Classical Encryption Techniques

Prof. A. A. Daptardar
Asst. Prof. Dept. Of CSE
HIT, Nidasoshi

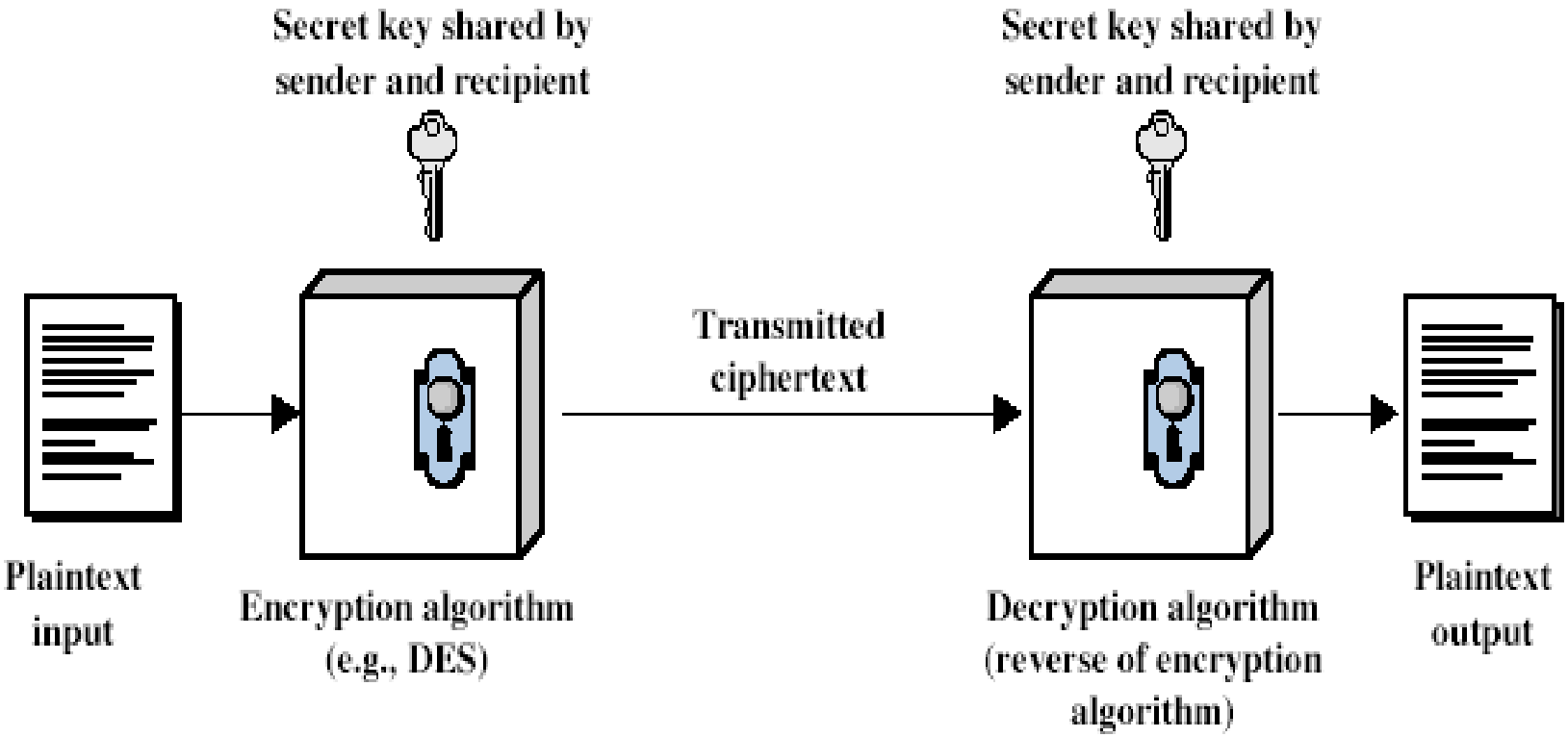
Symmetric Encryption

- or conventional / private-key / single-key
- sender and recipient share a common key
- all classical encryption algorithms are private-key
- was only type prior to invention of public-key in 1970’s
- and by far most widely used

Some Basic Terminology

- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

Symmetric Cipher Model



Requirements

- two requirements for secure use of symmetric encryption:
  - a strong encryption algorithm
  - a secret key known only to sender / receiver
- mathematically have:

  $Y = E_K(X)$

  $X = D_K(Y)$
- assume encryption algorithm is known
- implies a secure channel to distribute key

Model of Symmetric Cryptosystem

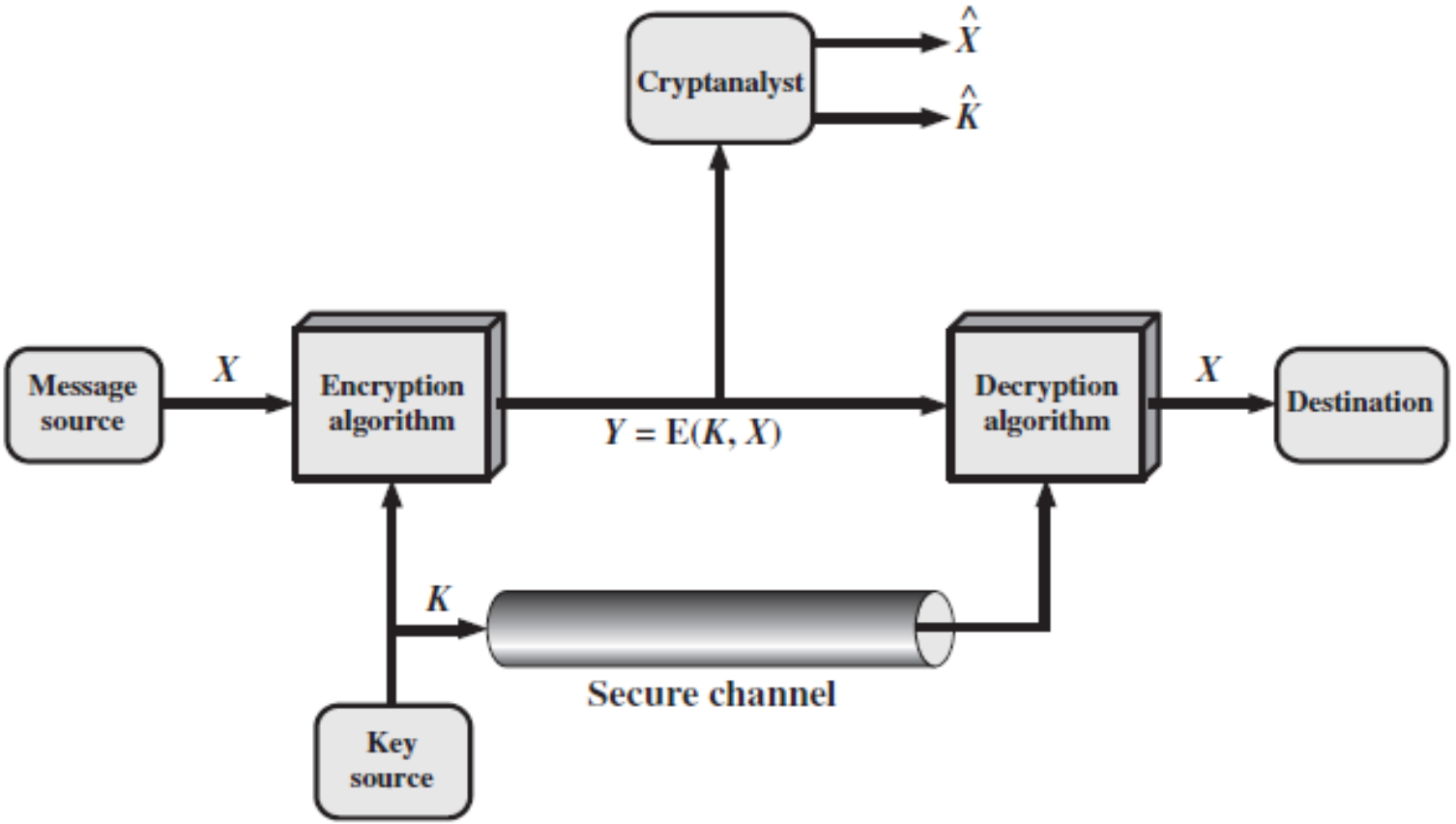


Figure 2.2 Model of Symmetric Cryptosystem

Cryptography

- characterize cryptographic system by:
  - type of operations used for transforming plaintext to ciphertext
    - substitution – in which each element in the plaintext is mapped into another element
    - transposition – in which elements in the plaintext are rearranged
  - number of keys used
    - single-key or private / two-key or public
  - way in which plaintext is processed
    - block / stream

Cryptanalysis

- objective to recover key not just message
- general approaches:
  - Cryptanalysis – rely on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext or some sample plaintext-ciphertext pairs
  - brute-force attack – Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained

Cryptanalytic Attacks

Table 2.1 Types of Attacks on Encrypted Messages

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

More Definitions

- **unconditional security**
  - no matter how much computer power or time is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
- **computational security**
  - given limited computing resources (eg time needed for calculations is greater than age of universe), the cipher cannot be broken

Brute Force Attack

- always possible to simply try every key
- most basic attack, proportional to key size
- assume either know / recognise plaintext

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/μs	Time required at 10^6 decryptions/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 = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

Classical Substitution Ciphers

- It is a one in which the letters of plaintext are replaced by other letters or by numbers or symbols
- or if plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

Caesar Cipher

- earliest known substitution cipher by Julius Caesar
- first attested use in military affairs
- The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.
- can define transformation as:

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
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

Caesar Cipher

- Ex:

  meet me after the toga party

  PHHW PH DIWHU WKH WRJD SDUWB

- mathematically give each letter a number

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

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

- then have Caesar cipher as:

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

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

Cryptanalysis of Caesar Cipher

- only have 26 possible ciphers
  - A maps to A,B,..Z
- could simply try each in turn
- a **brute force search**
  - The encryption and decryption algorithms are known.
  - There are only 25 keys to try.
  - The language of the plaintext is known and easily recognizable.

Monoalphabetic Cipher

- rather than just shifting the alphabet
- could shuffle (jumble) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

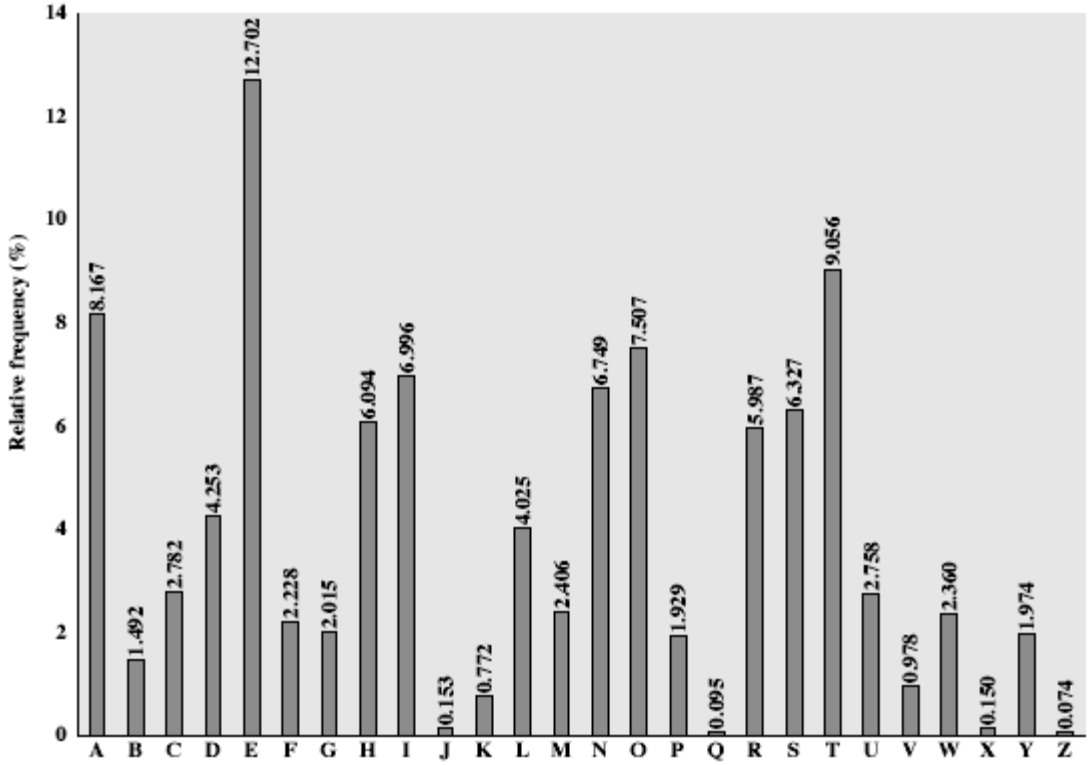
Monoalphabetic Cipher Security

- now have a total of 26! = 4 x 10^{26} keys
- with so many keys, might think is secure
- but would be **!!!WRONG!!!**
- problem is language characteristics

Language Redundancy and Cryptanalysis

- human languages are **redundant**
- eg "th lrd s m shphrd shll nt wnt"
- letters are not equally commonly used
- in English E is by far the most common letter
  - followed by T,R,N,I,O,A,S
- other letters like Z,J,K,Q,X are fairly rare
- have tables of single, double & triple letter frequencies for various languages

English Letter Frequencies



Use in Cryptanalysis

- key concept - monoalphabetic substitution ciphers do not change relative letter frequencies
- discovered by Arabian scientists in 9th century
- calculate letter frequencies for ciphertext
- compare counts/plots against known values
- if caesar cipher look for common peaks/troughs
  - peaks at: A-E-I triple, NO pair, RST triple
  - troughs at: JK, X-Z
- for monoalphabetic must identify each letter
  - tables of common double/triple letters help

Example Cryptanalysis

- given ciphertext:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

- count relative letter frequencies (see text)
- guess P & Z are e and t
- guess ZW is th and hence ZWP is the
- proceeding with trial and error finally get:

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

Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improve security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (sans duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

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

Encrypting and Decrypting

- plaintext is encrypted two letters at a time
  1. if a pair is a repeated letter, insert filler like 'X’
  2. if both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
  3. if both letters fall in the same column, replace each with the letter below it (again wrapping to top from bottom)
  4. otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair

Security of Playfair Cipher

- security much improved over monoalphabetic
- since have 26 x 26 = 676 digrams
- would need a 676 entry frequency table to analyse (verses 26 for a monoalphabetic)
- and correspondingly more ciphertext
- was widely used for many years
  - eg. by US & British military in WW1
- it **can** be broken, given a few hundred letters
- since still has much of plaintext structure

Polyalphabetic Ciphers

- **Another way to improve on the simple monoalphabetic cipher technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message.**
- **polyalphabetic substitution ciphers**
  - A set of related monoalphabetic substitution rules is used
  - A key determines which particular rule is chosen for a given transformation
- improve security using multiple cipher alphabets
- make cryptanalysis harder with more alphabets to guess and flatter frequency distribution

Vigenère Cipher

- simplest polyalphabetic substitution cipher
- In this scheme, the set of related monoaphabetic substitution rules consists of the 26 Caesar Ciphers with shift of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a.

- We can express the Vigenere cipher in the following manner:
- Assume a sequence of plaintext letters $P = p_0, p_1, p_2, \ldots\ldots, p_{n-1}$ *and a key consisting of the* sequence of letters $K = k_0, k_1, k_2, \ldots\ldots, k_{m-1}$, *where typically* $m < n$.
- *The sequence of* ciphertext letters $C = C_0, C_1, C_2, \ldots\ldots, C_{n-1}$ *is calculated as follows:*

- $C = C_0, C_1, C_2, \ldots\ldots, C_{n-1} = E(K, P)$
- $= E[(k_0, k_1, k_2, \ldots\ldots, k_{m-1}), (p_0, p_1, p_2, \ldots\ldots, p_{n-1})]$
- $= (p_0 + k_0) \bmod 26, (p_1 + k_1) \bmod 26, \ldots\ldots, (p_{m-1} + k_{m-1}) \bmod 26, \ldots\ldots\ldots\ldots\ldots$

  $(p_m + k_0) \bmod 26, (p_{m+1} + k_1) \bmod 26, \ldots\ldots, (p_{2m-1} + k_{m-1}) \bmod 26, \ldots\ldots\ldots$

- 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 m letters of the plaintext.
- For the next m 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

$$C_i = (p_i + k_{i \bmod m}) \bmod 26 \qquad \textbf{(2.3)}$$

- Similarly, decryption is a generalization of Equation

$$p_i = (C_i - k_{i \bmod m}) \bmod 26 \qquad \textbf{(2.4)}$$

Example of Vigenère Cipher

- write the plaintext out
- write the keyword repeated above it
- use each key letter as a caesar cipher key
- encrypt the corresponding plaintext letter
- eg using keyword *deceptive*

  key: deceptivedeceptivedeceptive

  plaintext: wearediscoveredsaveyourself

  ciphertext:ZICVTWQNGRZGVTWAVZHCQYGLMGJ

- Expressed numerically, we get the following result

key	3	4	2	4	15	19	8	21	4	3	4	2	4	15
plaintext	22	4	0	17	4	3	8	18	2	14	21	4	17	4
ciphertext	25	8	2	21	19	22	16	13	6	17	25	6	21	19

key	19	8	21	4	3	4	2	4	15	19	8	21	4
plaintext	3	18	0	21	4	24	14	20	17	18	4	11	5
ciphertext	22	0	21	25	7	2	16	24	6	11	12	6	9

Security of Vigenère Ciphers

- have multiple ciphertext letters for each plaintext letter
- hence letter frequencies are obscured

- repetitions in ciphertext give clues to period
- so find same plaintext an exact period apart
- which results in the same ciphertext
- of course, could also be random fluke
- eg repeated “VTW” in previous example
- suggests size of 3 or 9
- then attack each monoalphabetic cipher individually using same techniques as before

Autokey Cipher

- ideally want a key as long as the message
- Vigenère proposed the **autokey** cipher
- with keyword is prefixed to message as key
- knowing keyword can recover the first few letters
- use these in turn on the rest of the message
- but still have frequency characteristics to attack
- eg. given key *deceptive*

  key: deceptivewearediscoveredsav
  plaintext: wearediscoveredsaveyourself
  ciphertext:ZICVTWQNGKZEIIGASXSTSLVVWLA

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.

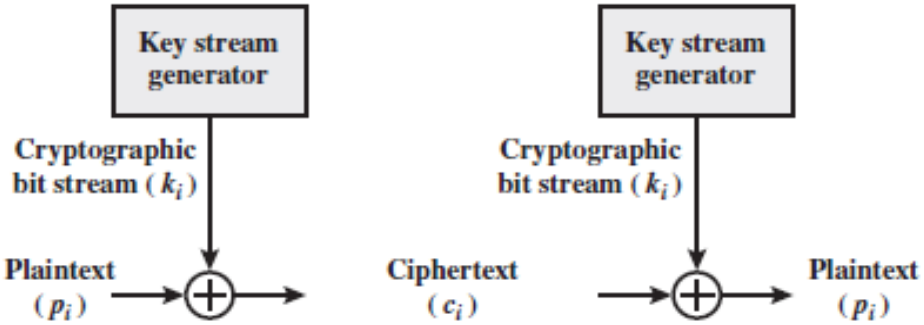


Figure 2.7 Vernam Cipher

- His system works on binary data (bits) rather than letters.
- The system can be expressed succinctly as follows (Figure 2.7):
- $c_i = p_i \oplus k_i$
- 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
- Because of the properties of the XOR, decryption simply involves the same bitwise operation:
- $p_i = c_i \oplus k_i$

One-Time Pad

- If a truly random key as long as the message is used, the cipher will be secure called a One-Time pad.
- It is unbreakable since ciphertext bears no statistical relationship to the plaintext.
- Since for **any plaintext** & **any ciphertext** there exists, a key mapping one to other.
- It can only use the key **once** though problems in generation & safe distribution of key

Plaintext	w	e	l	c	o	m	e	s	y	o	u
Plaintext value	23	5	12	3	15	13	5	19	25	15	21
One time pad	e	i	g	h	t	h	c	l	a	s	s
One time pad value	5	9	7	8	20	8	3	12	1	19	19
Ciphertext Value	28	14	19	11	35	21	8	31	26	34	40
	2				9			5		8	14
Ciphertext	B	N	S	K	I	U	H	E	Z	H	N

Summary

- have considered:
  - classical cipher techniques and terminology
  - monoalphabetic substitution ciphers
  - cryptanalysis using letter frequencies
  - Playfair cipher
  - polyalphabetic ciphers
  - transposition ciphers
  - product ciphers and rotor machines
  - stenography

Block Ciphers and The Data Encryption Standard (DES)

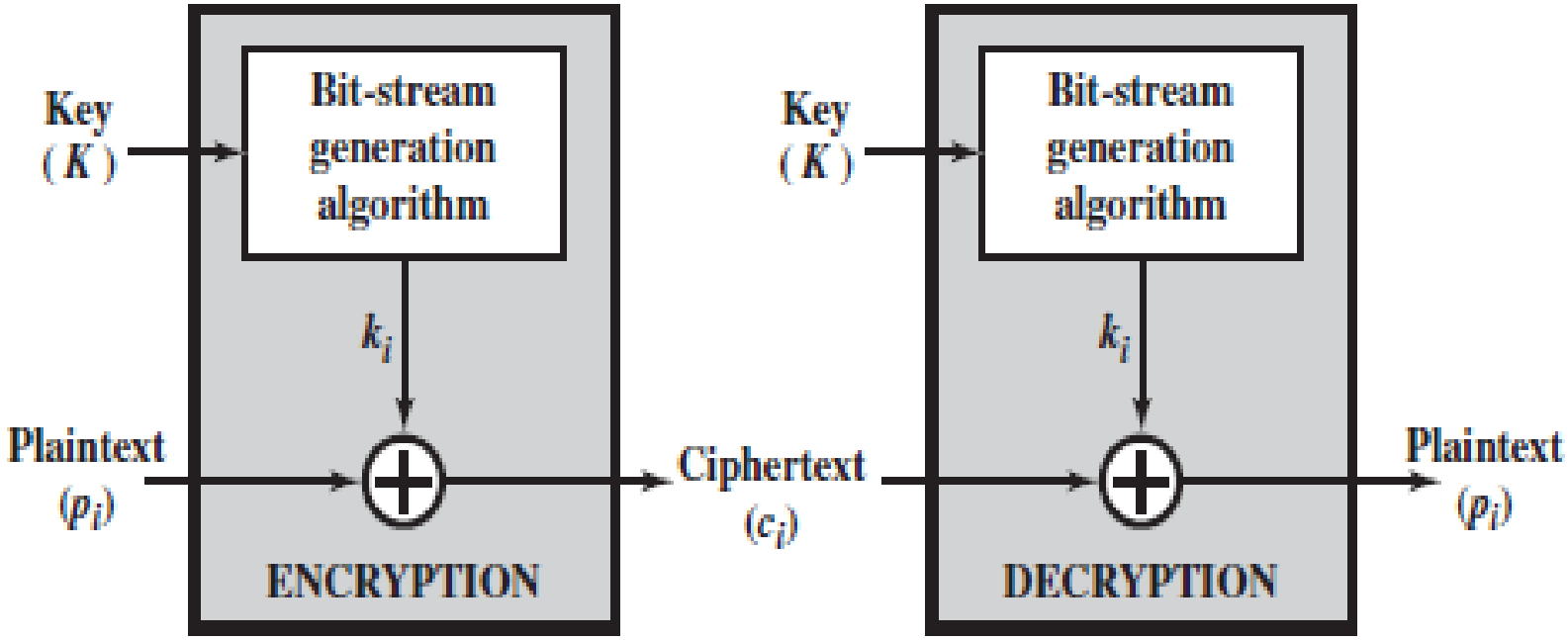
Traditional Block Cipher Structure

- Many symmetric block encryption algorithms in current use are based on a structure referred to as a Feistel block cipher.
- For that reason, it is important to examine the design principles of the Feistel cipher.

Stream Ciphers

- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
- Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.
- In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream (k_i) is as long as the plaintext bit stream (p_i).
- If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream.

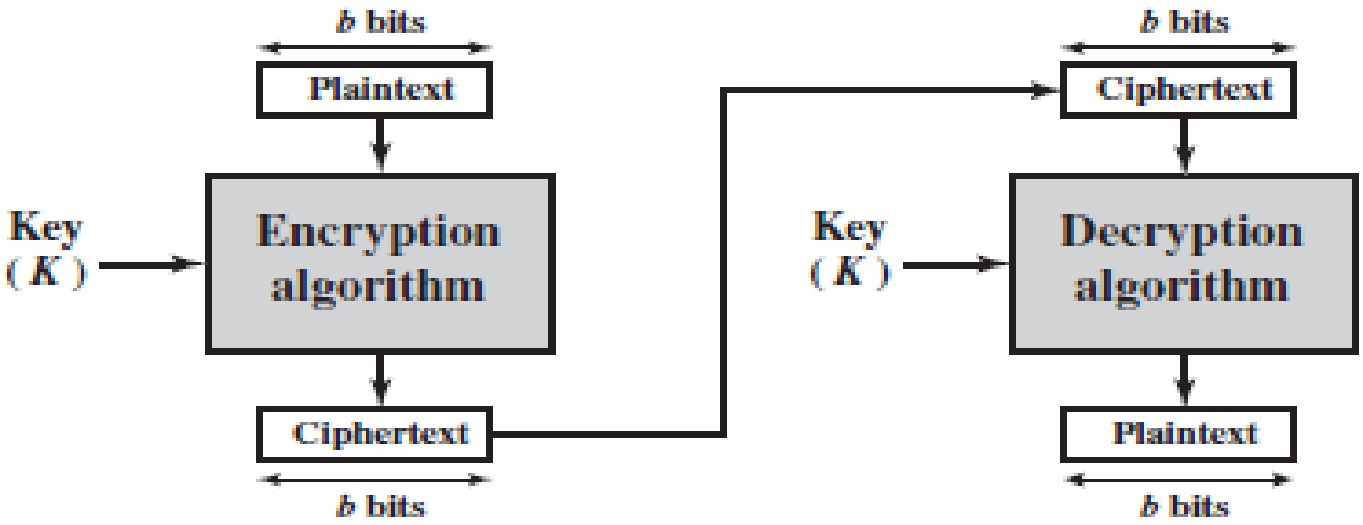
- Accordingly, for practical reasons, the bit-stream generator must be implemented as an algorithmic procedure, so that the cryptographic bit stream can be produced by both users.
- In this approach (Fig. 3.1a), the bit-stream generator is a key-controlled algorithm and must produce a bit stream that is cryptographically strong.
- That is, it must be computationally impractical to predict future portions of the bit stream based on previous portions of the bit stream.
- The two users need only share the generating key, and each can produce the keystream.



(a) Stream cipher using algorithmic bit-stream generator

Block Cipher Principles

- 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.
- The two users share a symmetric encryption key (Figure 3.1b).
- Using some of the modes of operation a block cipher can be used to achieve the same effect as a stream cipher.



(b) Block cipher

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), each must produce a unique ciphertext block.
- Such a transformation is called reversible, or nonsingular.
- The following examples illustrate nonsingular and singular transformations for n = 2.

Reversible Mapping	
Plaintext	**Ciphertext**
00	11
01	10
10	00
11	01

Irreversible Mapping	
Plaintext	**Ciphertext**
00	11
01	10
10	01
11	01

- Figure 3.2 illustrates the logic of a general substitution cipher for n = 4.
- A 4-bit input produces one of 16 possible input states, which is mapped by the substitution cipher into a unique one of 16 possible output states, each of which is represented by 4 ciphertext bits.
- The encryption and decryption mappings can be defined by a tabulation, as shown in Table 3.1.

Ideal Block Cipher

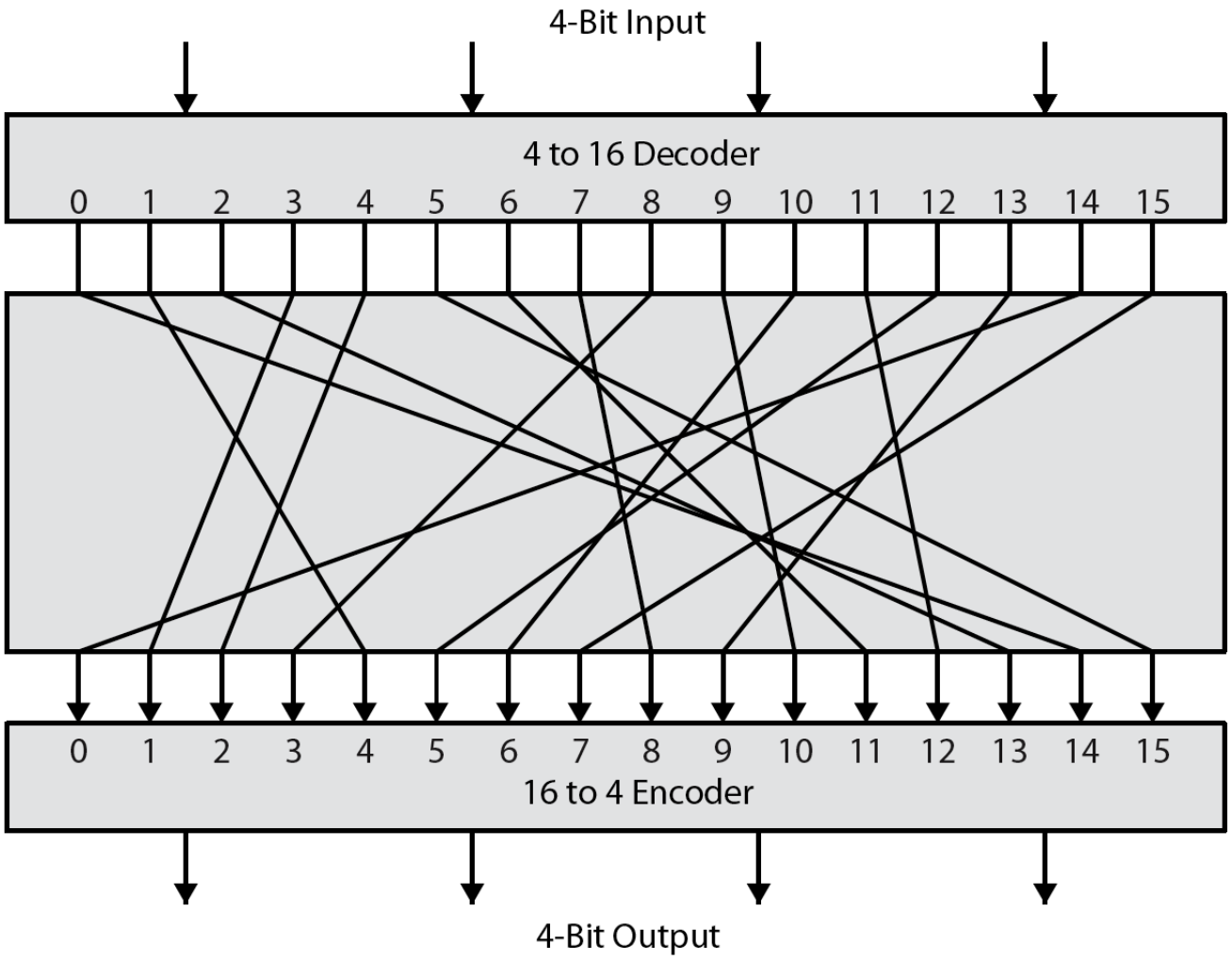


Table 3.1 Encryption and Decryption Tables for Substitution Cipher of Figure 3.2

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

- Feistel refers to this as the ideal block cipher, because it allows for the maximum number of possible encryption mappings from the plaintext block.
- There is a practical problem with the ideal block cipher.
- If a small block size, such as n = 4, is used, then the system is equivalent to a classical substitution cipher.
- Such systems, as we have seen, are vulnerable to a statistical analysis of the plaintext.
- In general, for an n-bit ideal block cipher, the length of the key defined in this fashion is $n * 2^n$ bits.
- For a 64-bit block, which is a desirable length to thwart statistical attacks, the required key length is $64 * 2^{64} = 2^{70} \approx 10^{21}$ bits.

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.
- Feistel's is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions.

Confusion and Diffusion

- Shannon suggests two methods for frustrating statistical cryptanalysis :
- diffusion – dissipates statistical structure of plaintext over bulk of ciphertext
- confusion – makes relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible.

Feistel Cipher Structure

- Horst Feistel devised the **feistel cipher**
  - based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which
  - perform a substitution on left data half
  - based on round function of right half & subkey
  - then have permutation swapping halves
- implements Shannon's S-P net concept

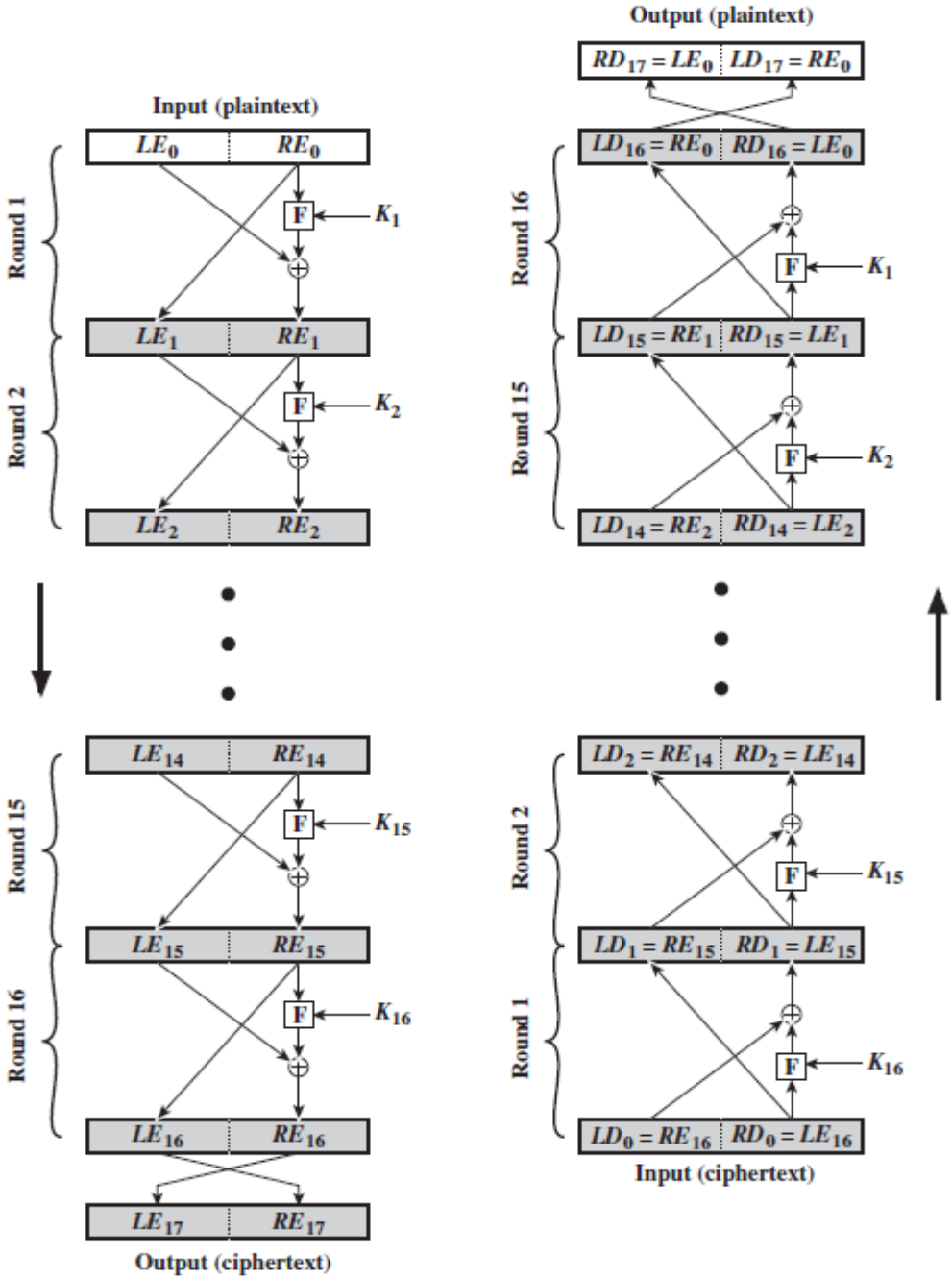


Figure 3.3 Feistel Encryption and Decryption (16 rounds)

Feistel Cipher Design Elements

- block size – 64 bits
- key size – 128 bits
- number of rounds – 16 rounds
- subkey generation algorithm -
- round function
- fast software en/decryption
- ease of analysis

Data Encryption Standard (DES)

- most widely used block cipher in world
- adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security

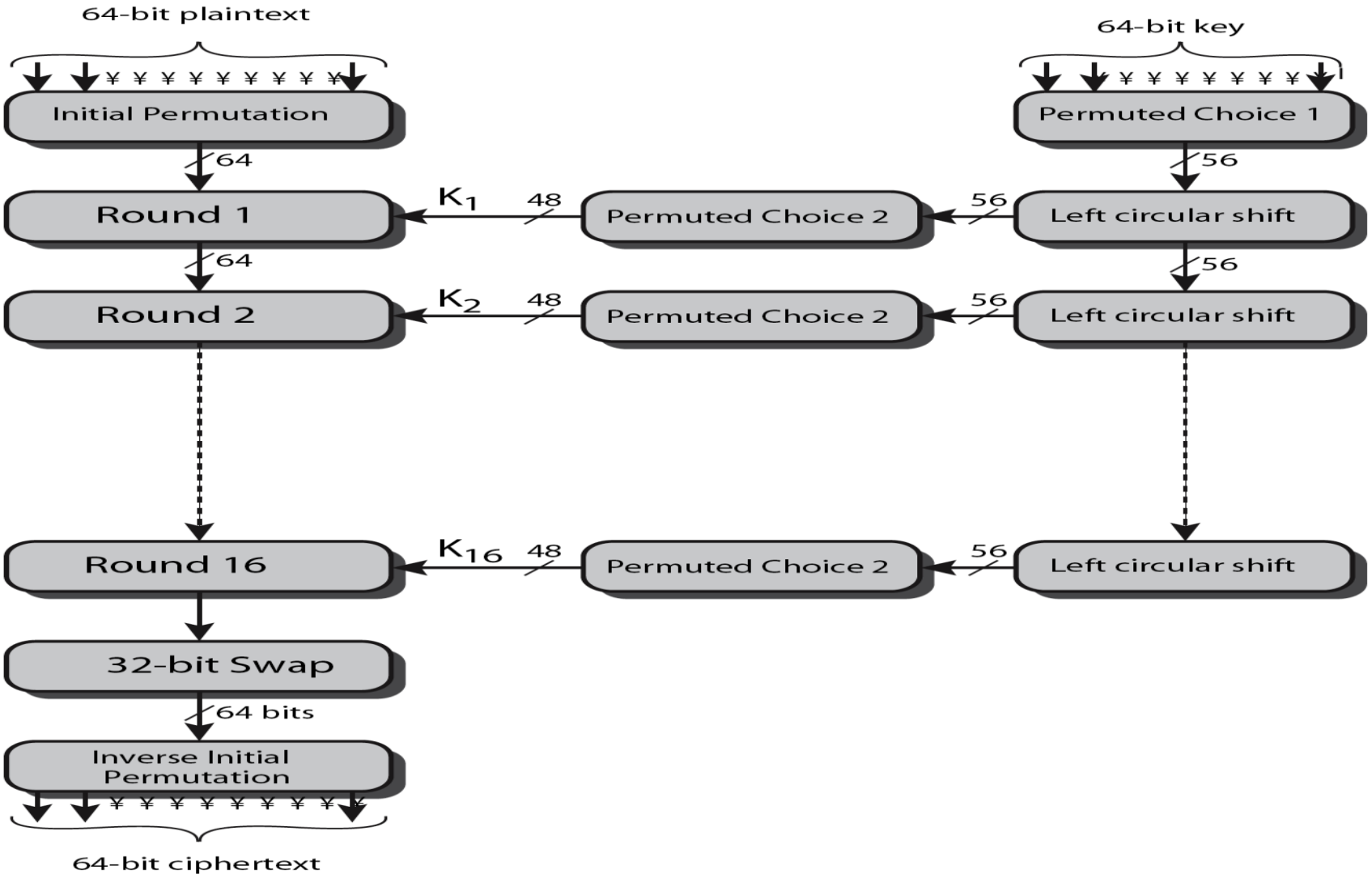
DES History

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

DES Design Controversy

- although DES standard is public
- was considerable controversy over design
  - in choice of 56-bit key (vs Lucifer 128-bit)
  - and because design criteria were classified
- subsequent events and public analysis show in fact design was appropriate
- use of DES has flourished
  - especially in financial applications
  - still standardised for legacy application use

DES Encryption Overview



Initial Permutation IP

- first step of the data computation
- IP reorders the input data bits
- even bits to LH half, odd bits to RH half
- quite regular in structure (easy in h/w)
- example:

```
IP(675a6967 5e5a6b5a) = (ffb2194d
004df6fb)
```

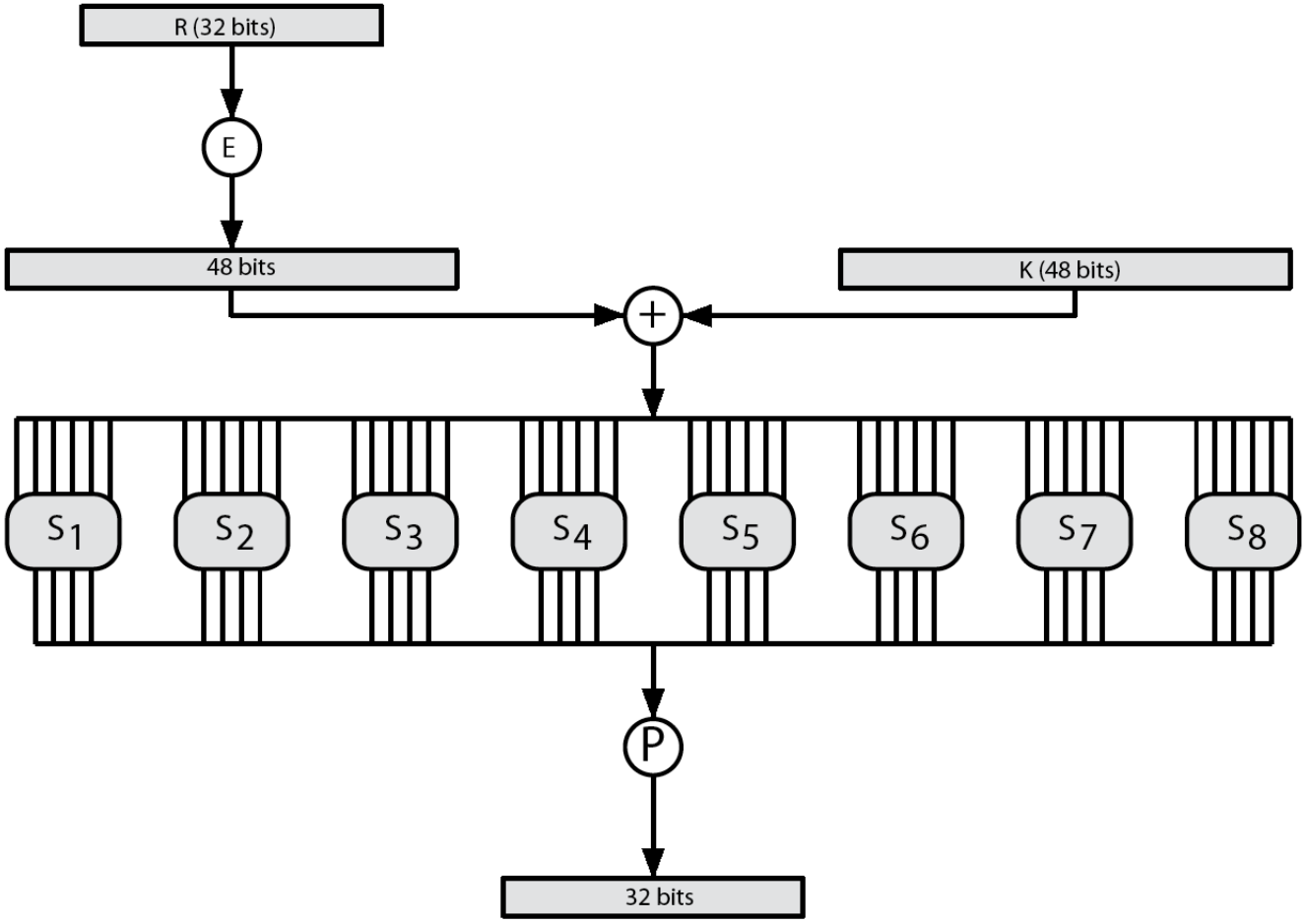
DES Round Structure

- uses two 32-bit L & R halves
- as for any Feistel cipher can describe as:

  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i)$
- F takes 32-bit R half and 48-bit subkey:
  - expands R to 48-bits using perm E
  - adds to subkey using XOR
  - passes through 8 S-boxes to get 32-bit result
  - finally permutes using 32-bit perm P

DES Round Structure



Substitution Boxes S

- have eight S-boxes which map 6 to 4 bits
- each S-box is actually 4 little 4 bit boxes
  - outer bits 1 & 6 (**row** bits) select one row of 4
  - inner bits 2-5 (**col** bits) are substituted
  - result is 8 lots of 4 bits, or 32 bits
- row selection depends on both data & key
  - feature known as autoclaving (autokeying)
- example:
  - `S(18 09 12 3d 11 17 38 39) = 5fd25e03`

DES Key Schedule

- forms subkeys used in each round
  - initial permutation of the key (PC1) which selects 56-bits in two 28-bit halves
  - 16 stages consisting of:
    - rotating **each half** separately either 1 or 2 places depending on the **key rotation schedule** K
    - selecting 24-bits from each half & permuting them by PC2 for use in round function F
- note practical use issues in h/w vs s/w

DES Decryption

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

A DES Example

- For this example, the plaintext is a exadecimal palindrome. The plaintext, key, and resulting ciphertext are as follows:

Plaintext:	`02468aceeca86420`
Key:	`0f1571c947d9e859`
Ciphertext:	`da02ce3a89ecac3b`

Table 3.2 DES Example

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^{-1}		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.

- Table 3.3 shows the result when the fourth bit of the plaintext is changed, so that the plaintext is **12468aceeca86420.**
- The second column of the table shows the intermediate 64-bit values at the end of each round for the two plaintexts.
- The third column shows the number of bits that differ between the two intermediate values.
- The table shows that, after just three rounds, 18 bits differ between the two blocks.
- On completion, the two ciphertexts differ in 32 bit positions.

Table 3.3 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

- Table 3.4 shows a similar test using the original plaintext of with two keys that differ in only the fourth bit position: the original key, **0f1571c947d9e859, and** the altered key, **1f1571c947d9e859.**
- Again, the results show that about half of the bits in the ciphertext differ and that the avalanche effect is pronounced after just a few rounds.

Table 3.4 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- brute force search looks hard
- recent advances have shown is possible
  - in 1997 on Internet in a few months
  - in 1998 on dedicated h/w (EFF) in a few days
  - in 1999 above combined in 22hrs!
- still must be able to recognize plaintext
- must now consider alternatives to DES

Table 3.5 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Strength of DES – Analytic Attacks

- now have several analytic attacks on DES
- these utilise some deep structure of the cipher
  - by gathering information about encryptions
  - can eventually recover some/all of the sub-key bits
  - if necessary then exhaustively search for the rest
- generally these are statistical attacks
- include
  - differential cryptanalysis
  - linear cryptanalysis
  - related key attacks

Strength of DES – Timing Attacks

- attacks actual implementation of cipher
- use knowledge of consequences of implementation to derive information about some/all subkey bits
- specifically use fact that calculations can take varying times depending on the value of the inputs to it
- particularly problematic on smartcards

Block Cipher Design

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

Summary

- have considered:
  - block vs stream ciphers
  - Feistel cipher design & structure
  - DES
    - details
    - strength
  - Differential & Linear Cryptanalysis
  - block cipher design principles