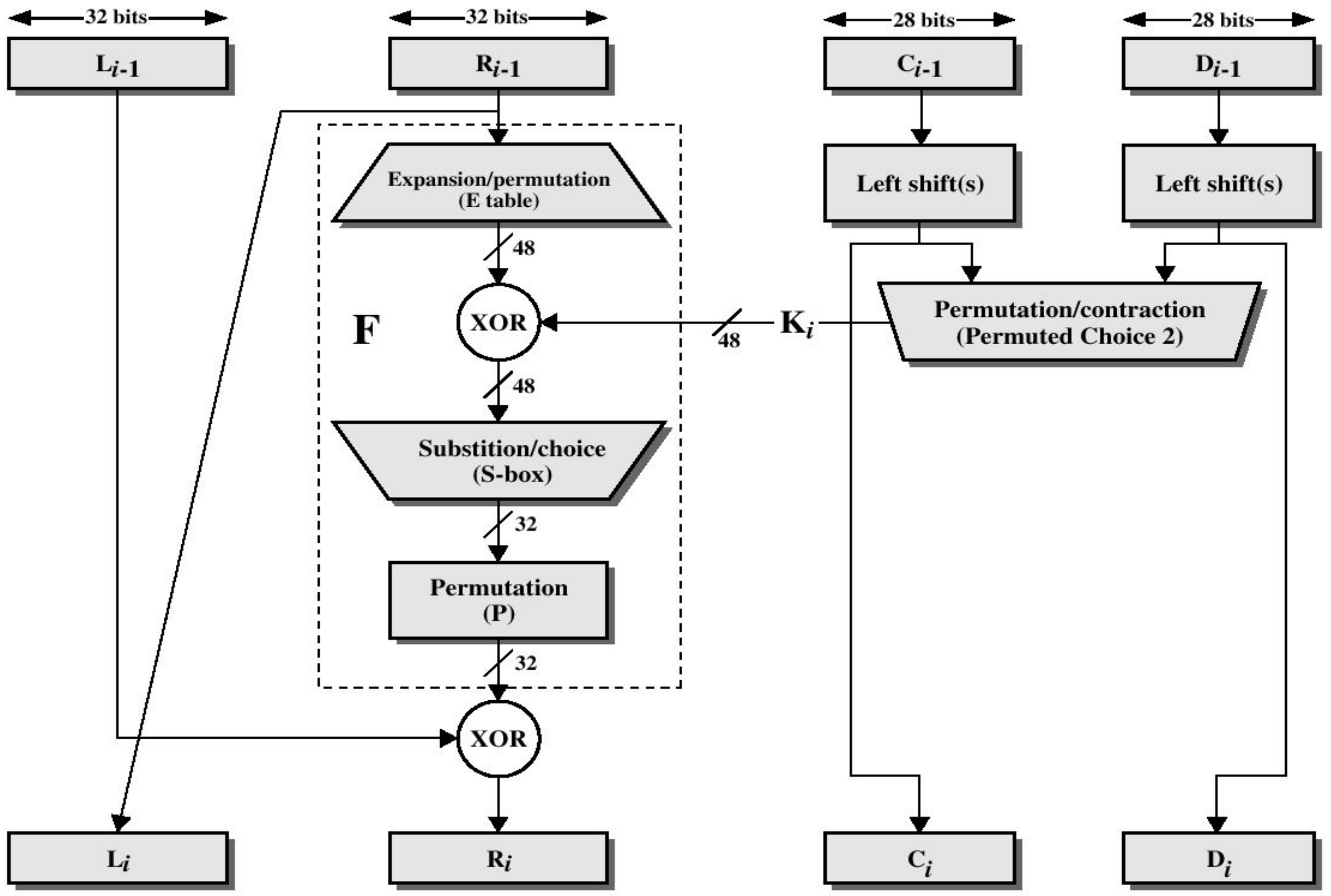
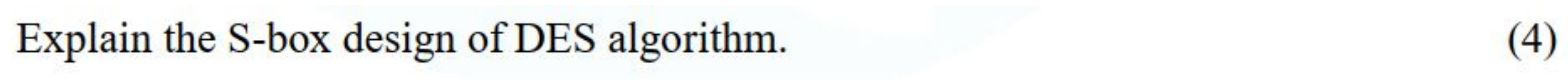


Figure 2.4 Single Round of DES Algorithm

- uses two 32-bit L & R halves
  - as for any Feistel cipher can describe as:
    - *Li* = *Ri*–1
    - *Ri* = *Li*–1 xor F(*Ri*–1, *Ki*)

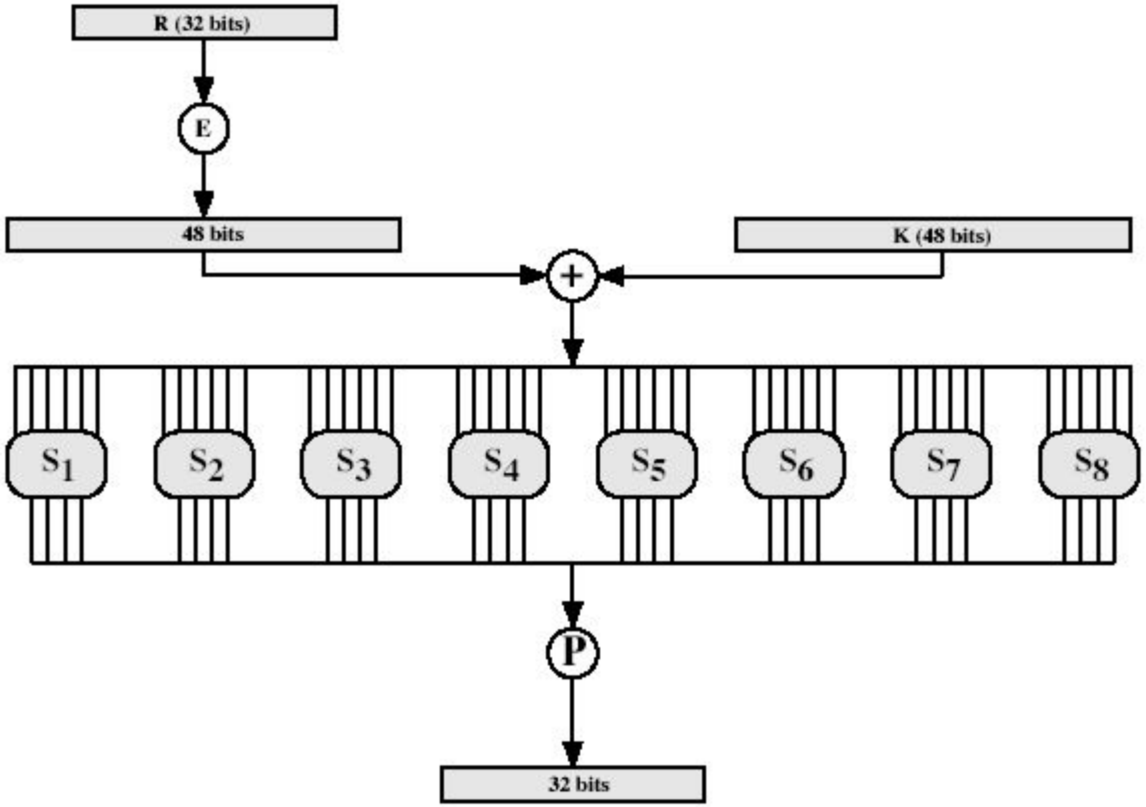
– takes 32-bit R half and 48-bit subkey and:

- expands R to 48-bits using perm E
- adds to subkey
- passes through 8 S-boxes to get 32-bit result



Explain the S-box design of DES algorithm. (4)

- **SUBSTITUTION BOXES S**
- The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration −



- have eight S-boxes which map 6 to 4 bits
  – each S-box is actually 4 little 4 bit boxes
  - o outer bits 1 & 6 (**row** bits) select one rows
  - o inner bits 2-5 (**col** bits) are substituted
  - o result is 8 lots of 4 bits, or 32 bits

  – row selection depends on both data & key
  - o feature known as autoclaving (autokeying)

How key generation is done in DES. (4)

- **DES Key Schedule**

– forms subkeys used in each round
– consists of:
o initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves

o 16 stages consisting of:
selecting 24-bits from each half
▪ permuting them by PC2 for use in function f,
▪ rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K

- **DES Decryption**

– decrypt must unwind steps of data computation
– with Feistel design, do encryption steps again
– using subkeys in reverse order (SK16 … SK1)
– note that IP undoes final FP step of encryption
– 1st round with SK16 undoes 16th encrypt round
– 16th round with SK1 undoes 1st encrypt round
– then final FP undoes initial encryption IP
– thus recovering original data value

Avalanche Effect

A small change in plaintext results in the very great change in the ciphertext.

– key desirable property of encryption alg
– where a change of **one** input or key bit results in changing approx **half** output bits
– making attempts to “home-in” by guessing keys impossible
– DES exhibits strong avalanche

STRENGTH OF DES

- **Concerns on key size and nature of algorithm**
- **Key Size**
  – 56-bit keys have 256 = 7.2 x 1016 values
  – brute force search looks hard
  – DES was finally and definitively proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose "DES cracker" machine that was built for less than $250,000. The attack took less than three days.
  The EFF has published a detailed description of the machine, enabling others to build their own cracker
  – It is important to note that there is more to a key-search attack than simply running through all possible keys. Unless known plaintext is provided, the analyst must be able to recognize plaintext as plaintext.

- **Nature of algorithm**
  – the design criteria for the S boxes ,and indeed for the entire algorithm ,were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an
  opponent who knows the weaknesses in the S-boxes.
  – This assertion is tantalizing, and over the years a number of regularities and unexpected behaviors of the S-boxes have been discovered.

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

- **Analytic Attacks**

  now have several analytic attacks on DES

  – these utilize some deep structure of the cipher

  o by gathering information about encryptions

  o can eventually recover some/all of the sub-key bits

  o if necessary then exhaustively search for the rest

  – generally these are statistical attacks

  – include

  o differential cryptanalysis

  o linear cryptanalysis

- **Differential Cryptanalysis**
  - One of the most significant recent (public) advances in cryptanalysis
  - Known by NSA in 70's DES design Murphy, Biham& Shamir published in 90’s
  - Powerful method to analyse block ciphers
  - Used to analyse most current block ciphers with varying degrees of success
  - DES reasonably resistant to it.
- **Linear Cryptanalysis**
  - Another recent development also a statistical method
  - Must be iterated over rounds, with decreasing probabilities
  - Developed by Matsui et al in early 90's based on finding linear approximations
  - Can attack DES with known plaintexts, easier but still in practice infeasible

BLOCK CIPHER DESIGN PRINCIPLES

- **DES Design Criteria**
  - As reported by Coppersmith in [COPP94]
  - 7 criteria for S-boxes provide for non-linearity
  - Resistance to differential cryptanalysis
  - Good confusion
  - 3 criteria for permutation P provide for increased diffusion

- **The criteria for the S-boxes are as follows.**

❖ No output bit of any S-box should be too close a linear function of the input bits. Specifically, if we select any output bit and any subset of the six input bits, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1,but rather should be near 1/2.

❖ Each row of an S-box (determined by a fixed value of the leftmost and rightmost input bits) should include all 16 possible output bit combinations.

❖ If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.

- If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.

❖ If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.

❖ For any nonzero 6-bit difference between inputs, no more than eight of the 32 pairs of inputs exhibiting that difference may result in the same output difference.
❖ This is a criterion similar to the previous one, but for the case of three S-boxes.

- **The criteria for the permutation P are as follows.**

❖ The four output bits from each S-box at round are distributed so that two of them affect (provide input for) “middle bits” of round and the other two affect end bits. The two middle bits of input to an S-box are not shared with adjacent Sboxes.The end bits are the two left-hand bits and the two right-hand bits, which are shared with adjacent S-boxes.

❖ The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.

❖ For two S-boxes , , if an output bit from affects a middle bit of on the next round, then an output bit from cannot affect a middle bit of .This implies that,for ,an output bit from must not affect a middle bit of . These criteria are intended to increase the diffusion of the algorithm.

- Three critical aspects of block cipher design:

  ✔ the number of rounds,

  ✔ design of the function F, and

  ✔ key scheduling.

  1.The number of rounds

  □ The greater the number of rounds, the more difficult it is to perform cryptanalysis

  □ The criterion should be that the number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack.

  □ This criterion was certainly used in the design of DES.

  □ If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search.

  □ This criterion is attractive, because it makes it easy to judge the strength of an algorithm and to compare different algorithms.

  □ In the absence of a cryptanalytic breakthrough, the strength of any algorithm that satisfies the criterion can be judged solely on key length.

- **2. Design of Function F**

 The heart of a Feistel block cipher is the function F in DES, this function relies on the use of S-boxes.

 DESIGN CRITERIA FOR F

o The function F provides the element of confusion in a Feistel cipher.
o One criterion is that F be nonlinear, The more nonlinear F, the more difficult any type of cryptanalysis will be.

o The algorithm should have good avalanche properties, this means that a change in one bit of the input should produce a change in many bits of the output. ie; **strict**

avalanche criterion (SAC)
o Another criterion is the **bit independence criterion (BIC),**which states that output bit „j‟ and „k‟ should change independently and should change independently when any single input bit „i‟ is inverted for all „i‟ , „j‟ ,‟k‟

-
  ○

- **S-BOX DESIGN**
  - **Larger S-boxes, by and large, are more resistant to differential and linear cryptanalysis**

3. key scheduling.

 select subkeys to maximize the difficulty of deducing individual subkeys and the
difficulty of working back to the main key.

 No general principles for this have yet been promulgated

BLOCK CIPHER DESIGN PRINCIPLES

Aspects of block cipher design:

- The cryptographic strength of a Feistel cipher derives from three aspects
  - The number of rounds
  - Design of the function F
  - Key scheduling

The number of rounds

- In DES, the differential cryptanalysis attack requires $2^{55.1}$ operations, whereas brute force requires 2^{55}
- If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than a brute-force key search.
- This criterion makes it easy to judge the strength of an algorithm and to compare different algorithms

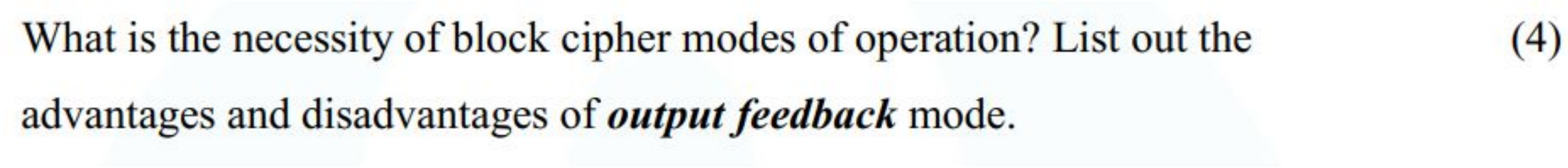
Design of Function F

- One obvious criterion is that F be nonlinear
  - The more nonlinear F, the more difficult any type of cryptanalysis will be.
- The **Strict Avalanche Criterion (SAC)**
  - any output bit j *of an S-box* should change with probability ½ when any single input bit i is inverted
- The **Bit Independence Criterion(BIC)**
  - output bits j *and* k *should change independently when any* single input bit i is inverted

Key Schedule Algorithm

- The key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

BLOCK CIPHER MODES of operation

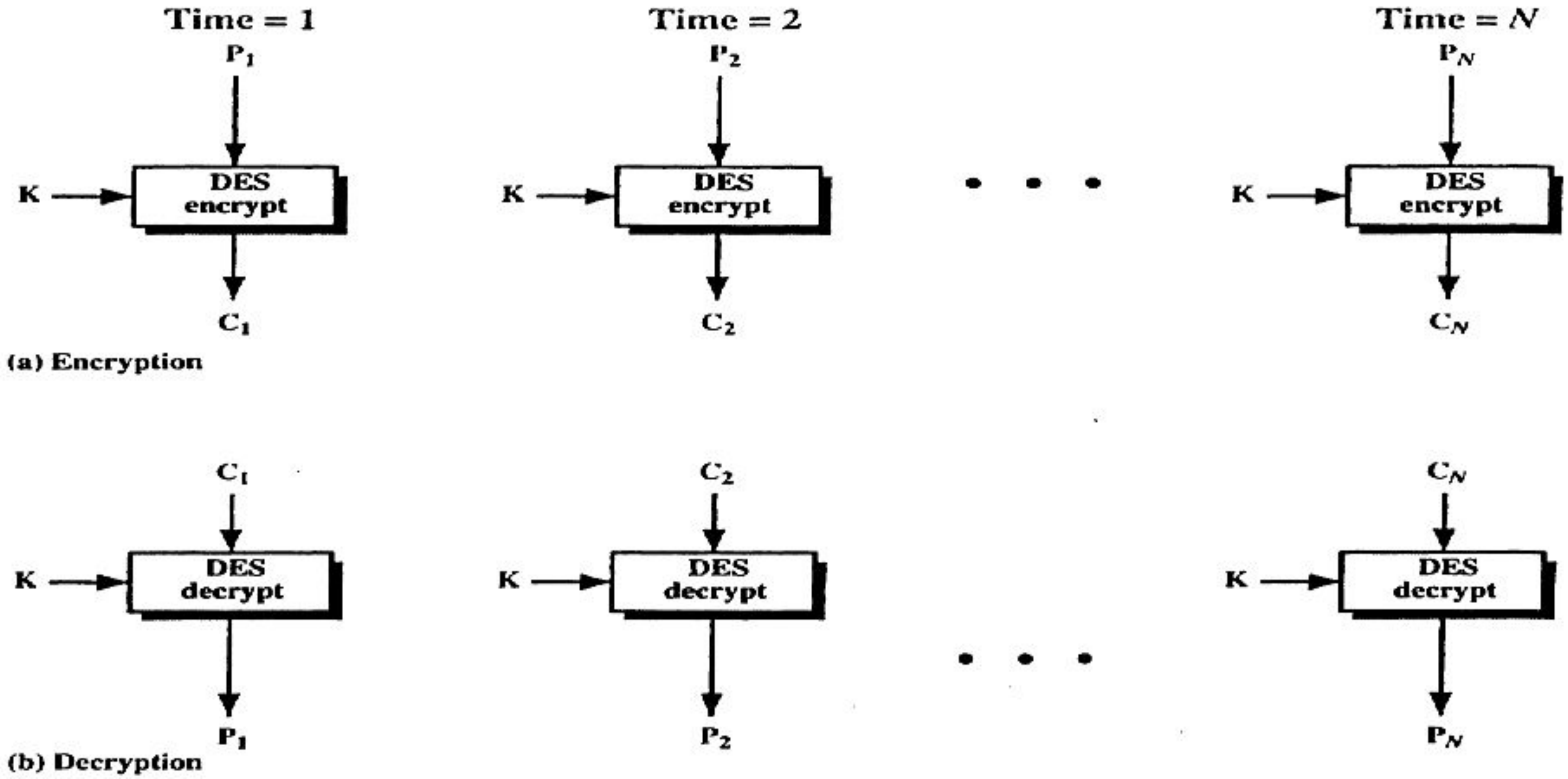


What is the necessity of block cipher modes of operation? List out the advantages and disadvantages of ***output feedback*** mode. (4)

BLOCK CIPHER MODES of operation

- To use DES in varies application, five modes of operation are defined
  - Electronic Codebook mode
  - Cipher Block Chaining Mode
  - Cipher Feedback mode
  - Output Feedback Mode
  - Counter Mode

Electronic Codebook Mode



- message is broken into independent blocks that are encrypted

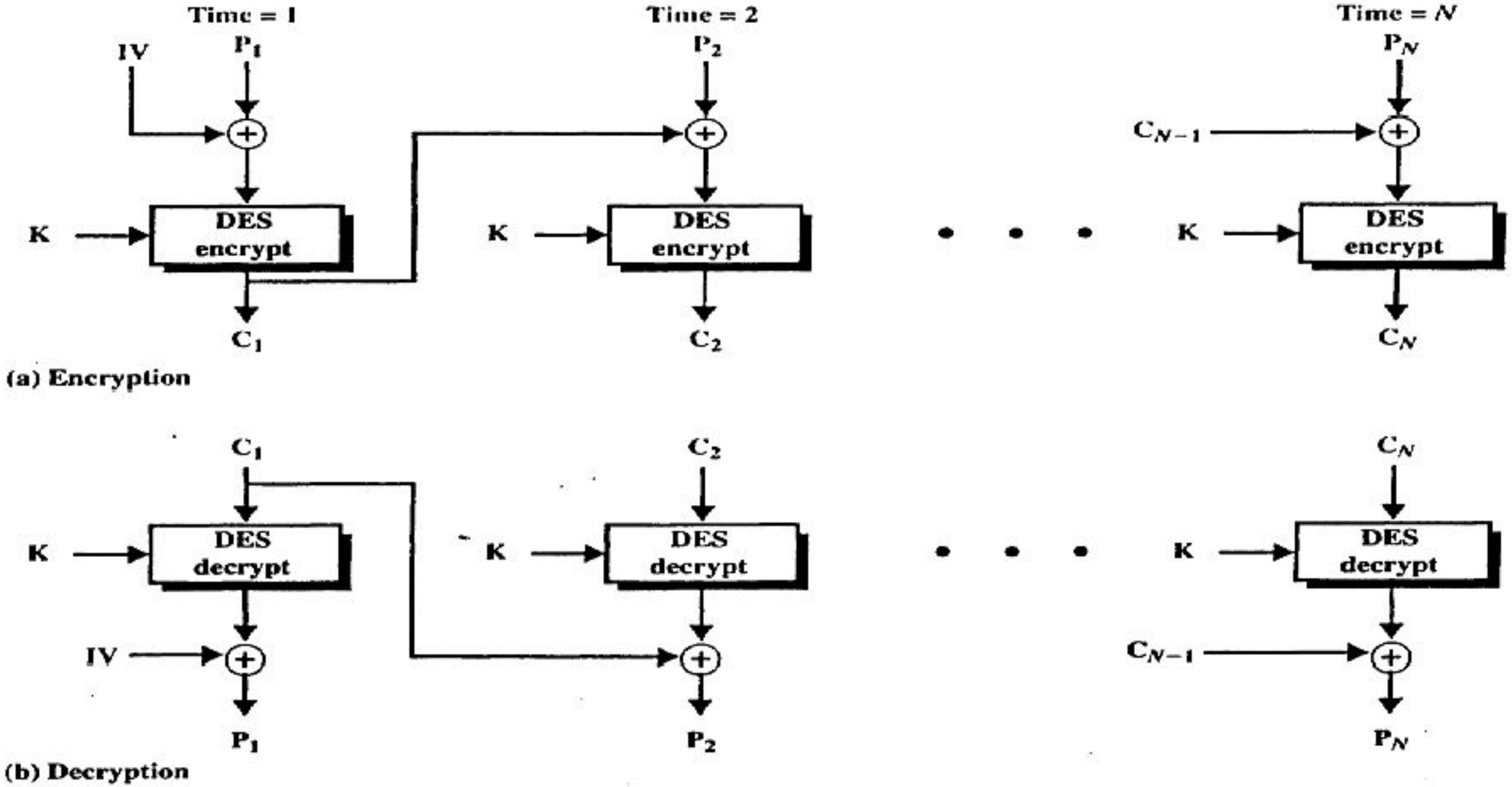
   each block is a value which is substituted, like a codebook, hence name

   each block is encoded independently of the other blocks

  Ci = EK(Pi)

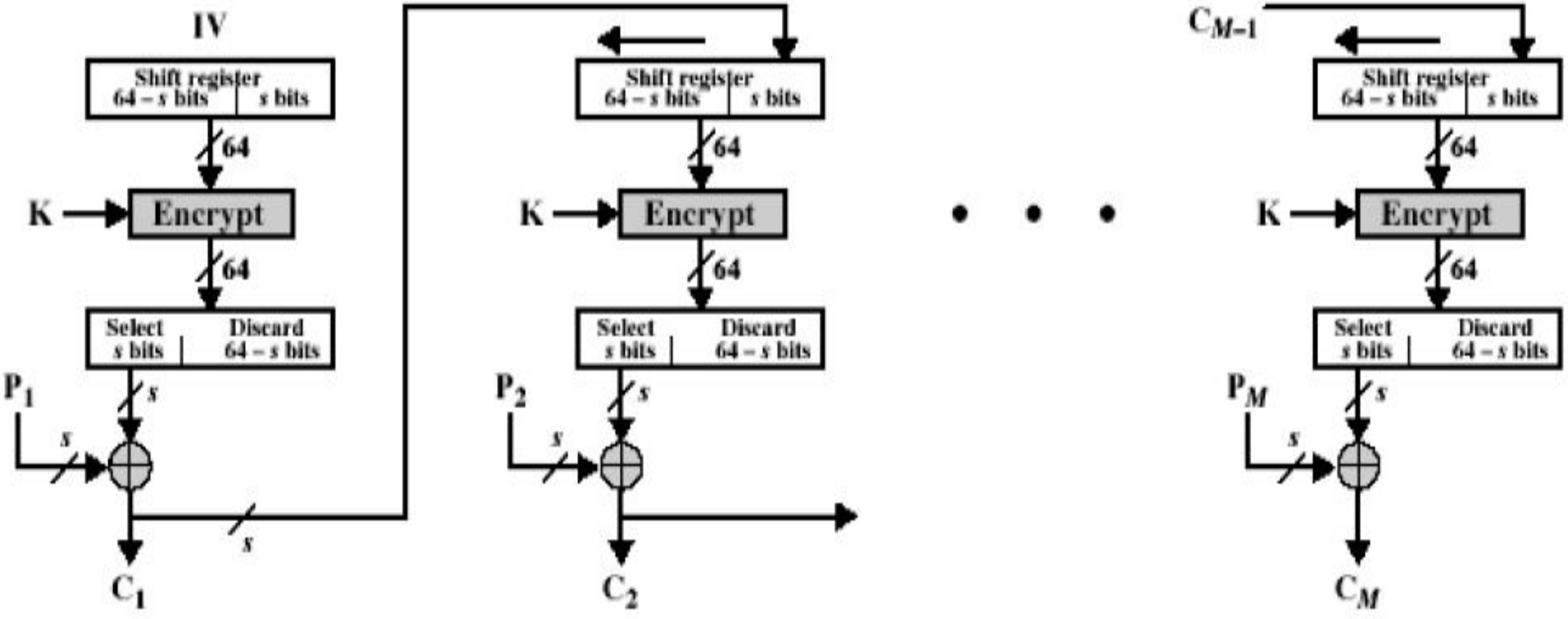
   uses: secure transmission of sing

Cipher Block Chaining Mode



- message is broken into blocks
   linked together in encryption operation
   each previous cipher block is chained with current plaintext block, hence name
   use Initial Vector (IV) to start process
  o Ci = EK(Pi XOR Ci-1)
  o C
  -1 = IV
   IV prevents same P from making same C
   uses: bulk data encryption, authentication

Cipher Feedback Mode



(a) Encryption

- 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 bits (1,8, 64 or 128 etc) to be feed back

  ● denoted CFB-1, CFB-8, CFB-64, CFB-128, etc.

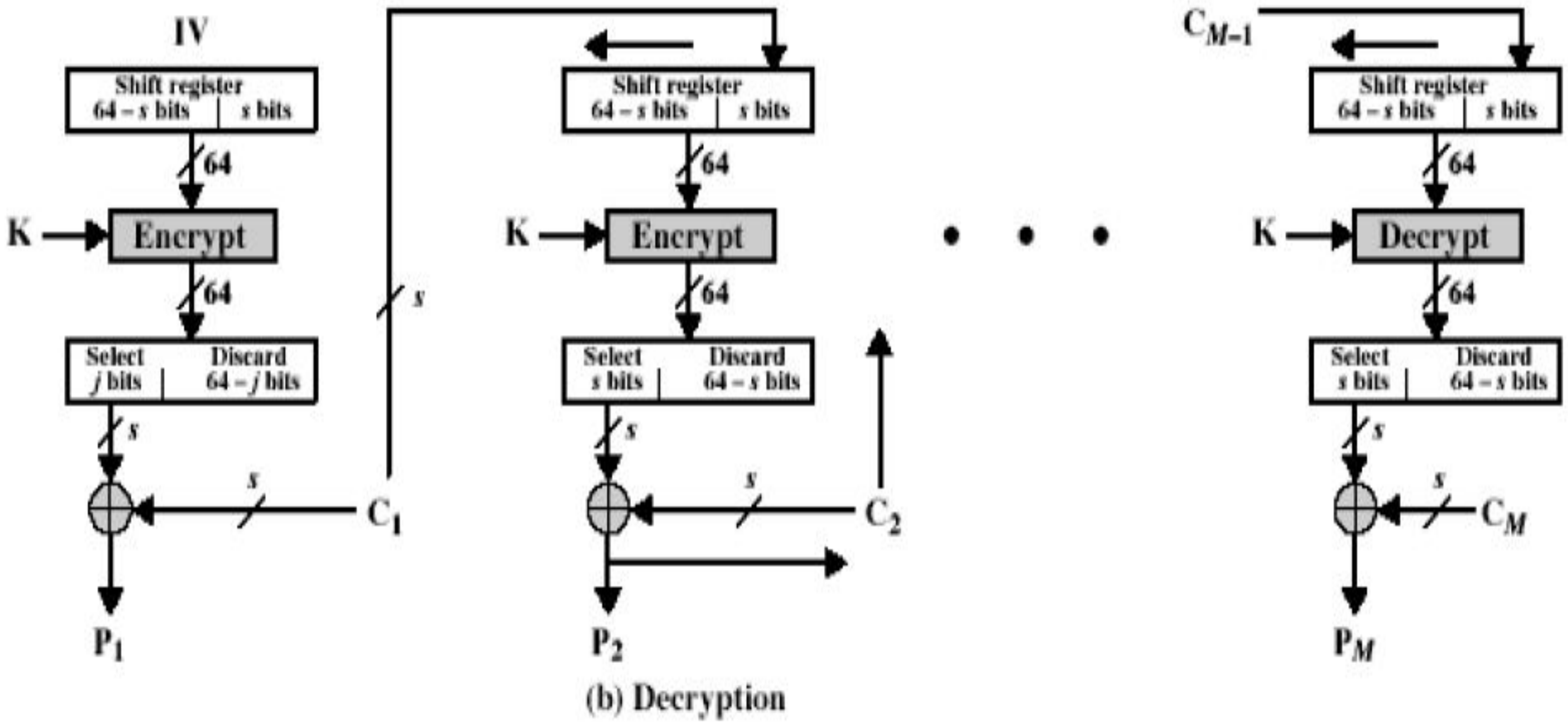
   most efficient to use all bits in block (64 or 128)

  Ci = Pi XOR EK(Ci-1)

  C-1 = IV

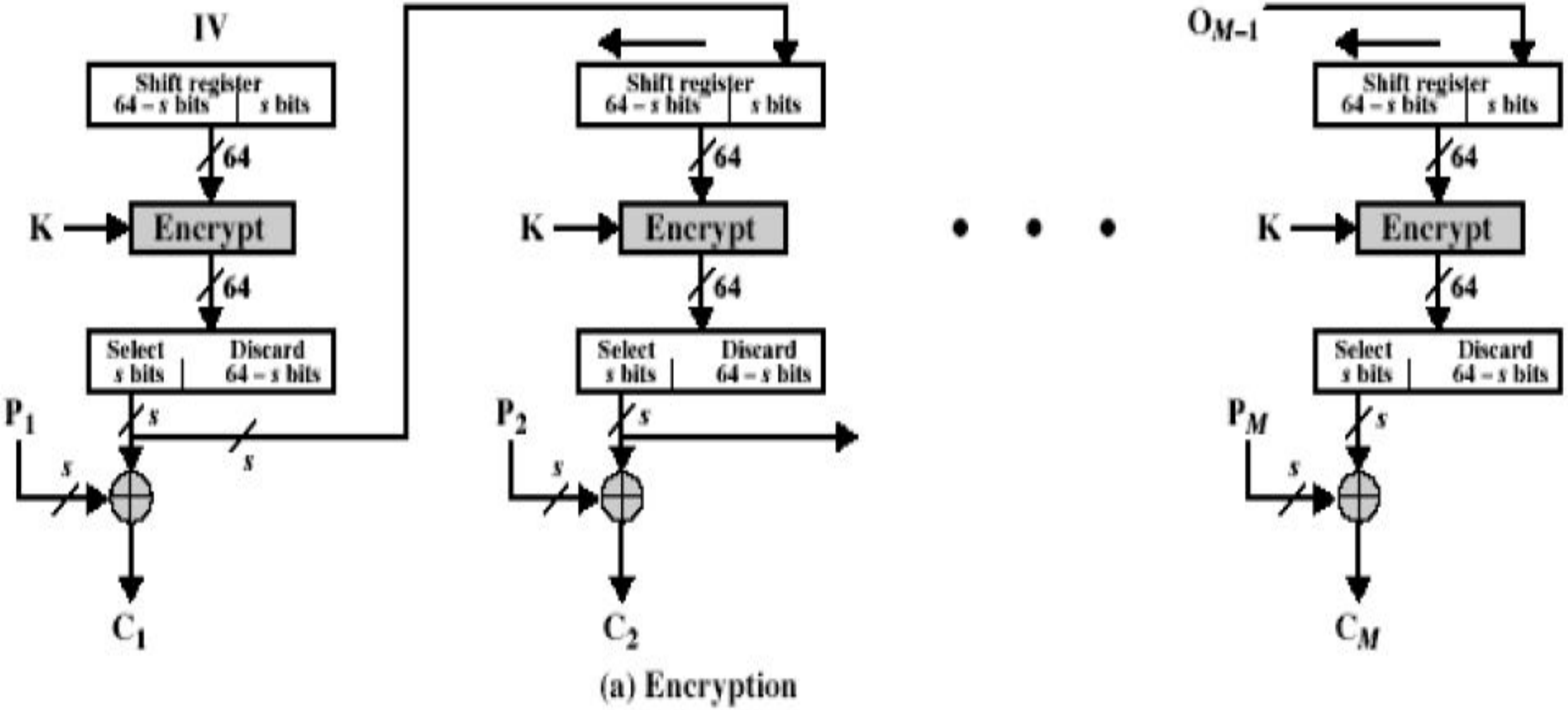
   uses: stream data encryption, authentication

Cipher Feedback Mode



(b) Decryption

Output Feedback Mode

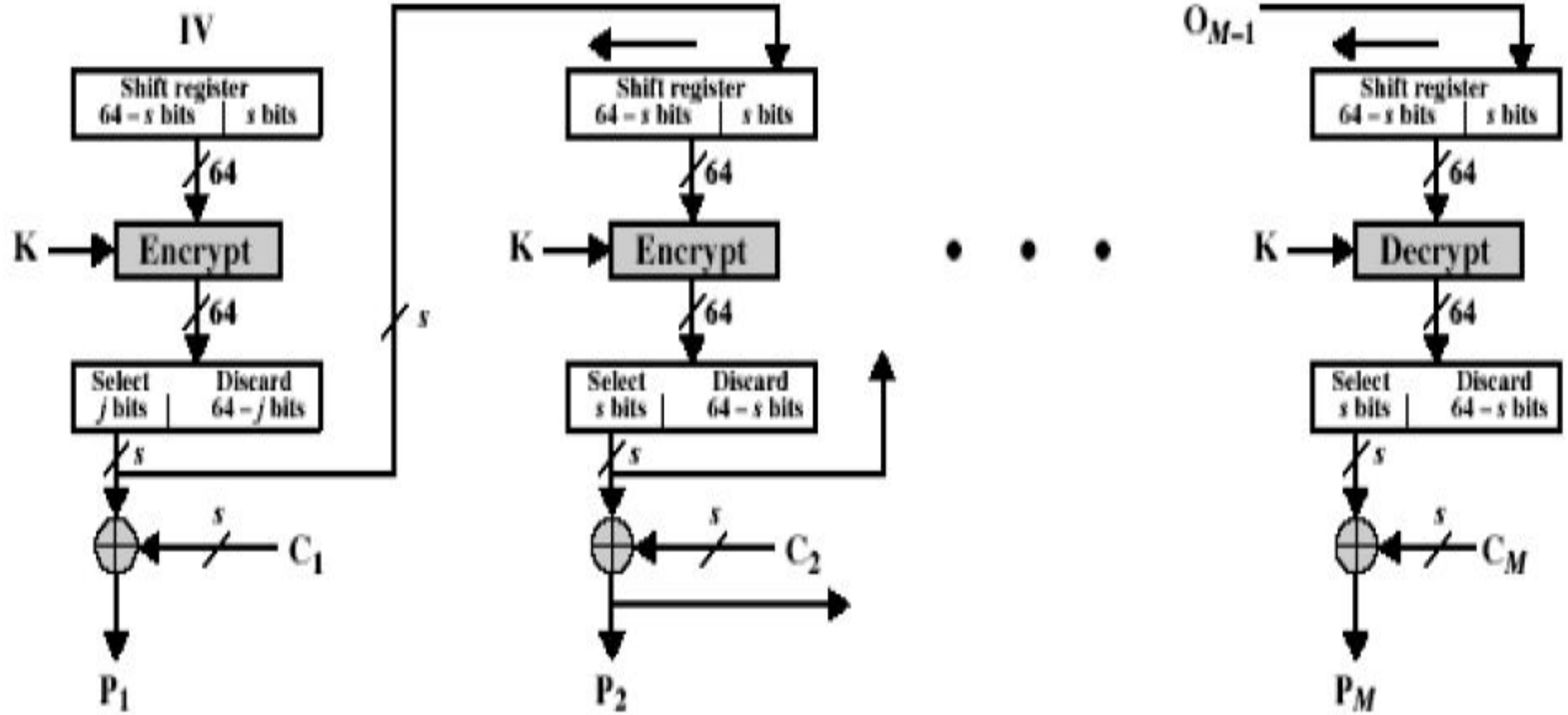


(a) Encryption

- message is treated as a stream of bits

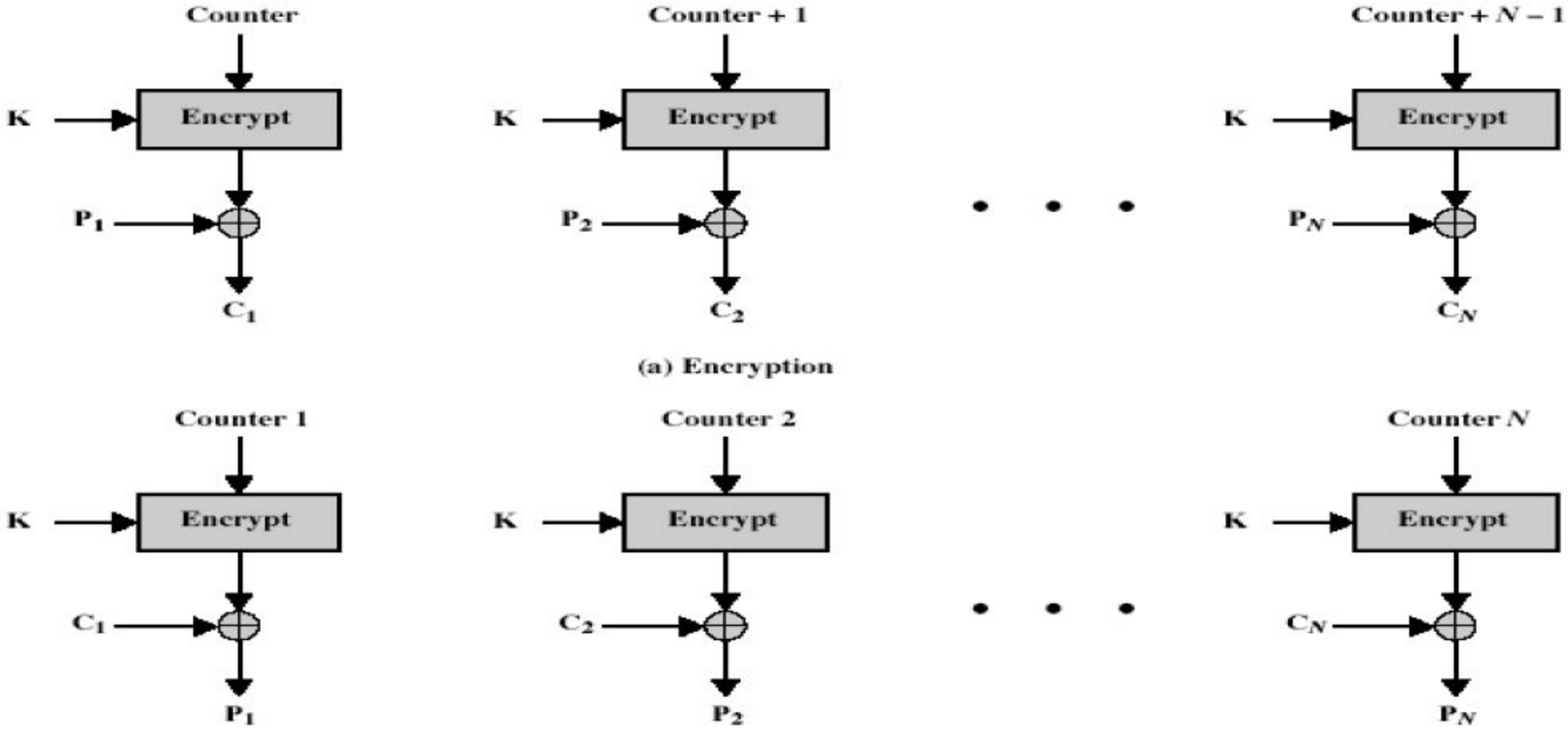
 output of cipher is added to message
 output is then feed back (hence name)
Oi = EK(Oi-1)
Ci = Pi XOR Oi
O
-1 = IV
 feedback is independent of message
 can be computed in advance

Output Feedback Mode



(b) Decryption

Counter Mode



(a) Encryption

(b) Decryption

- a “new” mode, though proposed early on
   similar to OFB but encrypts counter value rather than any feedback value
  Oi = EK(i)
  Ci = Pi XOR Oi
   must have a different key & counter value for every plaintext block (never reused)
   uses: high-speed network encryptions

Advantages of Counter Mode

- Hardware Efficiency
- Software efficiency
- Preprocessing
- Random access
- Provable Security
- Simplicity