

# **ATME COLLEGE OF ENGINEERING**

**13<sup>th</sup> KM Stone, Bannur Road, Mysore - 560 028**



# **A T M E**

**College of Engineering**

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **- CYBER SECURITY**



## **LECTURE NOTES**

**COURSE: CRYPTOGRAPHY AND NETWORK SECURITY**

**COURSE CODE: BCY602**

**SEMESTER: VI**

**COURSE COORDINATOR: Mrs. Suhasini**

**(ACADEMIC YEAR 2025-26)**

## INSTITUTIONAL MISSION AND VISION

### **Objectives**

- To provide quality education and groom top-notch professionals, entrepreneurs and leaders for different fields of engineering, technology and management.
- To open a Training-R & D-Design-Consultancy cell in each department, gradually introduce doctoral and postdoctoral programs, encourage basic & applied research in areas of social relevance, and develop the institute as a center of excellence.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To develop academic, professional and financial alliances with the industry as well as the academia at national and transnational levels.
- To cultivate strong community relationships and involve the students and the staff in local community service.
- To constantly enhance the value of the educational inputs with the participation of students, faculty, parents and industry.

### **Vision**

Development of academically excellent, culturally vibrant, socially responsible and globally competent human resources.

### **Mission**

- To keep pace with advancements in knowledge and make the students competitive and capable at the global level.
- To create an environment for the students to acquire the right physical, intellectual, emotional and moral foundations and shine as torch bearers of tomorrow's society.
- To strive to attain ever-higher benchmarks of educational excellence.

## Module – 1

### Chapter - 1

#### 1.1 A Model for Network Security

Data is transmitted over network between two communicating parties, who must cooperate for the exchange to take place. A logical information channel is established by defining a route through the internet from source to destination by use of communication protocols by the two parties. Whenever an opponent presents a threat to confidentiality, authenticity of information, security aspects come into play. Two components are present in almost all the security providing techniques.

A security-related transformation on the information to be sent making it unreadable by the opponent, and the addition of a code based on the contents of the message, used to verify the identity of sender.

Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

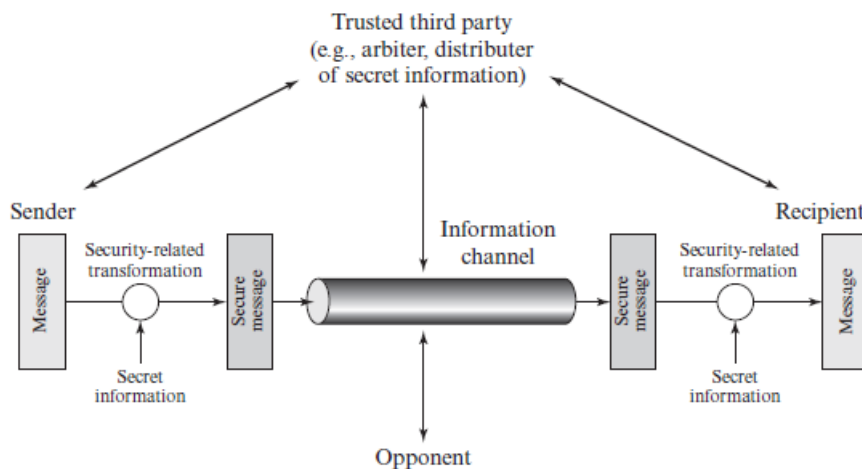


Figure 1.5 Model for Network Security

A trusted third party may be needed to achieve secure transmission. It is responsible for distributing the secret information to the two parties, while keeping it away from any opponent. It also may be needed to settle disputes between the two parties regarding authenticity of a message transmission. The general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose
2. Generate the secret information to be used with the algorithm

3. Develop methods for the distribution and sharing of the secret information
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

A general model of these other situations is illustrated by Figure 1.6, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers, who attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. The intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers.

Programs can present two kinds of threats:

- **Information access threats:** Intercept or modify data on behalf of users who should not have access to that data.
- **Service threats:** Exploit service flaws in computers to inhibit use by legitimate users.

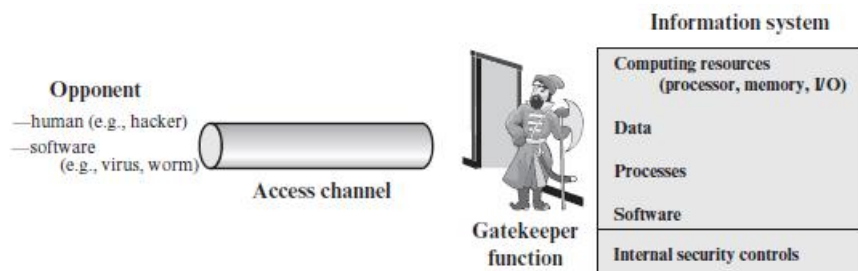


Figure 1.6 Network Access Security Model

Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software.

The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.6). The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks.

The second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

## Chapter – 2

### Classical Encryption Techniques

#### 2.1 Symmetric Cipher Model

##### Terminologies

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**. The areas of cryptography and cryptanalysis together are called **cryptology**.

A symmetric encryption scheme has five ingredients (Figure 3.1):

- **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 cipher text and the secret key and produces the original plaintext.

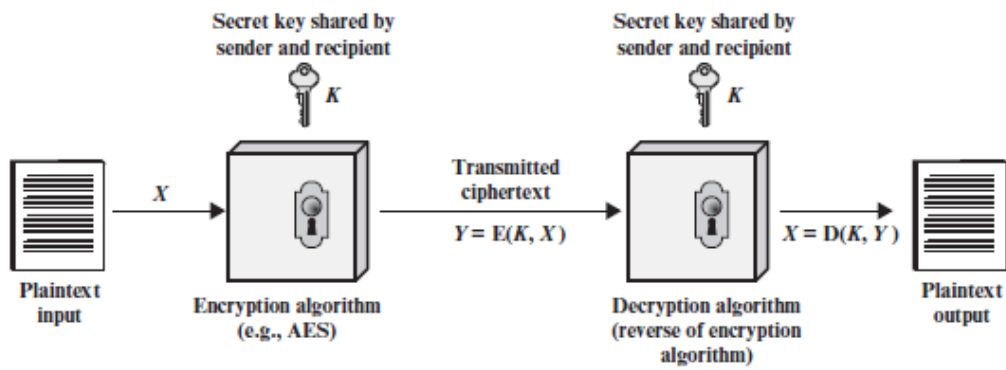


Figure 3.1 Simplified Model of Symmetric Encryption

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.

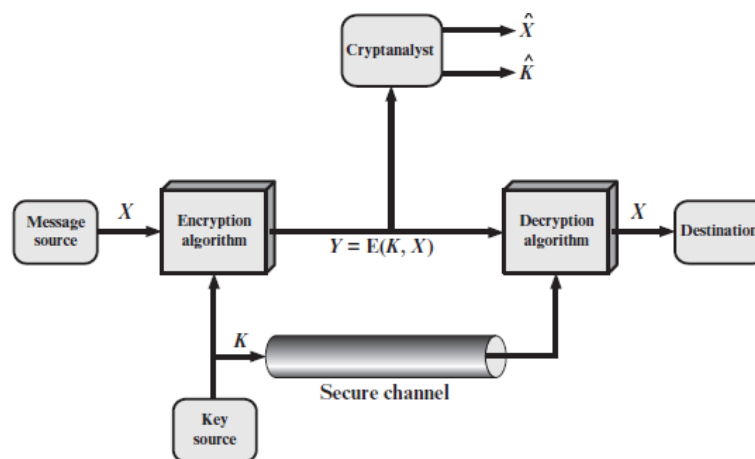


Figure 3.2 Model of Symmetric Cryptosystem

With the message  $X$  and the encryption key  $K$  as input, the encryption algorithm forms the ciphertext  $Y = [Y_1, Y_2, \dots, Y_N]$ . We can write this as  $Y = E(K, X)$ . This notation indicates that 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:  $X = D(K, Y)$ . An opponent, observing  $Y$  but not having access  $K$  to  $X$  or, 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  $\hat{X}$ . Often, however, the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover  $K$  by generating an estimate  $\hat{K}$ .

## Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. **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.
2. **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.
3. **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

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.

**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.

The various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst is given below.

Type of Attack	Known to Cryptanalyst
Ciphertext Only	<ul style="list-style-type: none"> <li>■ Encryption algorithm</li> <li>■ Ciphertext</li> </ul>
Known Plaintext	<ul style="list-style-type: none"> <li>■ Encryption algorithm</li> <li>■ Ciphertext</li> <li>■ One or more plaintext-ciphertext pairs formed with the secret key</li> </ul>
Chosen Plaintext	<ul style="list-style-type: none"> <li>■ Encryption algorithm</li> <li>■ Ciphertext</li> <li>■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li> </ul>
Chosen Ciphertext	<ul style="list-style-type: none"> <li>■ Encryption algorithm</li> <li>■ Ciphertext</li> <li>■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li> </ul>
Chosen Text	<ul style="list-style-type: none"> <li>■ Encryption algorithm</li> <li>■ Ciphertext</li> <li>■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key</li> <li>■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key</li> </ul>

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.

- 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.

A **brute-force attack** involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained.

## 2.2 Substitution Techniques

The two basic building blocks of all encryption techniques are substitution and transposition. A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.

### 1. Caesar Cipher

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

Then the algorithm can be expressed as follows.

For each plaintext letter  $p$ , substitute the cipher text letter  $C$ ,  $C = E(3, p) = (p + 3) \bmod 26$



A shift may be of any amount, so that the general Caesar algorithm is  $C = E(k, p) = (p + k) \bmod 26$  where  $k$  takes on a value in the range 1 to 25.

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

If it is known that a given cipher text is a Caesar cipher, then a brute-force cryptanalysis is easily performed: Simply try all the 25 possible keys. 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.

Below Figure shows the results of applying this strategy to the example ciphertext. In this case, the plaintext leaps out as occupying the third line.

KEY	PHHW	PH	DIWHU	WKH	WRJD	SDUWB
1	oggv	og	chvgt	vjg	vqic	retva
2	nffu	nf	bgufs	uif	uphb	qbsuz
3	meet	me	after	the	toga	party
4	ldds	ld	zesdq	sgd	snfz	ozqsx
5	kecr	kc	ydrpc	rfe	rmey	nyprw
6	jbbq	jb	xcqbo	qeb	qldx	msoqv
7	iaap	ia	wbpan	pda	pkcw	lwnpu
8	hzzo	hz	vaomz	ocz	objv	kvmot
9	gyyn	gy	uznyl	nby	niau	julns
10	fxxm	fx	tymxk	max	mhzt	itkmr
11	ewwl	ew	sxlwj	lzw	lgys	hsjlg
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	nsgrg	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	vnnv	vn	jocna	cqn	cxpj	yjach
21	ummb	um	inbmz	bpm	bwoi	xizbg
22	tlla	tl	hmaly	aol	avnh	whyaf
23	skkz	sk	glzlx	znk	zumg	vgxze
24	rjjy	rj	fkyjw	ymj	ytlf	ufwyd
25	qiix	qi	ejxiv	xli	xske	tevxv

Figure 3.3 Brute-Force Cryptanalysis of Caesar Cipher

## 2. 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. A **permutation** of a finite set of elements is an ordered sequence of all the elements of, with each element appearing exactly once. For example, if  $S = \{a, b, c\}$ , there are six permutations of:

abc, acb, bac, bca, cab, cba

In general, there are  $n!$  permutations of a set of 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.

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  $4 \times 10^{26}$  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.

The ciphertext to be solved is

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ

VUEPHZHMZSHZOWSFPAPPDTSVPQUZWMYXUZHUSX

EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

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 3.5. 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				

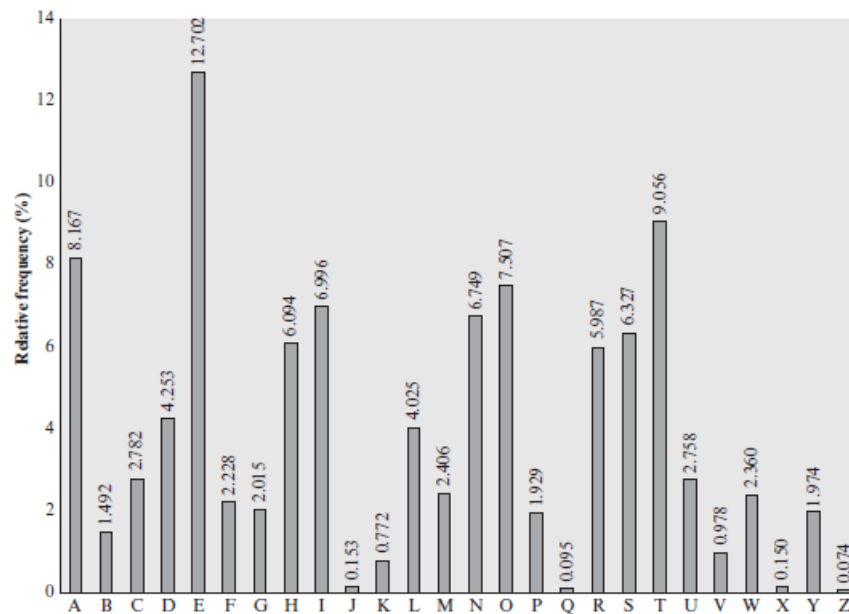


Figure 3.5 Relative Frequency of Letters in English Text

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, and H are all of relatively 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}.

A powerful tool is to look at the frequency of two-letter combinations, known as **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). 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 tht. If so, Sequates with a. So far, then, we have

```

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSAIZ
  t a       e e t e a t h a t e e a       a
VUEPHZHMZSHZOWSFPAPDTSVPQUZWMXUZHXS
  e t   t a t h a e e e a e t h   t a
EPYEPDPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
  e e e t a t e   t h e   t

```

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.

### 3. 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  $5 \times 5$  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 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 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  $26 \times 26 = 676$  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.

## 4. 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  $m$  cipher text letters.

The substitution is determined by  $m$  linear equations in which each character is assigned a numerical value ( $a = 0, b = 1 \dots z = 25$ ).

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

$$c_1 = (k_{11}P_1 + k_{12}P_2 + k_{13}P_3) \bmod 26$$

$$c_2 = (k_{21}P_1 + k_{22}P_2 + k_{23}P_3) \bmod 26$$

$$c_3 = (k_{31}P_1 + k_{32}P_2 + k_{33}P_3) \bmod 26$$

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

Where  $\mathbf{C}$  and  $\mathbf{P}$  are column vectors of length 3, representing the plaintext and cipher text, and  $\mathbf{K}$  is a  $3 \times 3$  matrix, representing the encryption key. Operations are performed in mod 26.

$$\mathbf{P} = \mathbf{D}(\mathbf{K}, \mathbf{C}) = \mathbf{CK}^{-1} \bmod 26 = \mathbf{PKK}^{-1} = \mathbf{P}$$

Problem solved in class refer it

## 5. 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.

### *VIGENÈRE CIPHER*

The best known, and one of the simplest, polyalphabetic ciphers is 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.

Express the Vigenère cipher in the following manner. Assume a sequence of plaintext letters and a key consisting of the sequence of letters, where typically  $k < n$ . The sequence of ciphertext letters is calculated as follows

$$\begin{aligned}
 C &= C_0, C_1, C_2, \dots, C_{n-1} = E(K, P) = E[(k_0, k_1, k_2, \dots, k_{m-1}), (p_0, p_1, p_2, \dots, p_{n-1})] \\
 &= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \dots, (p_{m-1} + k_{m-1}) \bmod 26, \\
 &\quad (p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \dots, (p_{2m-1} + k_{m-1}) \bmod 26, \dots
 \end{aligned}$$

Thus, the first letter of the key is added to the first letter of the plaintext, mod 26, the second letters are added, and so on through the first letters of the plaintext. For the next letters of the plaintext, the key letters are repeated. This process continues until all of the plaintext sequence is encrypted. A general equation of the encryption process is

$$C_i = (p_i + k_{i \bmod m}) \bmod 26$$

Decryption is a generalization of Equation

$$p_i = (C_i - k_{i \bmod m}) \bmod 26$$

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword is *deceptive*, the message “we are discovered save yourself” is encrypted as

key: *deceptivedeceptivedeceptive*

plaintext: *wearediscoveredsaveyourself*

ciphertext: *ZICVTWQNGRZGVTWAVZHCQYGLMGJ*

## Security Value

Vigenere Cipher was designed by tweaking the standard Caesar cipher to reduce the effectiveness of cryptanalysis on the ciphertext and make a cryptosystem more robust. It is significantly **more secure than a regular Caesar Cipher**.

In the history, it was regularly used for protecting sensitive political and military information. It was referred to as the **unbreakable cipher** due to the difficulty it posed to the cryptanalysis.

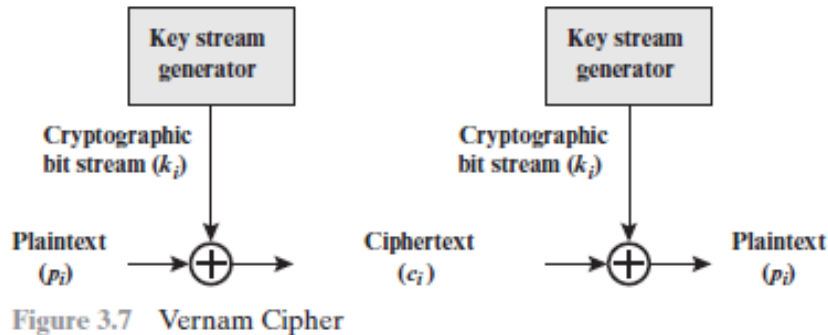
## Variants of Vigenere Cipher

There are two special cases of Vigenere cipher –

- The keyword length is same as plaintext 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

The ultimate defense against such a cryptanalysis is to choose a keyword that is as long as the plaintext and has no statistical relationship to it. Such a system was introduced by an AT&T engineer named Gilbert Vernam in 1918.



The system can be expressed succinctly as follows

$$c_i = p_i \oplus k_i$$

where

$p_i$  =  $i$ th binary digit of plaintext

$k_i$  =  $i$ th binary digit of key

$c_i$  =  $i$ th binary digit of ciphertext

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

Thus, the ciphertext is generated by performing the bitwise XOR of the plaintext and the key. Because of the properties of the XOR, decryption simply involves the same bitwise operation:

$$p_i = c_i \oplus k_i$$

## 6. One Time Pad

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.

The one-time pad offers complete security but, in practice, has two fundamental difficulties:

1. There is the practical problem of making large quantities of random keys.
2. Even more daunting is the problem of key distribution and protection. For every message to be sent, a key of equal length is needed by both sender and receiver.

### Security Value

Let us compare Shift cipher with one-time pad.

### Shift Cipher – Easy to Break

In case of Shift cipher, the entire message could have had a shift between 1 and 25. This is a very small size, and very easy to brute force. However, with each character now having its own individual shift between 1 and 26, the possible keys grow exponentially for the message.

### **One-time Pad – Impossible to Break**

Let us say, we encrypt the name “point” with a one-time pad. It is a 5 letter text. To break the cipher text by brute force, you need to try all possibilities of keys and conduct computation for  $(26 \times 26 \times 26 \times 26 \times 26) = 26^5 = 11881376$  times. That’s for a message with 5 alphabets. Thus, for a longer message, the computation grows exponentially with every additional alphabet. This makes it computationally impossible to break the cipher text by brute force.

## **2.3 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.

Various other techniques have been used historically, some of them are:

- **Character marking** – selected letters of printed or typewritten text are overwritten in pencil. The marks are ordinarily not visible unless the paper is held to 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 in front of the light.
- **Typewritten correction ribbon** – used between the lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light.

Figure shows an example in which a subset of the words of the overall message is used to convey the hidden message.



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 16t proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

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 a 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.

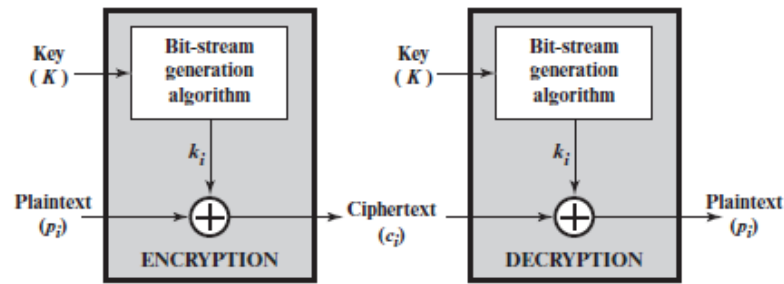
The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered. Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to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

## Chapter – 3

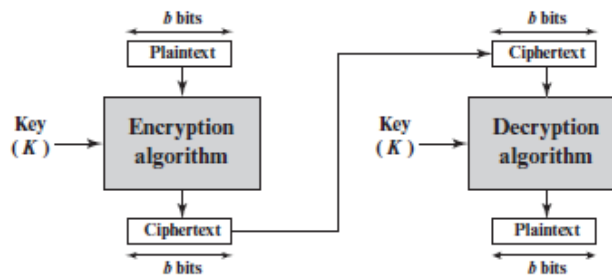
### 3.1 Block Ciphers and the Data Encryption Standard

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 auto keyed 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. Using some of the modes of operation, a block cipher can be used to achieve the same effect as a stream cipher. 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.



(a) Stream cipher using algorithmic bit-stream generator



b) Block Cipher

### 3.2 Feistel Block 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  $b$  bits, allowing a total of  $2^k$  possible transformations, rather than the  $2^b$  transformations available with the ideal block cipher.

In particular, Feistel proposed the use of a cipher that alternates substitutions and permutations, where these terms are defined as follows:

- **Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
- **Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

**Diffusion:** A cryptographic technique that seeks to obscure the statistical structure of the plaintext by spreading out the influence of each individual plaintext digit over many cipher text digits.

**Confusion:** A cryptographic technique that seeks to make the relationship between the statistics of the cipher text and the value of the encryption key as complex as possible. This is achieved by the use of a complex scrambling algorithm that depends on the key and the input.

The inputs to the encryption algorithm are a plaintext block of length  $2w$  bits and a key. The plaintext block is divided into two halves  $L_0$  and  $R_0$ . The two halves of the data pass through rounds of processing and then combine to produce the cipher text block. Each round  $i$  has as inputs  $L_{i-1}$  and  $R_{i-1}$  derived from the previous round, as well as a sub key  $K_i$  derived from the overall  $K$ . In general, the sub keys  $K_i$  are different from  $K$  and from each other. 16 rounds are used, although any number of rounds could be implemented.

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 sub key  $K_i$ . Another way to express this is to say that  $F$  is a function of right-half block of  $w$  bits and a sub key of  $y$  bits, which produces an output value of length  $w$  bits:  $F(R_{Ei}, K_{i+1})$ .

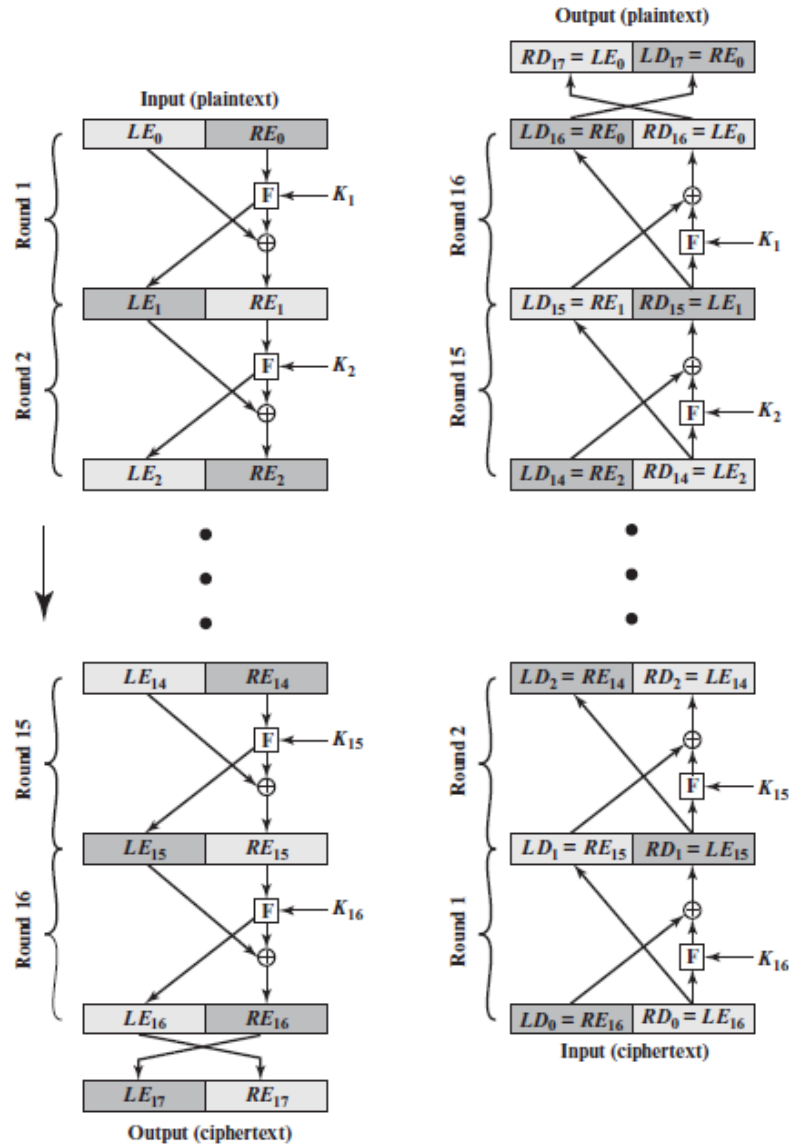


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

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 trade off 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.
- **Sub key generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function F:** 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  $K_i$  in reverse order. That is, use  $K_n$  in the first round,  $K_{n-1}$  in the second round, and so on until  $K_1$  is used in the last round.

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.

$$\begin{aligned} LE_{16} &= RE_{15} \\ RE_{16} &= LE_{15} \oplus F(RE_{15}, K_{16}) \end{aligned}$$

On the decryption side,

$$\begin{aligned} LD_1 &= RD_0 = LE_{16} = RE_{15} \\ RD_1 &= LD_0 \oplus F(RD_0, K_{16}) \\ &= RE_{16} \oplus F(RE_{15}, K_{16}) \\ &= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16}) \end{aligned}$$

The XOR has the following properties:

$$\begin{aligned} [A \oplus B] \oplus C &= A \oplus [B \oplus C] \\ D \oplus D &= 0 \\ E \oplus 0 &= E \end{aligned}$$

Thus, we have  $LD1 = RE15$  and  $RD1 = LE15$ . Therefore, the output of the first round of the decryption process is  $LE15||RE15$ , 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,

$$\begin{aligned}LE_i &= RE_{i-1} \\ RE_i &= LE_{i-1} \oplus F(RE_{i-1}, K_i)\end{aligned}$$

Rearranging terms:

$$\begin{aligned}RE_{i-1} &= LE_i \\ LE_{i-1} &= RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i)\end{aligned}$$

### 3.3 Data Encryption Standard(DES)

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Institute of Standards and Technology (NIST).

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.

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.

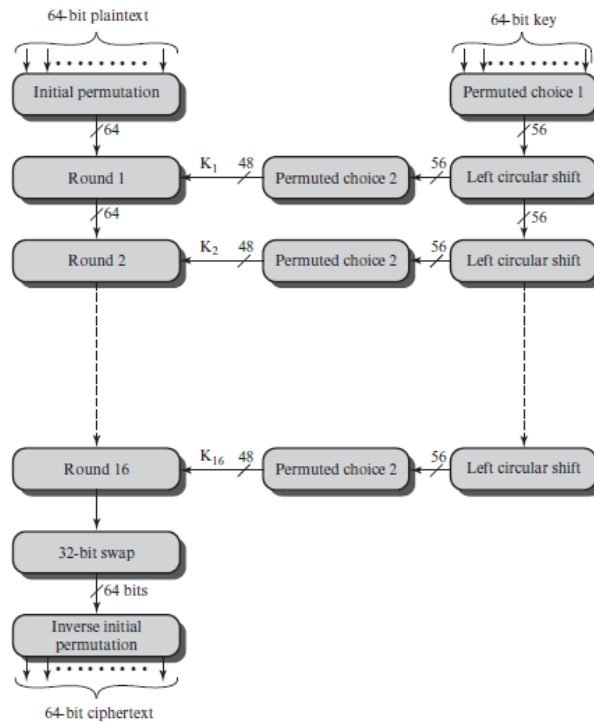


Figure 4.5 General Depiction of DES Encryption Algorithm

In the left-hand side of the figure, the processing of the plaintext proceeds in three phases.

1. 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.
2. 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 pre output.
3. Finally, the pre output 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

The right-hand portion 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 (K<sub>i</sub>) 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 IP:

First step of the data computation, IP reorders the input data bits.

#### Details of Single Round:

- The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right).

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

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

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

- The round key  $K_i$  is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the R bits.
- The resulting 48 bits are XORed with  $K_i$ . This 48-bit result passes through a substitution function that produces a 32-bit output.

### Key Generation

We see that a 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64, every eighth bit is ignored. The key is first subjected to a permutation governed by a table labeled Permuted Choice One. The resulting 56-bit key is then treated as two 28-bit quantities, labelled  $C_0$  and  $D_0$ . At each round,  $C_{i-1}$  and  $D_{i-1}$  are separately subjected to a circular left shift, or rotation, of 1 or 2 bits. These shifted values serve as input to the next round. They also serve as input to Permuted Choice Two, which produces a 48-bit output that serves as input to the function  $F(R_{i-1}, K_i)$ .

### DES Example

Plaintext:	02468aceeca86420
Key:	0f1571c947d9e859
Ciphertext:	da02ce3a89ecac3b



Round	$K_i$	$L_i$	$R_i$
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP <sup>-1</sup>		da02ce3a	89ecac3b

*Note: DES subkeys are shown as eight 6-bit values in hex format*

### Avalanche effect

- A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext. This is referred to as the avalanche effect. If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched.

## 3.4 The Strength of DES

### The Use of 56-Bit Keys

- With a key length of 56 bits, there are 256 possible keys, which is approximately  $7.2 \times 10^{16}$ . So a brute-force attack appears impractical.
- Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.
- If the message is just plain text in English, then the task of recognizing English would have to be automated.
- If the text message has been compressed before encryption, then recognition is more difficult. And if the message is some more general type of data, such as a numerical file, and this has been compressed, the problem becomes even more difficult to automate.
- Thus, to supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed.

### **The Nature of the DES Algorithm**

Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.

The focus of concern has been on the eight substitution tables or S-boxes, that are used in each iteration.

- Because the design criteria for these 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.
- No one has so far succeeded in discovering the supposed fatal weaknesses in the S boxes.

### **Timing Attacks**

A timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts. This is a long way from knowing the actual key, but it is an intriguing first step. A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.

- DES appears to be fairly resistant to a successful timing attack

## **3.5 Block Cipher Design Principles**

There are three critical aspects of block cipher design:

- The number of rounds,
- Design of the function  $F$ ,
- Key scheduling.

### **The number of rounds**

- The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak  $F$ .
- 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.
- If DES had 15 or fewer rounds, differential cryptanalysis would require less effort than brute-force key search.

### **Design of the function $F$**

- The function  $F$  provides the element of confusion in a Feistel cipher, want it to be difficult to “unscramble” the substitution performed by  $F$ .
- One obvious criterion is that  $F$  be nonlinear. The more nonlinear  $F$ , the more difficult any type of cryptanalysis will be.
- One of the most intense areas of research in the field of symmetric block ciphers is that of S-box design. Would like any change to the input vector to an S-box to result in random-looking changes to the output. The relationship should be nonlinear and difficult to approximate with linear functions.

### **Key scheduling**

- A final area of block cipher design, and one that has received less attention than S-box design, is the key schedule algorithm. With any Feistel block cipher, the key schedule is used to generate a subkey for each round.
- Would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key. The key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion.