Chapter 1
Information and Network Security Concepts

1. Cybersecurity, Information Security, And Network Security

- **Cybersecurity** is the protection of information that is stored, transmitted, and processed in a networked system of computers, other digital devices, and network devices and transmission lines, including the Internet. Protection encompasses confidentiality, integrity, availability, authenticity, and accountability. Methods of protection include organizational policies and procedures, as well as technical means such as encryption and secure communications protocols.
    - **Information security:** This term refers to preservation of confidentiality, integrity, and availability of information. In addition, other properties, such as authenticity, accountability, nonrepudiation, and reliability can also be involved.
    - **Network security:** This term refers to protection of networks and their service from unauthorized modification, destruction, or disclosure, and provision of assurance that the network performs its critical functions correctly and there are no harmful side effects.

Security Objectives

The cybersecurity definition introduces three key objectives that are at the heart of information and network security:

- **Confidentiality:** This term covers two related concepts:
    - **Data confidentiality:** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.
    - **Privacy:** Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.
- **Integrity:** This term covers two related concepts:
    - **Data integrity:** Assures that data (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner. This concept also encompasses **data authenticity**, which means that a digital object is indeed what it claims to be or what it is claimed to be, and nonrepudiation, which is assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of

    the sender's identity, so neither can later deny having processed the information.
    - **System integrity:** Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.
- **Availability:** Assures that a system performs its intended function in an unimpaired manner.

These three concepts form what is often referred to as the **CIA triad**. Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture (Figure 1.1). Two of the most commonly mentioned are as follows:

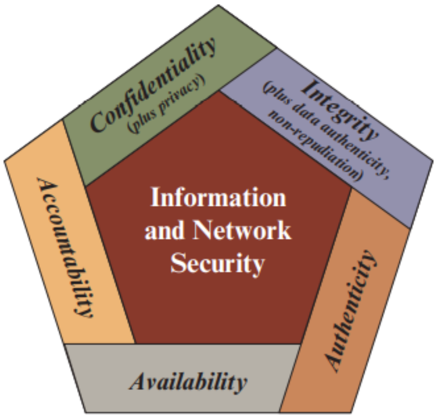


Figure 1.1 Essential Information and Network Security Objectives

- **Authenticity:** verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
- **Accountability:** The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence[1], fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

[1] The concept of deterrence is that people choose not to act in a specific way because they are afraid of what will happen if they do.

2. The OSI Security Architecture

ITU-T Recommendation X.800, Security Architecture for OSI, defines a systematic approach to assess effectively the security needs of an organization and to evaluate and choose various security products and policies. The OSI security architecture focuses on **security attacks**, **mechanisms**, and **services**. These can be defined briefly as:

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Some examples of security mechanisms include:
    - **Encipherment (Encryption)** involves the use of algorithms to transform data into a form that can only be read by someone with the appropriate decryption key. Encryption can be used to protect data it is transmitted over a network, or to protect data when it is stored on a device.
    - **Digital signature** Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery.
    - **Traffic padding** is a technique used to add extra data to a network traffic stream in an attempt to obscure the true content of the traffic and make it more difficult to analyze.
    - **Routing control:** Enables selection of particular physically or logically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
- **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service. A security service is a capability that supports one or more of the security requirements (confidentiality, integrity, availability, authenticity, and accountability). The most important security services are summarized below.
    - **Authentication**: The authentication service is concerned with assuring that a communication is authentic. Two specific authentication services are defined in X.800:
        - **Peer entity authentication**: Provides for the corroboration of the identity of a peer entity in an association. Two entities are considered peers if they implement the same protocol in different systems; for example, two TCP modules in two communicating systems. Peer entity authentication is provided for use at the establishment of, or at times during the data transfer phase of, a connection. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.

        - **Data origin authentication:** Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail, where there are no ongoing interactions between the communicating entities.
    - **Access Control**: In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links.
    - **Data Confidentiality:** is the protection of transmitted data from passive attacks.
    - **Data Integrity:** As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service. On the other hand, a connectionless integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.
    - **Nonrepudiation:** prevents either sender or receiver from denying a transmitted message.
    - **Availability Service:** Availability is the property of a system, or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them).

In the literature, the terms threat and attack are commonly used, with the following meanings:

- **Threat:** Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.
- **Attack:** Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself.

A useful means of classifying security attacks, used both in X.800, is in terms of **passive attacks** and **active attacks**. *A passive attack* attempts to learn or make use of information from the system but does not affect system resources. *An active attack* attempts to alter system resources or affect their operation.

- **Passive attacks** are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the attacker is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.
- **Active attacks** involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: replay, masquerade, modification of messages, and denial of service.
    - A masquerade takes place when one entity pretends to be a different entity.
    - Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.
    - The denial of service prevents or inhibits the normal use or management of communication facilities.

CLASSICAL ENCRYPTION TECHNIQUES

Terminology:

Plaintext- original message

Ciphertext – coded message

Enciphering, encryption – process of converting from plaintext to ciphertext

Deciphering, decryption – restoring the plaintext from the ciphertext

Cryptography – area of study schemes for enciphering

Cryptographic system, cipher – scheme of enciphering

Cryptanalysis – techniques for deciphering a message without knowledge of the enciphering details

Cryptology – areas of cryptography and cryptanalysis

OUTLINE

1. SYMMETRIC CIPHER MODEL
2. SUBSTITUTION TECHNIQUES
3. TRANSPOSITION TECHNIQUES
4. ROTOR MACHINES
5. STEGANOGRAPHY

CRYPTOGRAPHY

Cryptographic systems (cryptosystems) are characterized by

1. The type of operations used for transforming plaintext to ciphertext (substitution, transposition). Fundamental requirement – no information be lost

2. The number of keys used (1 key – symmetric, single-key, secret-key; 2 keys – asymmetric, two-key, public-key)

3. The way in which the plaintext is processed (block cipher, stream cipher). Stream cipher may be viewed as a block cipher with block size equal to 1 element.

SYMMETRIC CIPHER MODEL

Symmetric (conventional) encryption scheme has the following ingredients

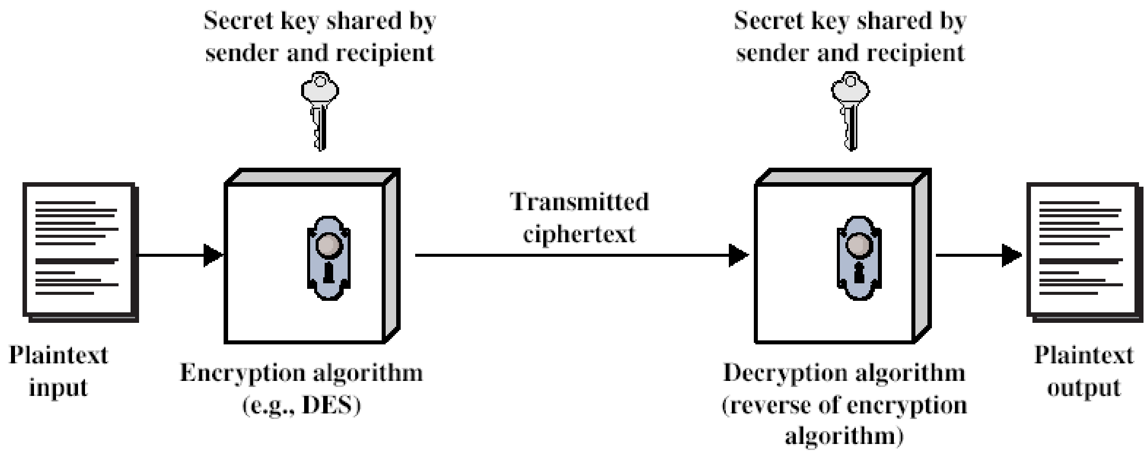

Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 2.1 Simplified Model of Symmetric Encryption

There are 2 requirements for secure use of conventional encryption:

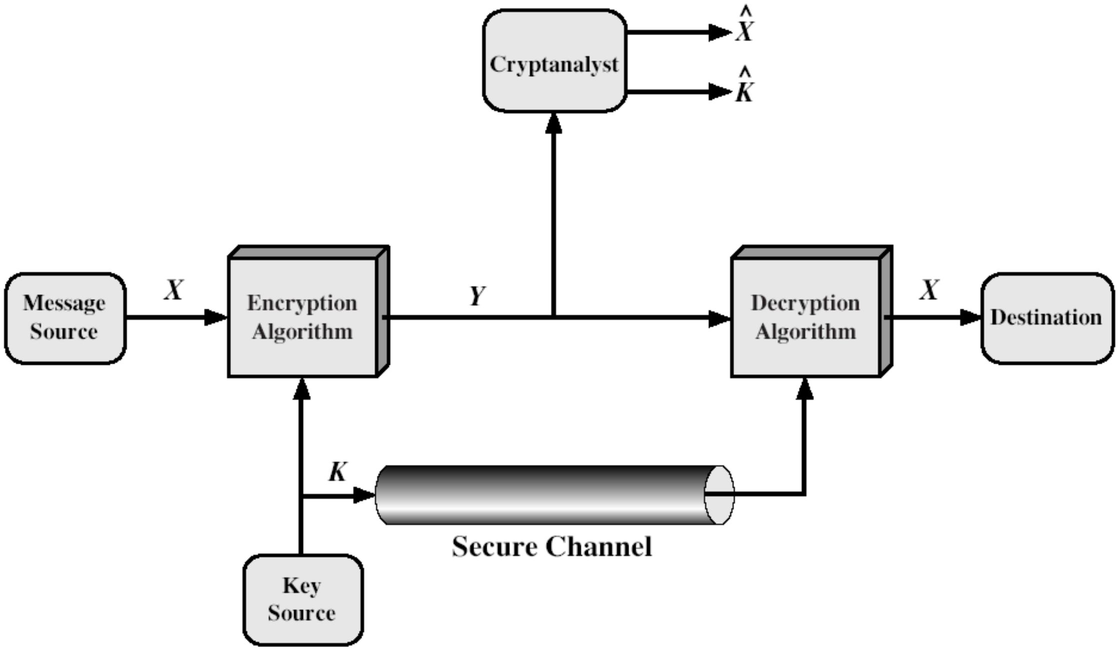
1. We need a strong encryption algorithm – the opponent should be unable to decrypt ciphertext or to discover the key even if s/he is in the

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, i.e., we do not need to keep the algorithm secret; we need to keep only the key secret.

Let's consider the essential elements of a symmetric encryption scheme:



We can write:

$Y=E_K(X)$

$X= D_K(Y)$

Opponent knows Y, E, D. He may be interested in recovering X or/and K. Knowledge of K allows him to read future messages.

CRYPTANALYSIS

There are two general approaches to attacking a conventional encryption scheme:

1. **Cryptanalysis:** attempts to use characteristics of the plaintext or even some plaintext-ciphertext pairs to deduce a specific plaintext or key being used

2. **Brute-force attack:** every possible key is tried until an intelligible translation into plaintext is obtained. On average, half of all possible keys should be tried to achieve success.

Cryptosystems can be categorized based on their security to:

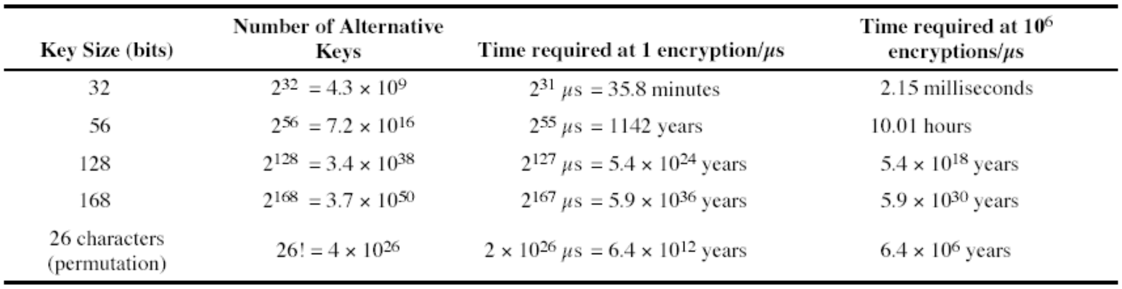
Unconditionally secure encryption scheme – ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. Excepting a scheme known as one-time pad, there is no encryption algorithm that is unconditionally secure. Therefore, encryption algorithm should meet 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

Such algorithm is called **computationally secure**. Table below shows how much time is involved for various key sizes. The 56-bit key size is used with the DES (Data Encryption Standard), 168-bit – for triple DES, 128-bit – for AES (Advanced Encryption Standard). Results are also shown for substitution codes that use 26-character key, in which all possible permutations of the 26 characters serve as keys. It is assumed that it takes 1 μs to perform a single decryption or encryption (in last column – 10^6 decryptions per 1 μs)

Table 2.2 Average Time Required for Exhaustive Key Search



Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs $= 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs $= 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs $= 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs $= 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs $= 6.4 \times 10^{12}$ years	6.4×10^6 years

All forms of cryptanalysis for symmetric encryption try to exploit the fact that traces of structure or pattern in the plaintext may survive encryption and be discernible in the ciphertext. Cryptanalysis for public-key schemes tries to use mathematical properties of pair of keys to deduce one from the other.

SUBSTITUTION TECHNIQUE

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

CAESAR CIPHER

It was used 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

Transformation is made using the following mapping:

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

Let us assign a numerical equivalent to each letter from 0 to 25. Then the algorithm may be expressed as follows. For each plaintext letter p, substitute the ciphertext letter C:

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

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

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

where k takes on a value in the range 1 to 25. The decryption algorithm is simply

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

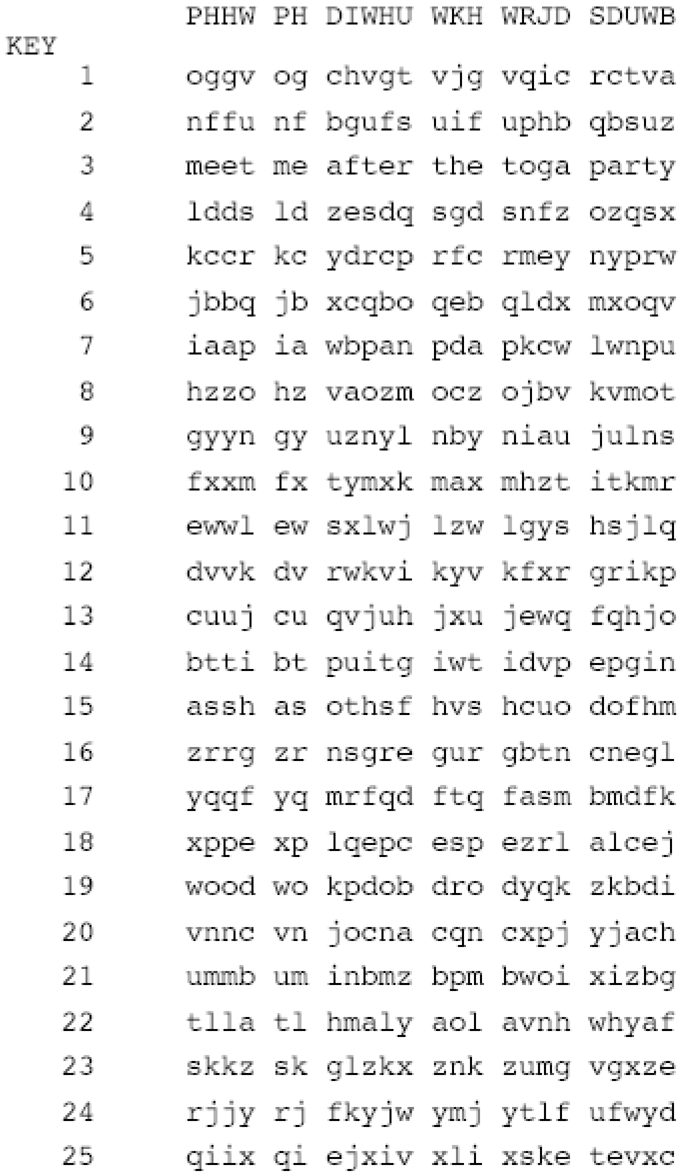
If it is known that a given ciphertext is a Caesar cipher, then a brute-force cryptanalysis is easily performed: simply try all possible 25 keys.

Three important characteristics of this problem enable us to use 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 algorithms are assumed to be known. Brute-force analysis is impractical when algorithm employs large size of keys. The 3rd characteristic is also significant. If the language of the plaintext is not known, then the plaintext output may not be recognizable.

CAESAR CIPHER (CONT)

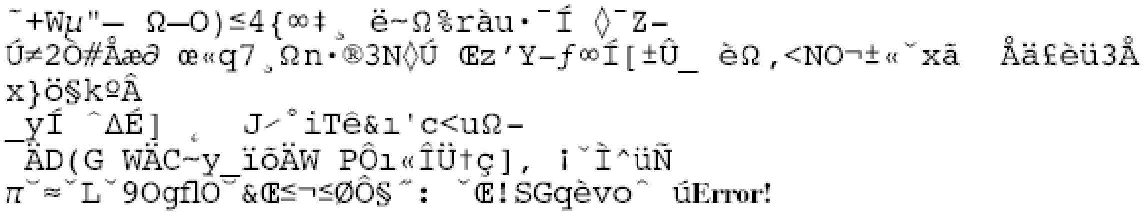


KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby 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

Figure 2.3 Brute-Force Cryptanalysis of Caesar Cipher

CAESAR CIPHER (CONT)

Furthermore, if the input is compressed in some manner, again recognition is difficult. Below is example of compression by ZIP:



```
¯+Wµ"— Ω—O)≤4{∞‡¸ ë~Ω%ràu•¯Í ◊¯Z-
Ú≠2Ò#Åæ∂ œ«q7¸Ωn•®3N◊Ú Œz'Y–ƒ∞Í[±Û_ èΩ,<NO¬±«˘xã  Åä£èü3Å
x}ö§kºÂ
_yÍ ˆΔÉ] ˛  J⁄˚iTê&ı'c<uΩ-
 ÄD(G WÄC~y_ïõÄW PÔı«ÎÜ†ç], ¡˘Ì^üÑ
π˘≈˘L˘9OgflO¯&Œ≤¬≤ØÔ§˝: ˘Œ!SGqèvoˆ úError!
```

Figure 2.4 Sample of Compressed Text

If this file is then encrypted with a simple substitution cipher (expanded to include more than just 26 characters), then the plaintext may not be recognized

MONOALPHABETIC CIPHERS

With only 25 keys Caesar cipher is far from secure. A dramatic increase in the key space may be achieved by allowing an arbitrary substitution. If instead of

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*

the cipher line can be any permutation of the 26 alphabetic symbols, then there are 26! or greater than $4*10^{26}$ possible keys. 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.

MONOALPHABETIC CIPHERS (CONT)

Let’s consider example of ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

As a first step, relative frequency of the letters can be determined and compared to a standard frequency distribution for English:

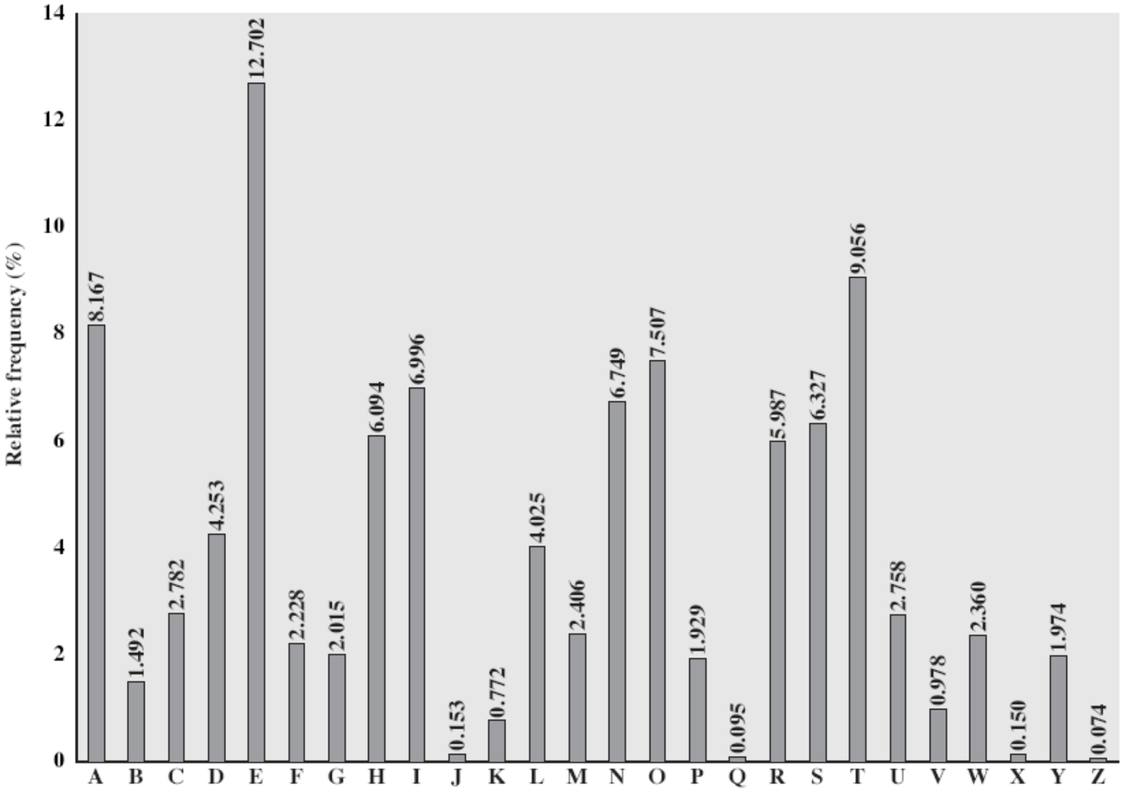


Figure 2.5 Relative Frequency of Letters in English Text

The relative frequencies of the letters in the ciphertext (in percentages):

MONOALPHABETIC CIPHERS (CONT)

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				

Comparing this with Fig.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, and H are all of the 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 (A,B,G,Y,I,J) are likely included in the set {b,j,k,q,v,x,z}. Now we could make some tentative assignments and start to fill plaintext to see if it looks like a reasonable “skeleton” of a message.

Another way, to consider frequency of two-letter combinations, is known as digrams. The most common digram is th. In our ciphertext, the most common digram is ZW, which appears 3 times. So, we make correspondence: Z – t, W – h. Then, P is equated with e. Now notice that sequence ZWP appears in the ciphertext, and we can translate it as “the”. Next, notice ZWSZ in the first line. If they form a complete word, it will be th_t. If so, S equates with a. So far, then, we have

MONOALPHABETIC CIPHERS (CONT)

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a        e  e te  a that e e a       a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t   ta t ha e ee  a e  th    t  a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e  e e tat  e   the  et
```

Continued analysis of frequencies plus trial and error may lead us to the solution:

it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in Moscow

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 the plaintext (Playfair Cipher, Hill Cipher), and the other is to use multiple cipher alphabets (Polyalphabetic Ciphers)

VIGENERE CIPHER

The best known and one of the simplest polyalphabetic ciphers is Vigenere cipher. The **Vigenère cipher** is a method of encryption invented by Giovan Batista Belaso and described in his 1553 book *La cifra del. Sig. Giovan Batista Belaso*. It was misattributed to Blaise de Vigenère in the 19th century, and given his name (http://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher).

In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts from 0 to 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 3 is denoted by the key value d.

A matrix known as Vigenere tableau is used:

VIGENERE CIPHER (CONT)

	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

Each of the 26 ciphers is laid out horizontally, with the key letter for each cipher to the left. The encryption process:

VIGENERE CIPHER (CONT)

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

Key: dec e p t i v e de c e p t i ve d e c e p t i v e
Plaintext: wea r e d i s c ov e r e d s av e y o u r s e l f
Ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Decryption is equally simple. The key letter identifies the row. The position of ciphertext letter in that row determines the column, and the plaintext is at the top of that column.

In spite of using multiple alphabets, some frequency information is preserved in Vigenere ciphertext.

Let’s sketch a method of breaking this cipher.

Suppose that the opponent believes that the ciphertext was encrypted using either monoalphabetic substitution or a Viginere cipher. A simple test can be made to make a determination. If a monoalphabetic substitution is used, then the statistical properties of the ciphertext should be the same as that of the language of the plaintext. If, on the other hand, a Viginere cipher is suspected, then progress depends on determining the length of the keyword, as it will be seen in a moment. How keyword length can be determined? If 2 identical sequences of the plaintext letters occur at a distance of integer multiple of the keyword length, they will generate identical ciphertext sequences. In our example, 2 instances of the sequence “red” are separated by 9 character positions. Consequently, in both cases, r is encrypted using key letter e, e is encrypted using key letter p, and d is encrypted using key letter t. Thus, in both cases ciphertext is VTW. Analyst may make assumption, that keyword length is either 3, either 9. Having long enough messages, cryptanalyst can determine keyword length definitely by finding common factor of all displacements of such sequences.

If keyword length is N, then the cipher consists of N monoalphabetic substitution ciphers. For example, with the keyword DECEPTIVE, the letters in positions 1, 10, 19, and so on, are all encrypted with the same monoalphabetic cipher. Thus, we can use the known frequency characteristics of the plaintext language to attack each of the monoalphabetic ciphers separately.

This scheme is vulnerable to cryptanalysis, because the key and the plaintext share the same frequency distribution of letters, so a statistical technique can be applied.

The periodic nature of the keyword can be eliminated by using a non-repeating keyword that is as long as message.

.

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 Gilbert Vernam in 1918. His system works on binary data rather than letters. The system can be expressed succinctly as follows:

$c_i = p_i \oplus k_i$,

where Ci- ith binary digit of ciphertext, Pi – of the plaintext, Ki – of the key, $\oplus$ - exclusive or (XOR) operation

Decryption is made by

$p_i = c_i \oplus k_i$

Keyword here is long enough but repeating. It can be broken with the use of known plaintext sequences.

ONE-TIME PAD

An US Army Signal Corps Captain, Joseph Mauborgne, in 1918, proposed an improvement (http://en.wikipedia.org/wiki/One-time_pad) to Vernam cipher that yields the ultimate in security. He suggested using of a **random key that was truly as long as the message**, with **no repetitions**. Such a scheme, known as one-time pad, is unbreakable. It produces random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is no way to break the code.

But in practice, one-time pad has 2 fundamental difficulties:

- there is the practical problem of making large quantities of random keys. Any heavily used system might require millions of random characters on a regular basis. Supplying truly random characters in this volume is a significant task
- 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. Thus, the key distribution problem exists.

TRANSPOSITION TECHNIQUE

Another approach to enciphering is the usage of transpositions or permutations on the plaintext letters. 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

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
MEMATRHTGPRYETEFETEOAAAT

A more complex scheme is to write the message in a rectangle, row by row, and read the message off, column by column, but to permute the order of columns. The order of 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.

The transposition cipher can be made more secure by performing more than 1 transposition

HILL CIPHER

It was 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	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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

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

C1=(k11p1+k12p2+k13p3) mod 26

C2=(k21p1+k22p2+k23p3) mod 26

C3=(k31p1+k32p2+k33p3) mod 26

This can be expressed in terms of column vectors and matrices:

C=KP mod 26,

where C and P are column vectors of length 3, representing the plaintext and ciphertext, and K is 3x3 matrix, representing the encryption key.

Operations are performed mod 26.

For example, consider the plaintext “paymoremoney”, and use the encryption key

K=

17	17	5
21	18	21
2	2	19

The first 3 letters of the plaintext are represented by the vector (15 0 24). Then K(15 0 24) = (375 819 486) mod 26 = (11 13 18) = LNS.

Continuing in this fashion, the ciphertext for the entire plaintext is LNSHDLEWMTRW.

Decryption requires using the inverse of the matrix K. The inverse K^{-1} of a matrix K is defined by K K^{-1} = K^{-1} K=I, where I is the unit matrix (1-s on the diagonal, other elements – zeroes). The inverse of the matrix does not always exist, but when it does, it satisfies the preceding equation. In this case, the inverse is

K^{-1}=

4	9	15
15	17	6
24	0	17

This is demonstrated as follows:

K K^{-1} =

443	442	442
858	495	780
494	52	365

And after taking mod 26 of the elements above, unit matrix is obtained.

In general terms, the Hill system can be expressed as follows:

$C=E_K(P)=KP \bmod 26$

$P= D_K(C)=K^{-1}C \bmod 26 = K^{-1}KP = P$

As with Playfair, the strength of the Hill cipher is that it completely hides single-letter frequencies.

ATTACKING HILL CIPHER

Although the Hill cipher is strong against a ciphertext-only attack (opponent has only ciphertext), it is easily broken with a known plaintext attack (opponent has pairs plaintext – ciphertext). For an m*m Hill cipher, suppose we have m plaintext-ciphertext pairs, each of length m. We label the pairs Pj=(p_{1j}, p_{2j},…, p_{mj}) and Cj=(c_{1j}, c_{2j},…, c_{mj}) such that Cj=KPj for 1<=j<=m and for some unknown key matrix K. Now define two m*m matrices X=(p_{ij}) and Y=(c_{ij}). Then we can form matrix equation Y=KX. If X has an inverse, then we can determine $K=YX^{-1}$. If X is not invertible, then a new version of X can be formed until an invertible X is obtained.

Suppose that the plaintext "friday" is encrypted using a 2*2 Hill cipher to yield the ciphertext PQCFKU. Thus, we know that

K(5 17) = (15 16);

K(8 3) = (2 5);

K(0 24) = (10 20).

Using the first 2 plaintext-ciphertext pairs, we have

$$\begin{pmatrix} 15 & 2 \\ 16 & 5 \end{pmatrix} = K \begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix} \bmod 26$$

The inverse of X can be computed:

$$\begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$$

$$K = \begin{pmatrix} 15 & 2 \\ 16 & 5 \end{pmatrix} \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix} = \begin{pmatrix} 137 & 60 \\ 149 & 107 \end{pmatrix} \bmod 26 = \begin{pmatrix} 7 & 8 \\ 19 & 3 \end{pmatrix}$$

Let's check now that this key matrix produces required transformation:

$$\begin{pmatrix} 7 & 8 \\ 19 & 3 \end{pmatrix}\begin{pmatrix} 5 \\ 17 \end{pmatrix} = \begin{pmatrix} 35+136 \\ 95+51 \end{pmatrix} = \begin{pmatrix} 171 \\ 146 \end{pmatrix} \bmod 26 = \begin{pmatrix} 15 \\ 16 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 8 \\ 19 & 3 \end{pmatrix}\begin{pmatrix} 8 \\ 3 \end{pmatrix} = \begin{pmatrix} 56+24 \\ 152+9 \end{pmatrix} = \begin{pmatrix} 80 \\ 161 \end{pmatrix} \bmod 26 = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 8 \\ 19 & 3 \end{pmatrix}\begin{pmatrix} 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 192 \\ 72 \end{pmatrix} \bmod 26 = \begin{pmatrix} 10 \\ 20 \end{pmatrix}$$

Cryptography and Network Security 1214-0527

Assist. Prof. Dr. Anas Melhem
Computer System Engineering Department
Palestine Technical University

Today:

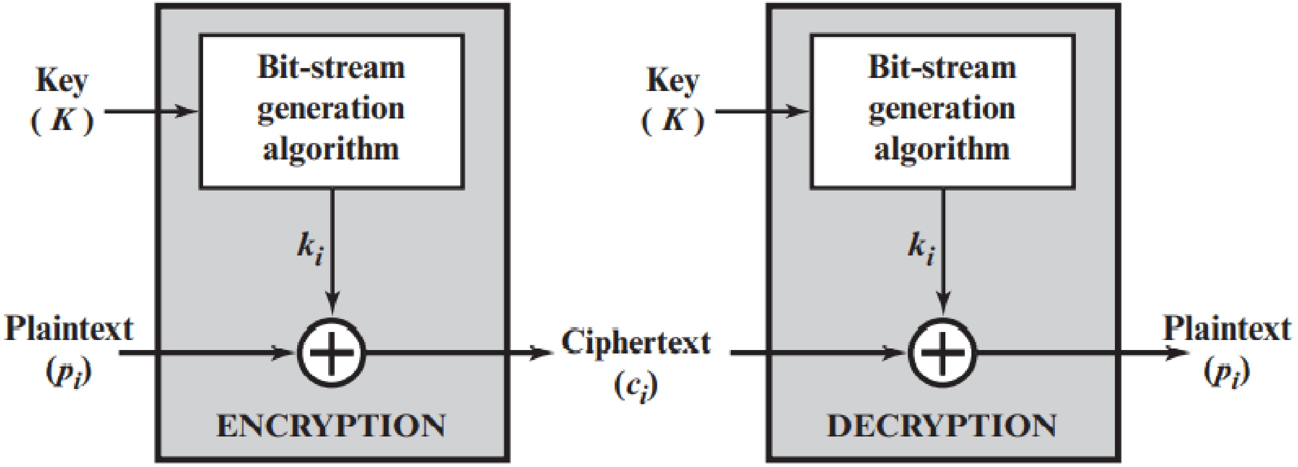
- Feistel Cipher

Readings:

- W. Stallings, Cryptography and Network Security, Chapter 3

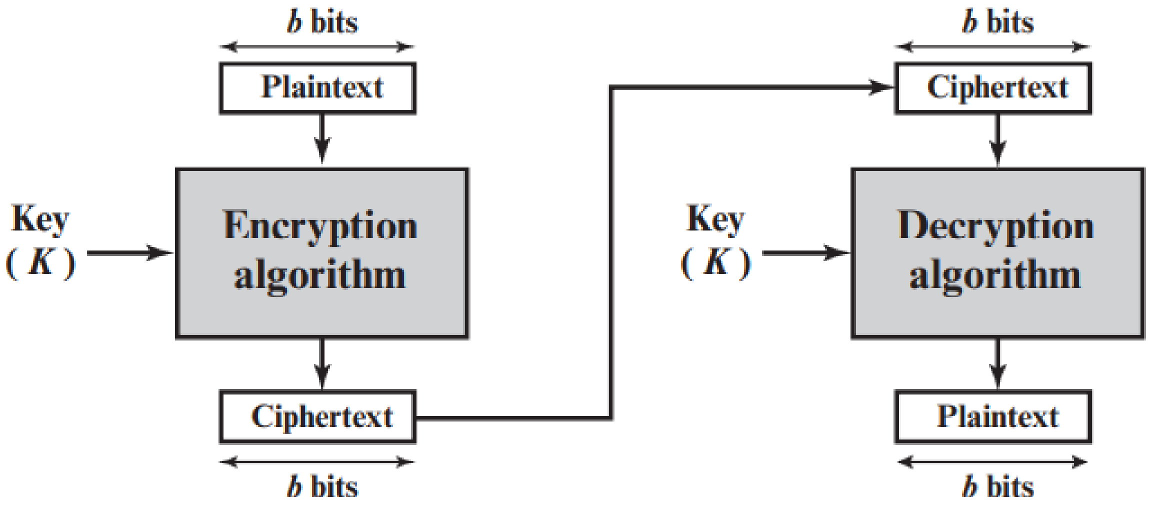
Stream Ciphers VS Block Ciphers

- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
  - Example: Vigenère cipher



Stream Ciphers VS Block Ciphers

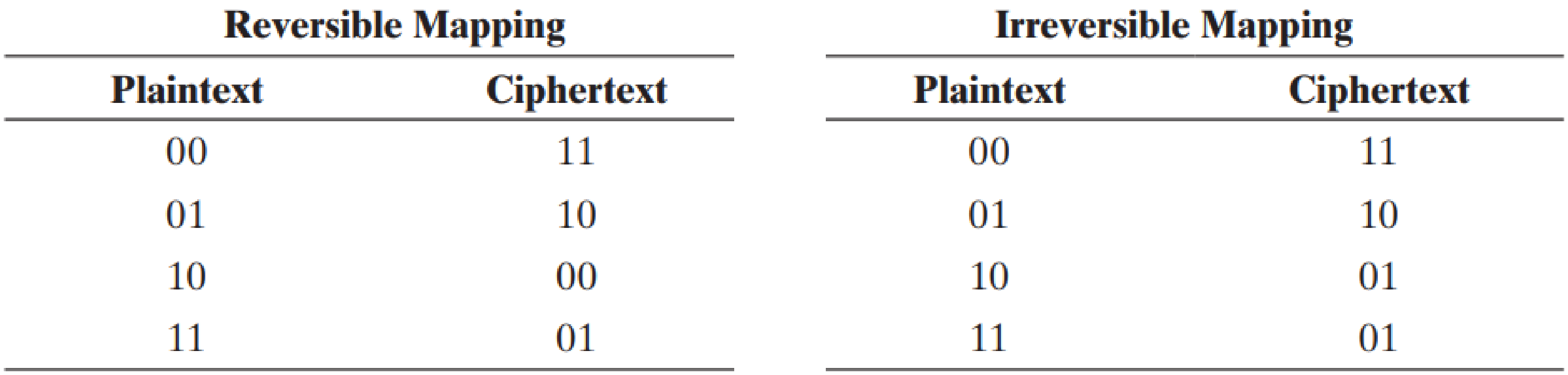
- A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used.



Both of stream ciphers and block ciphers are symmetric key ciphers.
block ciphers are applicable to a broader range of applications than stream ciphers. The vast majority of network-based symmetric cryptographic applications make use of block ciphers.

Motivation for the Feistel Cipher Structure

- A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits.
- There are 2^n! possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible).
- In the reversible transformation, each plaintext must produce a unique ciphertext block.



Reversible Mapping

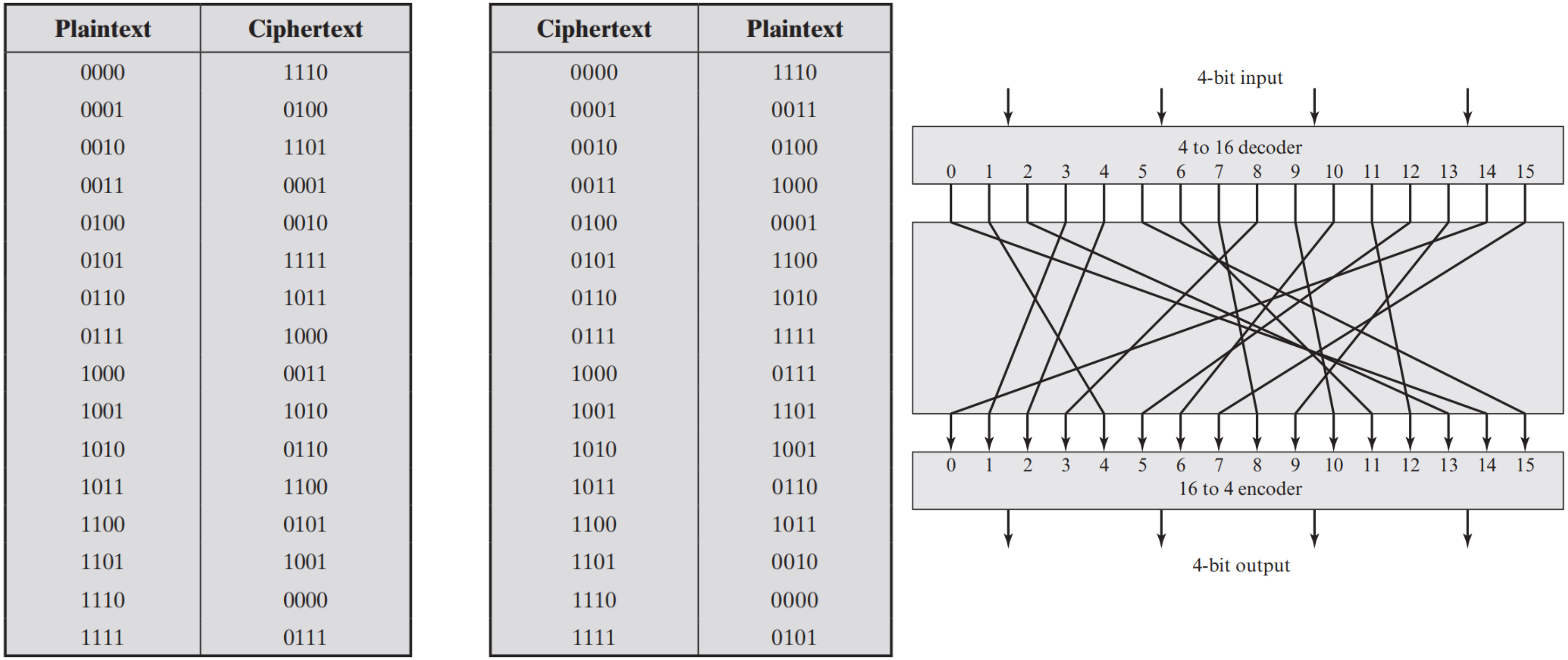
Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Motivation for the Feistel Cipher Structure

- The most general form of block cipher is $2^n!$,and can be used to define any reversible mapping between plaintext and ciphertext.



Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

General n-bit-n-bit Block Substitution (shown with $n = 4$)

Motivation for the Feistel Cipher Structure

- Feistel refers to this as the *ideal block cipher*, because it allows for the maximum number of possible encryption mappings from the plaintext block.
- An arbitrary reversible substitution cipher (the ideal block cipher) for a large block size is not practical, however, from an implementation and performance point of view.
  - for $n = 4$, the required key length is $(4\ bits) \times (16\ rows)\ = 64$ bits.
  - for $n = 64$, the required key length is $(64\ bits) \times (2^{64}\ rows)$ $= 2^{70} \approx 10^{21}$ bits.
- Feistel points out that what is needed is an approximation to the ideal block cipher system for large n, built up out of components that are easily realizable

Motivation for the Feistel Cipher Structure

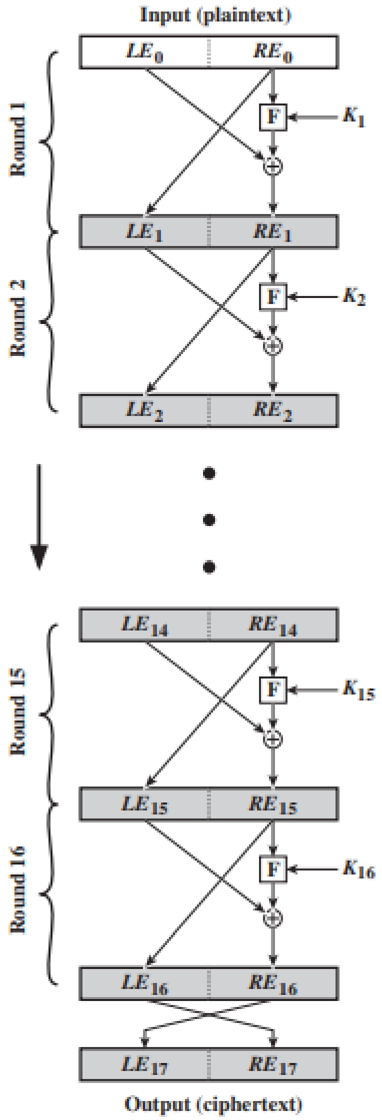
- Feistel refers to this as the *ideal block cipher*, because it allows for the maximum number of possible encryption mappings from the plaintext block.
- An arbitrary reversible substitution cipher (the ideal block cipher) for a large block size is not practical, however, from an implementation and performance point of view.
  - for $n = 4$, the required key length is $(4\ bits) \times (16\ rows)\ = 64$ bits.
  - for $n = 64$, the required key length is $(64\ bits) \times (2^{64}\ rows) = 2^{70} \approx 10^{21}$ bits.
- Feistel points out that what is needed is an approximation to the ideal block cipher system for large n, built up out of components that are easily realizable

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 2^k possible transformations, rather than the 2^n! transformations available with the ideal block cipher.

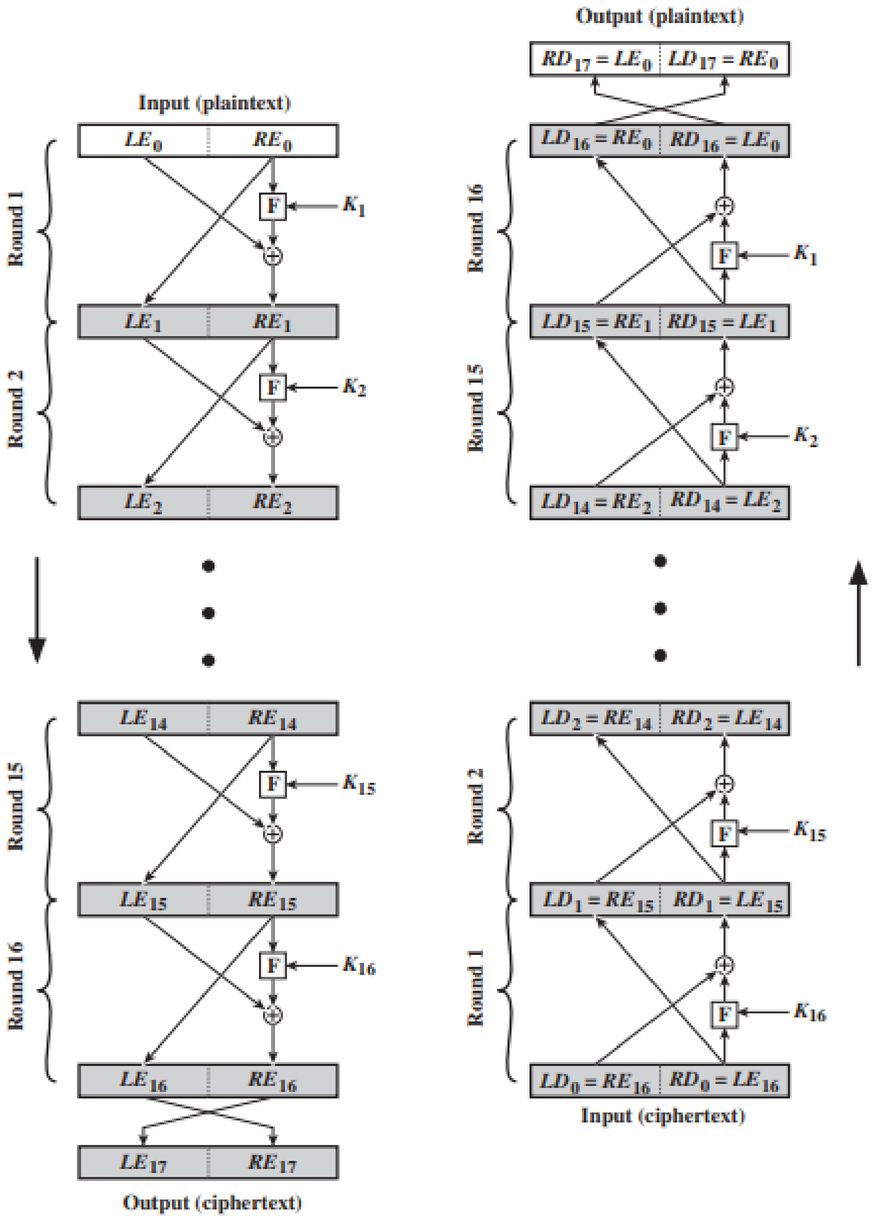
Feistel Cipher Structure

- The inputs to the encryption algorithm are a plaintext block of length $2w$ bits and a key K.
- The plaintext block is divided into two halves, LE_0 and RE_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 $LE_i - 1$ and $RE_i - 1$ derived from the previous round, as well as a subkey K_i derived from the overall K.
- The subkeys K_i are different from K and from each other.



Feistel Encryption

Feistel Cipher Structure



The Feistel Cipher

- Feistel's is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates ***confusion*** and ***diffusion*** functions.
- 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.
- confusion seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, 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.

Cryptography and Network Security 1214-0527

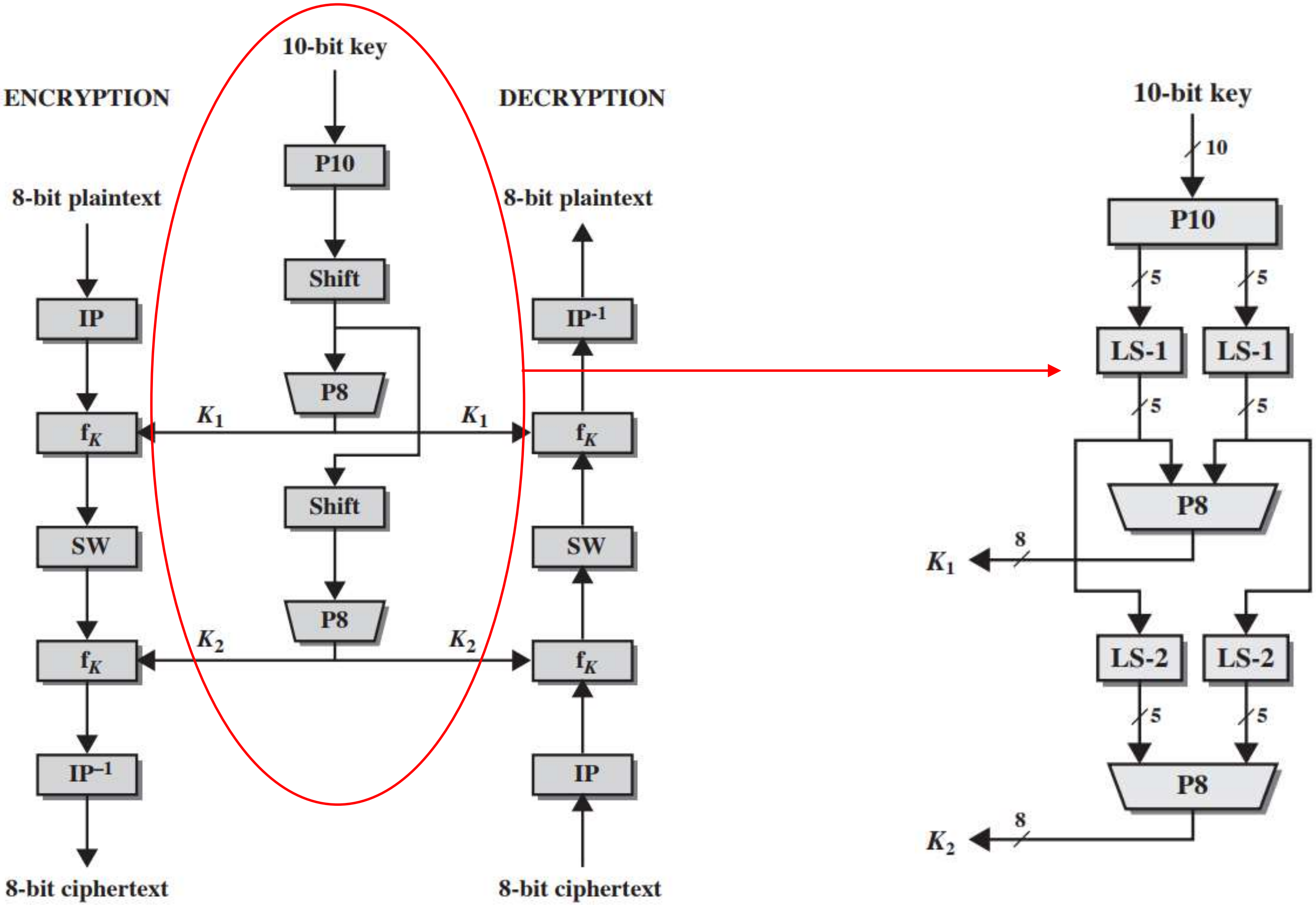
S-DES

Anas Melhem
Computer System Engineering Department
Palestine Technical University

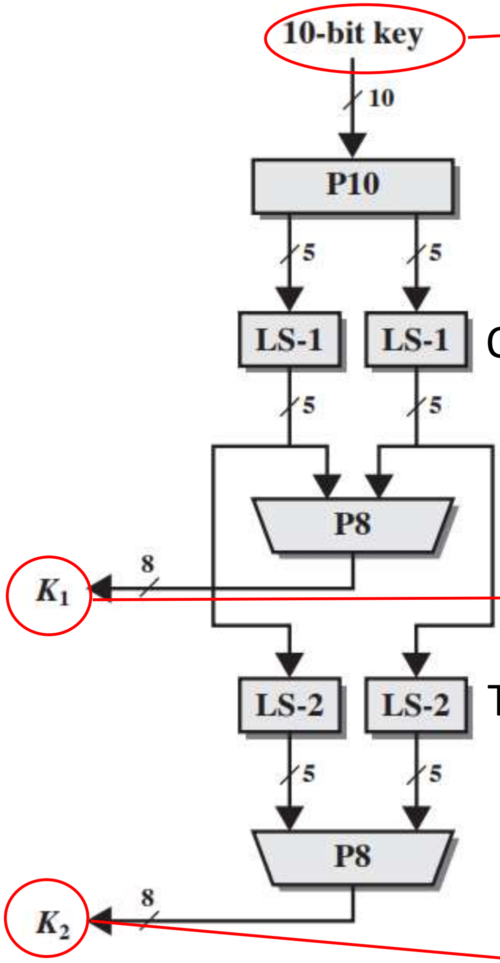
Simplified DES

- Developed by Professor Edward Schaefer of Santa Clara University, as an educational rather than a secure encryption algorithm.
- It has similar properties and structure to DES with much smaller parameters.
- Takes an 8-bit block of plaintext (example: 10111101) and a 10-bit key as input and produces an 8-bit block of ciphertext as output.

S-DES Scheme


ENCRYPTION
8-bit plaintext
IP
f_K
SW
f_K
IP^{-1}
8-bit ciphertext
10-bit key
P10
Shift
P8
K_1
K_1
Shift
P8
K_2
K_2
DECRYPTION
8-bit plaintext
IP^{-1}
f_K
SW
f_K
IP
8-bit ciphertext
10-bit key
10
P10
5
5
LS-1
LS-1
5
5
P8
8
K_1
LS-2
LS-2
5
5
P8
8
K_2

S-DES Key Generation



One bit rotate lift

Two bits rotate lift

1	2	3	4	5	6	7	8	9	10
1	0	1	0	0	0	0	0	1	0

P10									
3	5	2	7	4	10	1	9	8	6

1 0 0 0 0 | 0 1 1 0 0

0 0 0 0 1 | 1 1 0 0 0

P8							
6	3	7	4	8	5	10	9

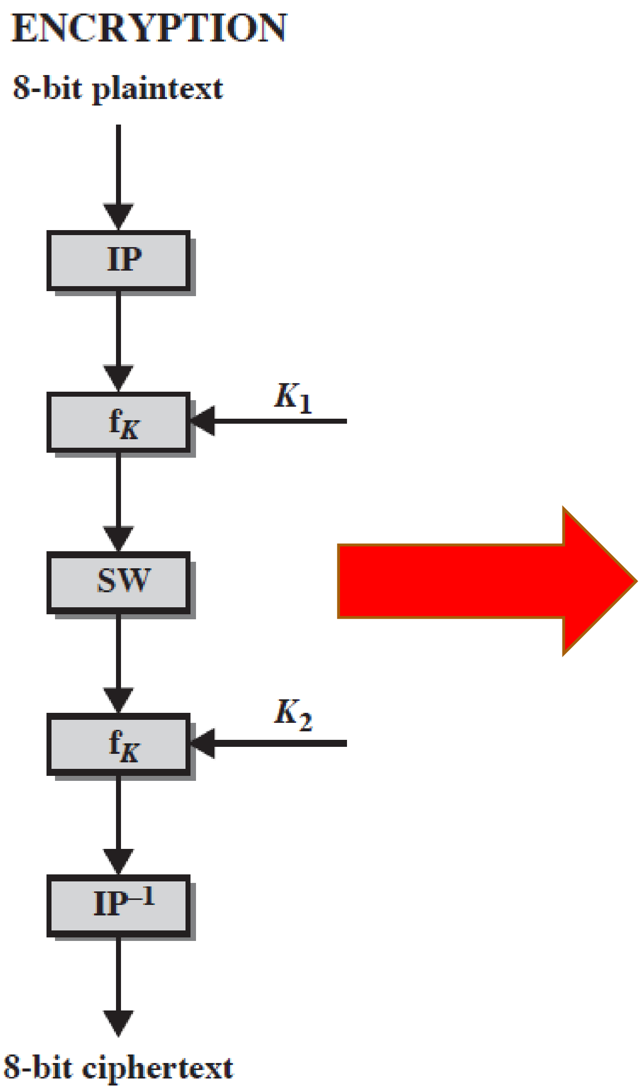
1 0 1 0 0 1 0 0

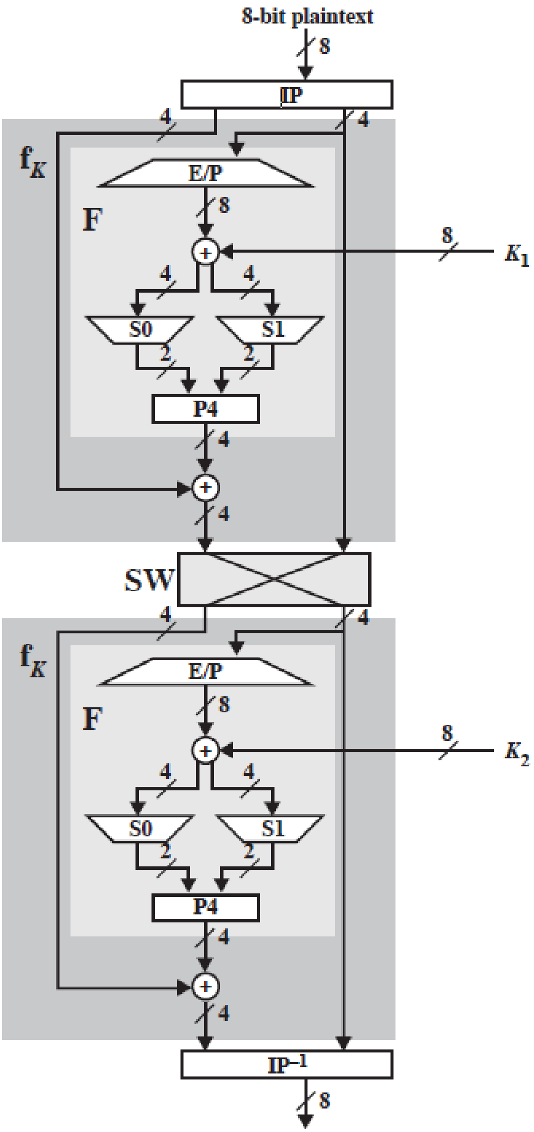
0 0 1 0 0 | 0 0 0 1 1

P8							
6	3	7	4	8	5	10	9

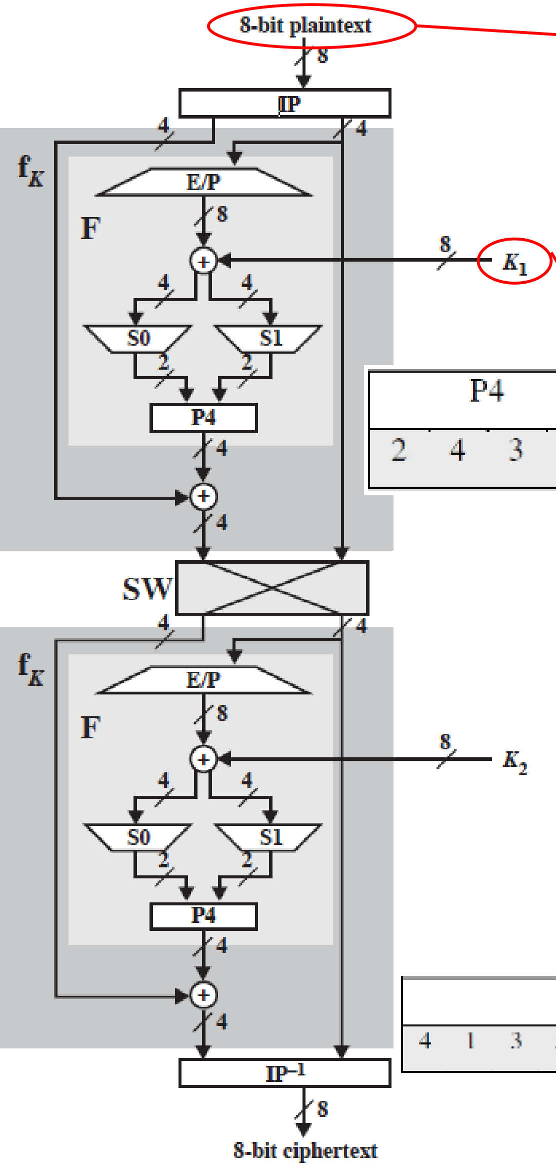
0 1 0 0 0 0 1 1

S-DES Encryption





S-DES Encryption



Z = 0 1 1 1 1 0
1 0

IP							
2	6	3	1	4	8	5	7

1 0 1 0 1 0 1
1

E/P							
4	1	2	3	2	3	4	1

P4			
2	4	3	1

1 1 0 1 0 1 1
1 0 1 0 0 1 0 0 ⊕
1

0 1 1 1 0 0 1
1

$$S0 = \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{pmatrix} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array} \qquad S1 = \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{pmatrix} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array}$$

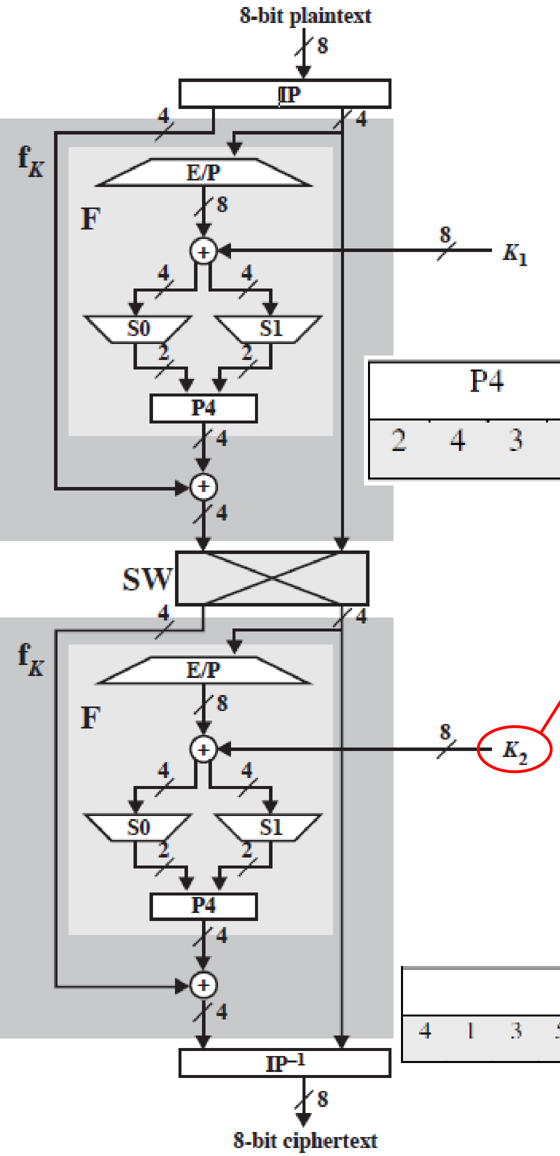
0 0 0 0 ⊕
1 0 1 0

IP−1							
4	1	3	5	7	2	8	6

1 0 1 1 0 1
0 1
1 0 1 1 0 1
1 0

The 1st and 4th input bits are treated as a 2-bit number that specify a row of the S-box, and the 2nd and 3rd input bits specify a column of the S-box.

S-DES Encryption



P4			
2	4	3	1

1 0 1 1 1 0 1
0

E/P							
4	1	2	3	2	3	4	1

0 1 0 1 0 1 0
1

⊕

0 1 0 0 0 0 1 1

0 0 0 1 0 1 1
0

$$S0 = \begin{array}{c|cccc|c} & 0 & 1 & 2 & 3 & \\ \hline & 1 & 0 & 3 & 2 & 0 \\ & 3 & 2 & 1 & 0 & 1 \\ & 0 & 2 & 1 & 3 & 2 \\ & 3 & 1 & 3 & 2 & 3 \end{array} \qquad S1 = \begin{array}{c|cccc|c} & 0 & 1 & 2 & 3 & \\ \hline & 0 & 1 & 2 & 3 & 0 \\ & 2 & 0 & 1 & 3 & 1 \\ & 3 & 0 & 1 & 0 & 2 \\ & 2 & 1 & 0 & 3 & 3 \end{array}$$

1 1 1 1

⊕

1 0 1 1

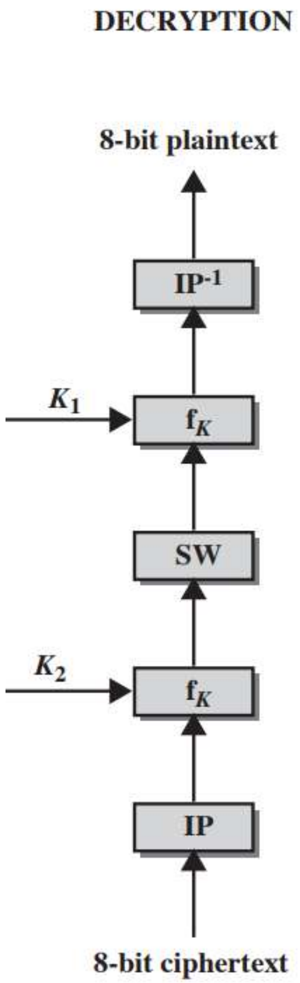
0 1 0 1 0 1
0 0

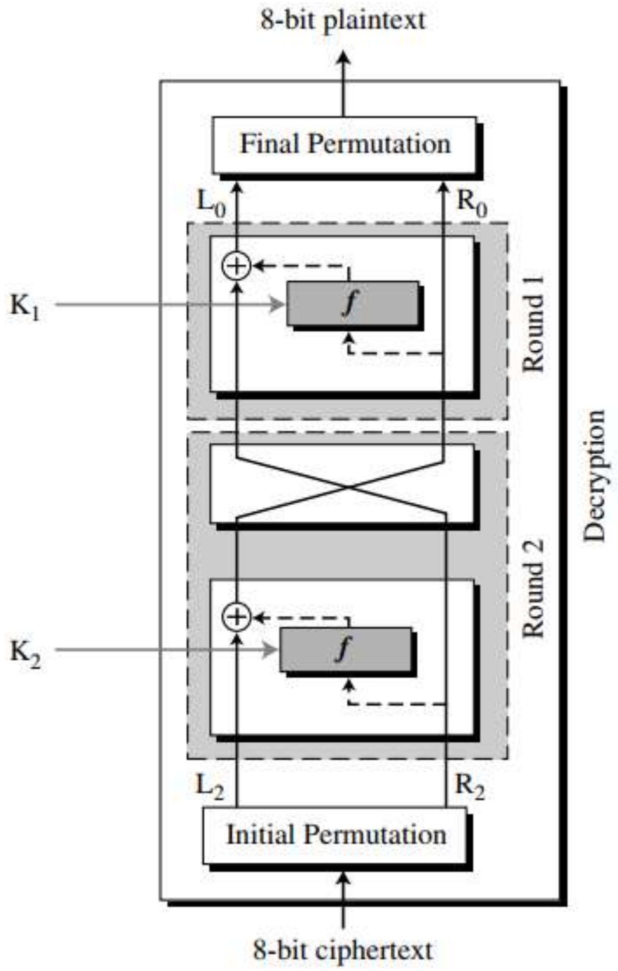
0 0 0 1 1 0 0
1

IP−1							
4	1	3	5	7	2	8	6

The 1st and 4th input bits are treated as a 2-bit number that specify a row of the S-box, and the 2nd and 3rd input bits specify a column of the S-box.

S-DES Decryption





S-DES VS DES

	DES	S-DES
Block Size (bit)	64	8
Key Size (bit)	56	10
# Feistel Rounds	16	2
# S-Boxes/ Round	8 Boxes of 6- bits each	2 Boxes of 4-bit each
# of Sub keys/ Round	16 sub key of 48 bit each	2 sub key of 8 bit each

- EFF (Electronic Frontier Foundation) in 1998 designed the DES Cracker with cost less than $250,000 which broke a DES key in 3 days. Using a network of computers this was reduced to 22 hours 15 minutes in 1999.
- **Triple DES: 3DES(x)= E(K3,(D(K2 (E(K1,x)))))**
- New competition announced AES selected in 2002.

DATA ENCRYPTION STANDARD

It was adopted in 1977 by the National Bureau of Standards (NBS), now National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). In 1971, IBM's team under Horst Feistel leadership developed algorithm LUCIFER, operating on 64-bit blocks with 128-bit key. Further, IBM's team headed by Walter Tuchman and Carl Meyer revised LUCIFER to make it more resistant to cryptanalysis, but they reduced key size to 56 bits. In 1973, NBS issued a request for proposals for a national cipher standard. IBM submitted results of its Tuchman-Meyer project. This was by far the best algorithm proposed and was adopted in 1977 as Data Encryption Standard. In 1994, NIST reaffirmed DES for federal use for another 5 years. In 1999, NIST issued a new version of its standard (FIPS PUB 46-3) that indicated that DES should only be used for legacy systems and that triple DES be used.

DES ENCRYPTION

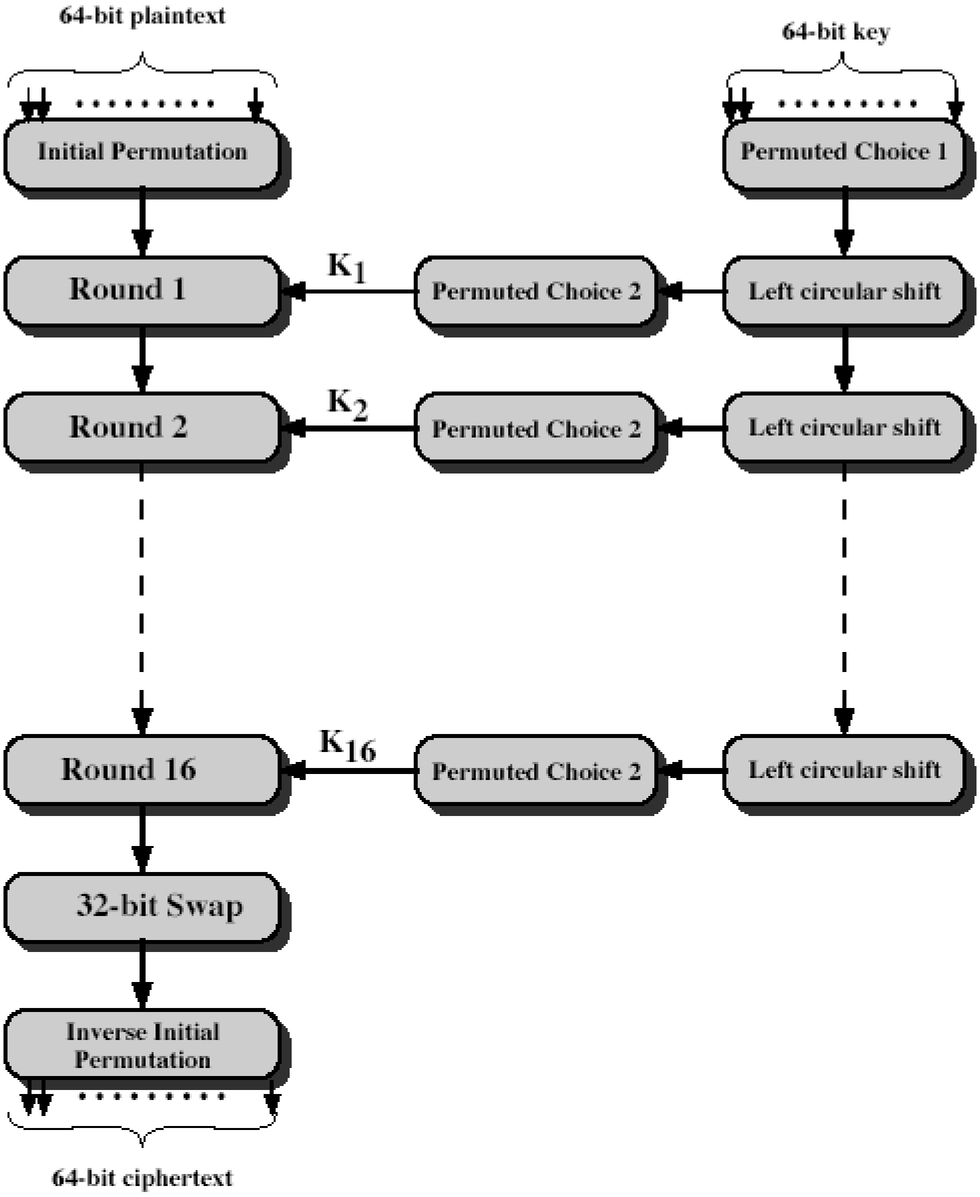


Figure 3.7 General Depiction of DES Encryption Algorithm

32-bit swap swaps left and 32-bit halves obtained after Round 16, we get pre-output. Finally, pre-output passes through a permutation IP^{-1}, that is an inverse to initial permutation IP, to produce the 64-bit cipher-text. The right-hand portion of Fig. 3.7 shows the way in which 56-bit is used. For each of 16 rounds a sub-key Ki is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round.

INITIAL PERMUTATION AND ITS INVERSE

It affects on 64-bit input

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

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

DETAILS OF SINGLE ROUND

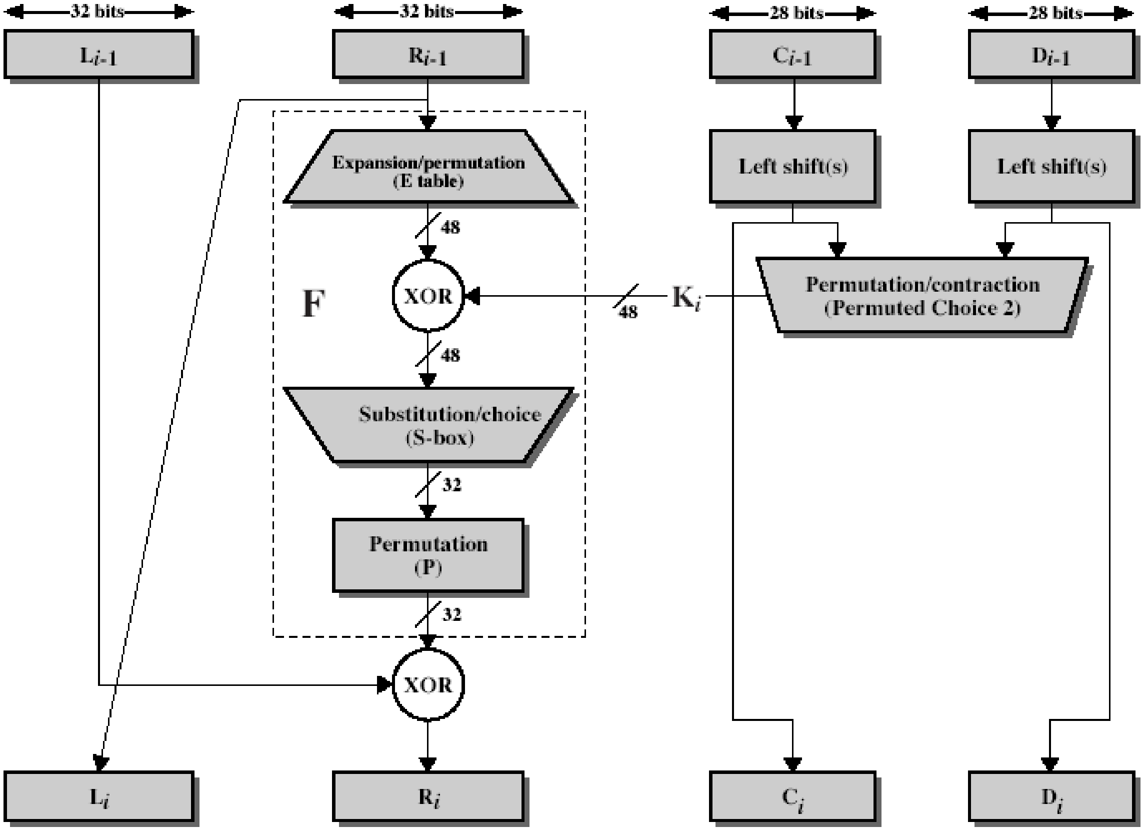


Figure 3.8 Single Round of DES Algorithm

The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L and R. As in the classic Feistel cipher, the overall process at each round is summarized as follows:

$L_i = R_{i-1}$

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

The round key Ki is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by Expansion/Permutation (E table):

Expansion/Permutation (E table)		
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

DETAILS OF SINGLE ROUND (CONT 1)

The resulting 48 bits are XORed with Ki. This 48 bit result passes through a substitution function that produces 32-bit output, which is permuted by Permutation function (P):

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

The role of S-boxes is illustrated in Fig. 3.9:

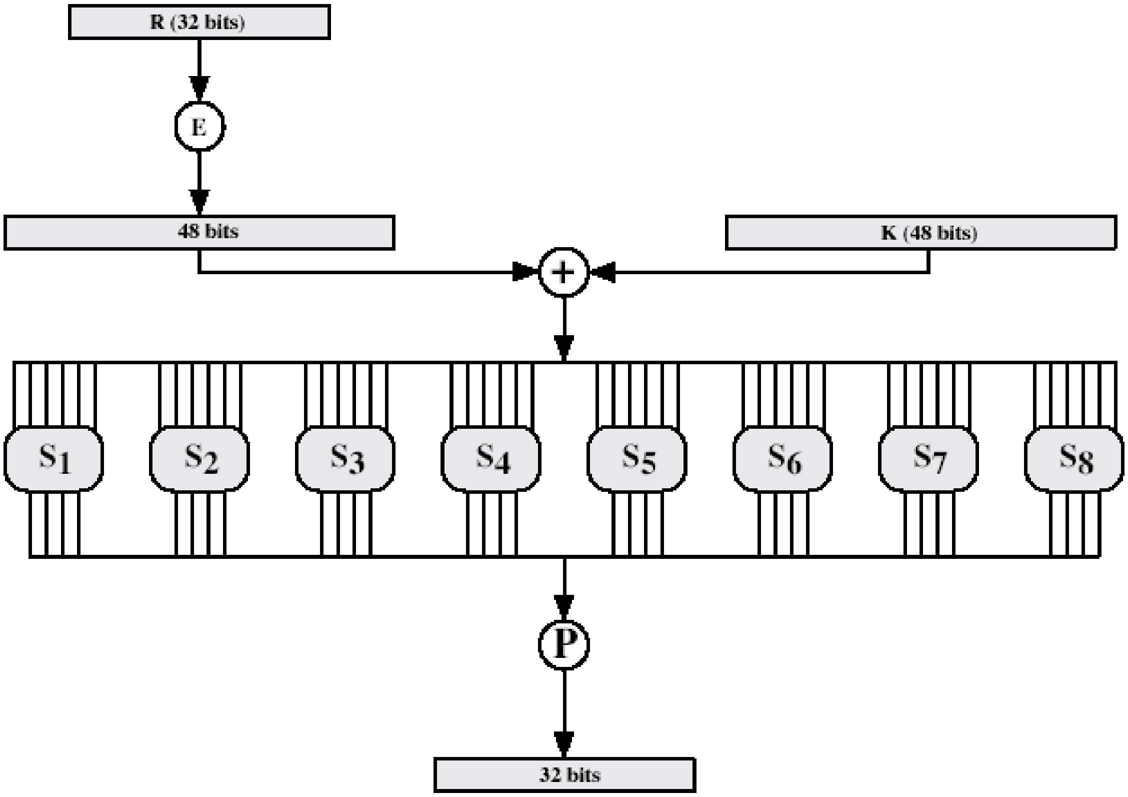


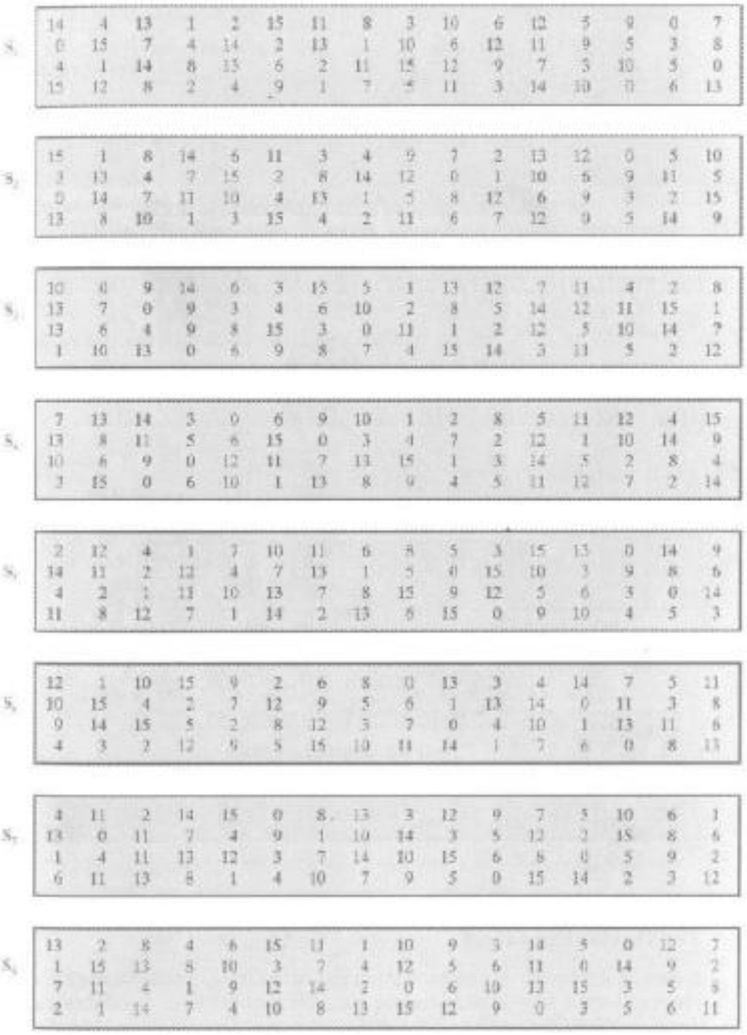
Figure 3.9 Calculation of F(R, K)

The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits input and produces 4 bits as output.

DETAILS OF SINGLE ROUND (CONT 2)

These transformations are:

Table 3.3 Definition of DES S-Boxes



S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Each row of an S-box defines a general reversible substitution: middle 4 bits of each group of 6-bit input are substituted by S-box output, 1st and last 6th bits define what particular substitution out of four to use.

KEY GENERATION

Input key has 64 bits. But each 8th bit is not used: bits 8,16,24,32,40,48,56,64 are not further used. The 56-bit key is first subjected to permutation Permuted Choice 1:

Permuted Choice 1 (PC-1)						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

The resulting 56-bit key is then treated as two 28-bit quantities, labeled C0 and D0. 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 as governed by the following:

Schedule of Left Shifts																
Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

These shifted values serve as input to the next round. They also serve as input to Permuted Choice 2, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

Permuted Choice 2 (PC-2)							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

DES DECRYPTION

As with any Feistel cipher, decryption uses the same algorithm as encryption, except that the application of sub-keys is reversed.

THE AVALANCHE EFFECT IN DES

1 bit change in the plaintext leads to 34 bit difference in the cipher-text. 1 bit change in the key leads to 35 bit difference in the cipher-text.

THE STRENGTH OF DES

DES 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 3 days.

Design criteria for S-boxes were not made public, so there was a concern that cryptanalysis is possible for an opponent who knows the weaknesses in S-boxes. Up to now, there are no published results about such weaknesses in S-boxes.

DES also appears to be resistant to timing attack but suggest some avenues to explore. Timing attack tries to understand essence of algorithm by analysis of time of its work on different inputs. One of such approaches yields a Hamming weight (number of bits equal to 1) of the secret key.

DIFFERENTIAL AND LINEAR CRYPTANALYSIS

Differential cryptanalysis attack is first published attack that is capable of breaking DES in less than 2^{55} complexity. The scheme can successfully crypt-analyze DES with an effort of the order of 2^{47}, requiring 2^{47} chosen plaintexts. Idea is to follow differences in 2 plaintexts in the rounds of DES transformations, and to estimate probability of the output difference depending on the used key. The first open publication on the differential cryptanalysis was in 1990.

Linear cryptanalysis was described in 1993. Idea is to find linear equation (with XOR operations) between bits of plaintext, cipher-text and key that holds with probability greater than 0.5.

BLOCK CIPHER DESIGN PRINCIPLES

1. No output of any S-box should be too close a linear function of the input bits.
2. Each row of an S-box should include all 16 possible output combinations
3. If two inputs to an S-box differ in exactly 1 bit, the outputs must differ in at least 2 bits.
4. If 2 inputs to an S-box differ in the 2 middle bits exactly, the outputs must differ in at least 2 bits.
5. If 2 inputs to an S-box differ in their first 2 bits and are identical in their last 2 bits, the 2 outputs must not be the same
6. For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference

These criteria are intended to increase confusion properties.

BLOCK CIPHER DESIGN PRINCIPLES (CONT 1)

The criteria for the permutation P are as follows:

1. The 4 output bits from each S-box at round I are distributed so that 2 of them affect (provide input for) “middle bits” of round (i+1) and the other 2 affect end bits. The 2 middle bits of input to an S-box are not shared by adjacent S-boxes. The end bits are the 2 left-hand bits and the 2 right-hand bits which are shared with adjacent S-boxes (for example, bits 1,2,3,4 – outputs of S1, affect on bits 9, 17 (end-bits) and 23,31 (middle-bits) respectively
2. The 4 output bits from each S-box affect 6 different S-boxes on the next round, and no 2 affect the same S-box (for example, bits 1,2,3,4 - outputs of S1, affect (S2, S3) – bit 1, (S4,S5) – bit 2, S6 – bit 3, S8 – bit 4
3. For 2 S-boxes j, k, if an output bit from Sj affects a middle bit of Sk on the next round, then an output bit from Sk cannot affect a middle bit of Sj. This implies that for j=k, an output bit from Sj must not affect a middle bit of Sj. For example, output bit 3 of S1 affects middle bit of S6. Then we are to have that output bits of S6 are not to affect middle bits of S1. Output bits of S6 are 21,22,23,24. Bit 21 affects bit 4 – end bit of S1, S2; bit 22 affects bit 29 – end bit of S7, S8; bit 23 affects bit 11 – middle bit of S3; bit 24 affects bit 19 – middle bit of S5. So, middle bits of S1 are not affected by output bits of S6.

These criteria are intended to increase diffusion properties.

Number of rounds – the greater this number, the more difficult is cryptanalysis. If DES had 15 or less rounds, differential cryptanalysis would require less effort than brute-force attack

Function F should be nonlinear, provide avalanche effect. Also, bit independence criterion is used: output bits j and k should change independently when any single input bit i is inverted, for all i,j,k.

Size of S-boxes: larger size – more resistant to differential and linear cryptanalysis. S-boxes may be made randomly or according to mathematical rules automatically. Contents of S-boxes may depend on the key.

Key schedule algorithm – no general principles for this have yet been promulgated. Key schedule (production of sub-keys) should guarantee key/cipher-text avalanche criterion and bit independence criterion.

BLOCK CIPHER MODES OF OPERATION

Four DES modes of operations have been defined (FIPS 81, http://www.itl.nist.gov/fipspubs/fip81.htm):

Mode	Description	Typical application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key	Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of the plaintext and the preceding 64 bits of the cipher-text	General-purpose block-oriented transmission Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding cipher-text is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of cipher-text	General-purpose stream-oriented transmission Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output	Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block	General-purpose block-oriented transmission Useful for high-speed requirements

ELECTRONIC CODEBOOK MODE

It may be considered that each 64-bit plaintext is mapped to respective 64-bit cipher-text, and each such possible pair represents 1 page of the codebook

For lengthy messages ECB mode may be not secure. If the message has repetitive elements with a period of repetition a multiple of 64 bits, then these elements can be identified by the analyst.

CIPHER BLOCK CHAINING MODE

We need that same plaintext block, if repeated, produces different cipher-text blocks. The simple way is CBC mode:

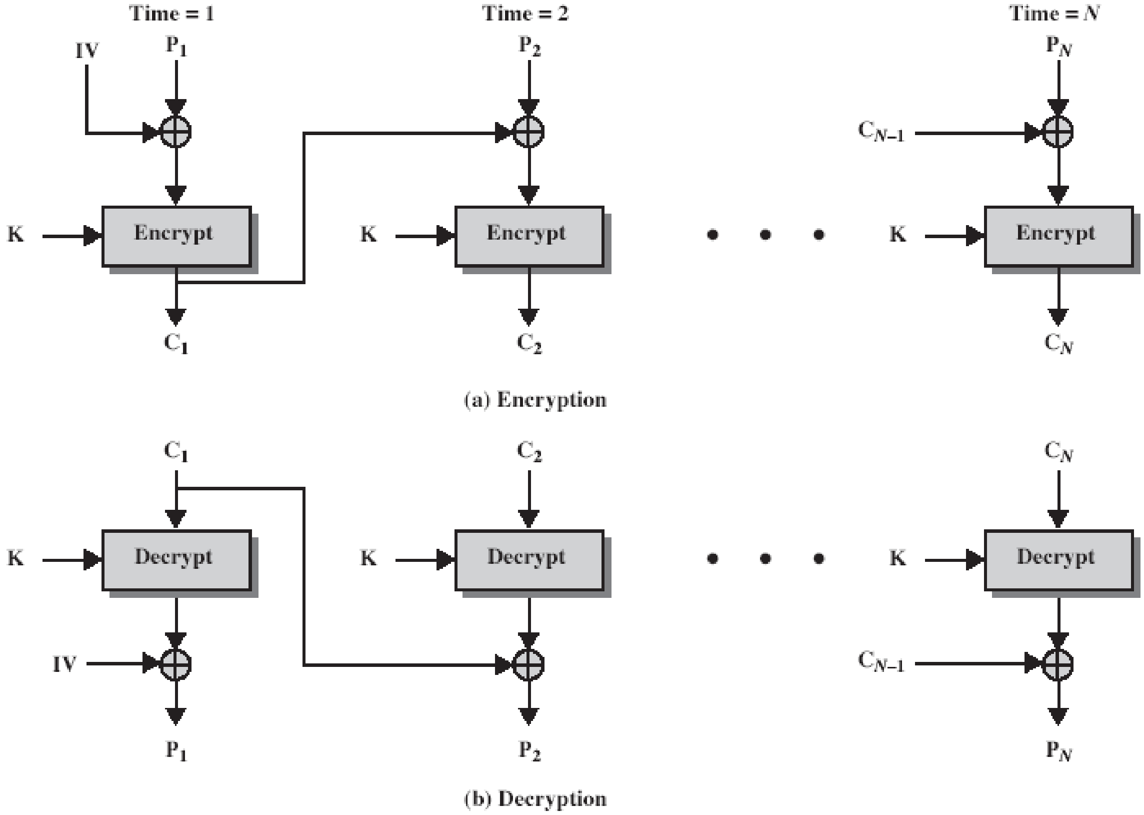


Figure 3.12 Cipher Block Chaining (CBC) Mode

Here IV- Initialization Vector – must be known to both sender and receiver.

CIPHER FEEDBACK MODE

DES is a block cipher, but it may be used as a stream cipher if to use the Cipher Feedback Mode (CFB) or the Output Feedback Mode (OFB). A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can be encrypted and transmitted immediately using a character-oriented stream cipher.

CFB scheme follows:

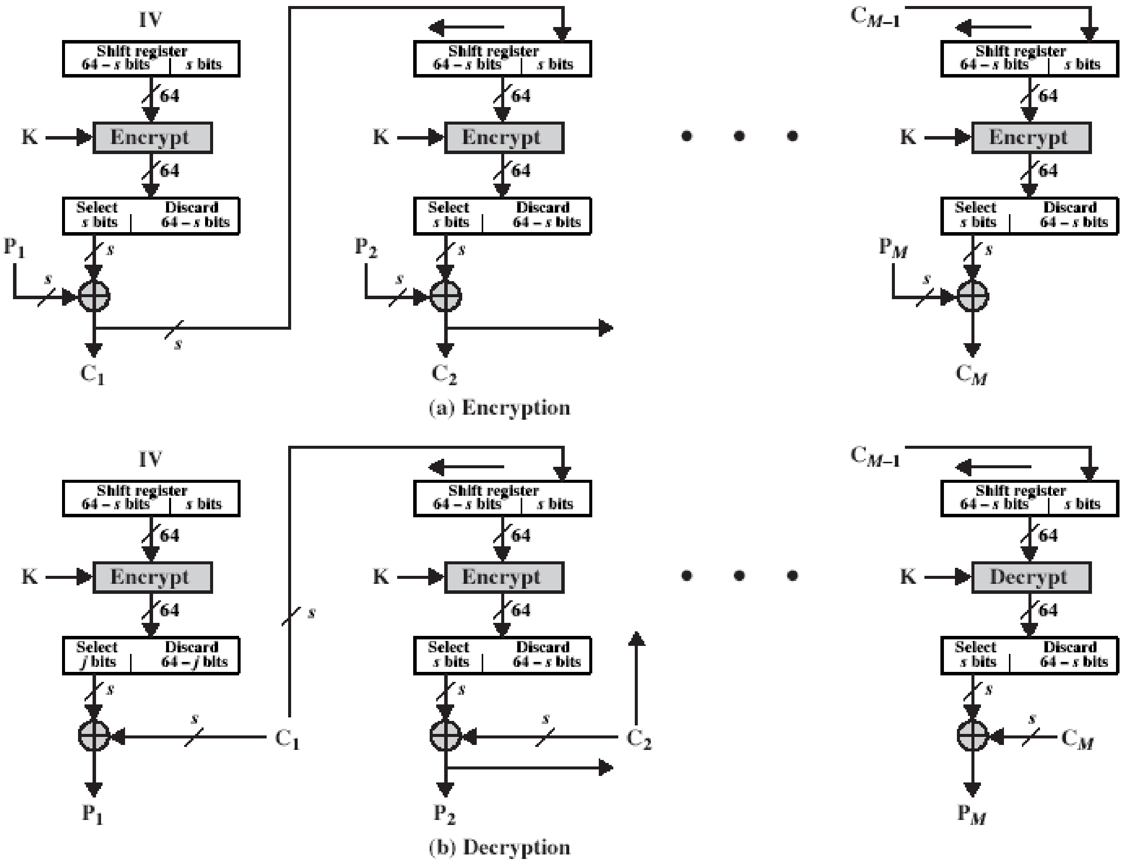


Figure 3.13 *s*-bit Cipher Feedback (CFB) Mode

In Fig. 3.13, it is assumed that the unit of transmission is s bits; usually, s=8. As with CBC, the units of plaintext are chained together, so that the ciphertext of any plaintext unit is a function of all the preceding plaintext. In this case, rather than units of 64 bits, the plaintext is divided into segments of s bits.

Consider encryption. The input to the encryption function is a 64-bit shift register that is initially set to some initialization vector (IV). The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of cipher-text C1, which is then transmitted. In addition, the contents of the shift register

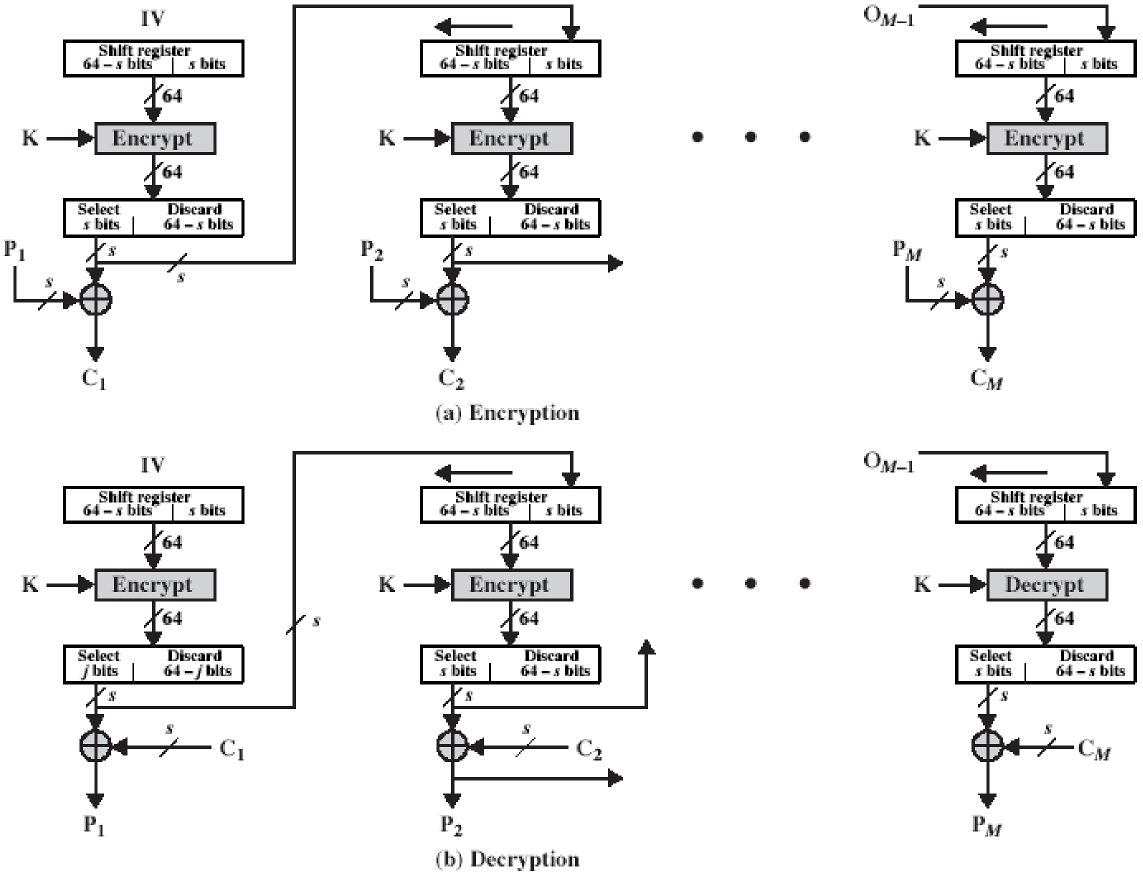
CIPHER FEEDBACK MODE (CONT 1)

are shifted left by s bits and C1 is placed in the rightmost (least significant) s bits of the shift register. This process continues until all plaintext units have been encrypted.

For decryption, the same scheme is used except that the received cipher-text unit is XORed with the output of the encryption function to produce the plaintext unit.

OUTPUT FEEDBACK MODE

The Output Feedback Mode (OFB) is similar in structure to that of CFB:



(a) Encryption

(b) Decryption

Figure 3.14 *s*-bit Output Feedback (OFB) Mode

As can be seen, it is the output of the encryption function that is fed back to the shift register in OFB, whereas in CFB the cipher-text unit is fed back to the shift register. One advantage of the OFB method is that bit errors in transmission do not propagate.

COUNTER MODE

A counter, equal to the plaintext block size is used. The only requirement stated in SP 800-38 A(http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf, NIST Special Publication 800 -38 A, 2001 Edition, Morris Dworkin, Recommendations for Block Cipher Modes of Operation) is that the counter value must be different for each plaintext block that is encrypted. This mode is with applications to ATM (asynchronous transfer mode) and IPsec (IP security) nowadays, but it was proposed in 1979.

Counter Mode works as follows:

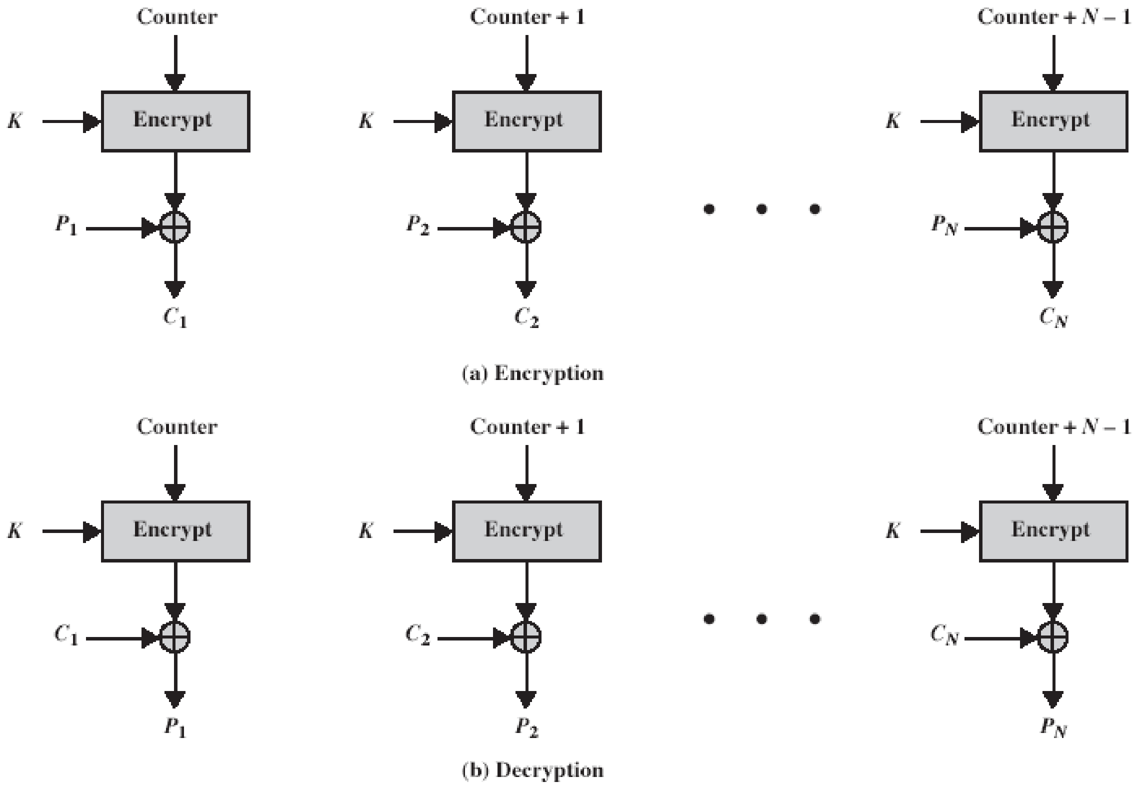


Figure 3.15 Counter (CTR) Mode

Addition is made modulo 2^b, where b is a block size. CTR mode is effective because blocks may be processed in parallel; encryption of keys may be made in advance, and only XOR will be made on-line; only necessary blocks may be decrypted; provides not less security than chaining modes but significantly simpler.