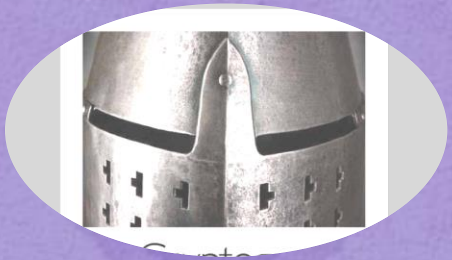


Chapter 4

Block Ciphers and the Data Encryption Standard

Stream Cipher

Encrypts a digital data stream one bit or one byte at a time

Examples:

- Autokeyed Vigenère cipher
- Vernam cipher

In the ideal case, a one-time pad version of the Vernam cipher would be used, in which the keystream is as long as the plaintext bit stream

If the cryptographic keystream is random, then this cipher is unbreakable by any means other than acquiring the keystream

- Keystream must be provided to both users in advance via some independent and secure channel
- This introduces insurmountable logistical problems if the intended data traffic is very large

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

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

Block Cipher

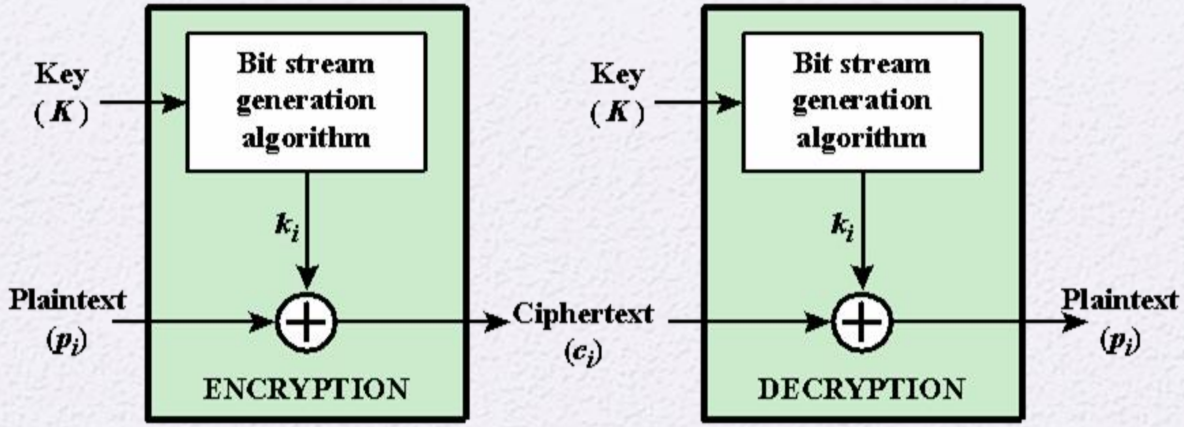
C(k, ,)

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

As with a stream cipher, the two users share a symmetric encryption key

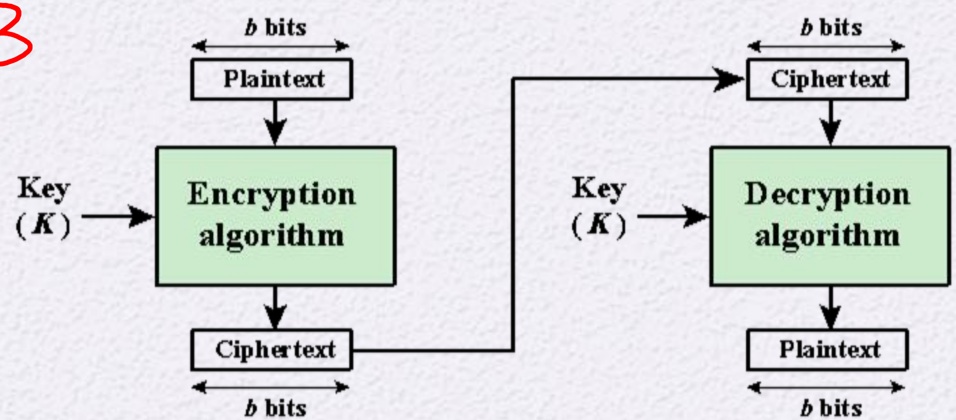
The majority of network-based symmetric cryptographic applications make use of block ciphers



$A \oplus B = C$

$C \oplus A = B$

(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Figure 4.1 Stream Cipher and Block Cipher

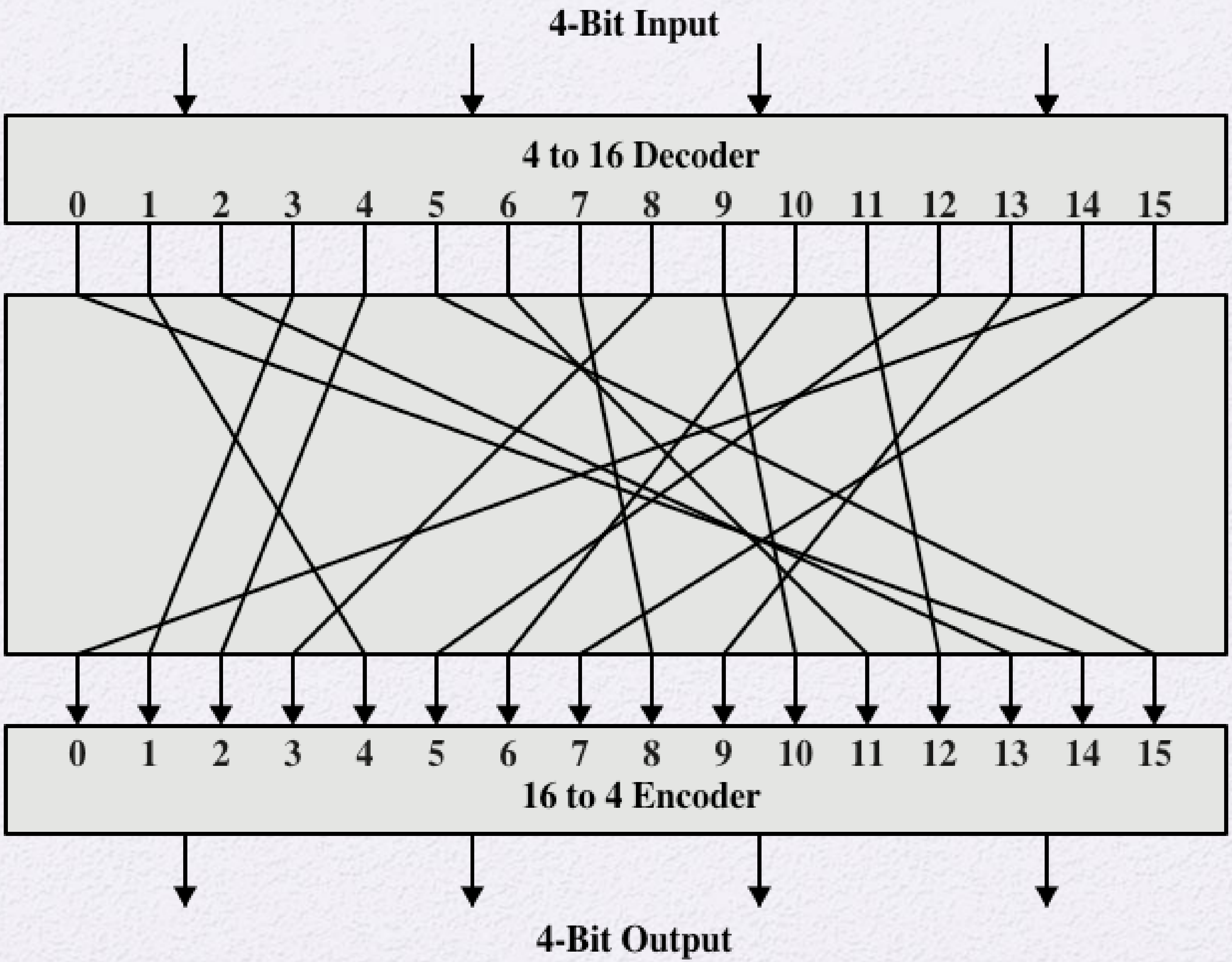


Figure 4.2 General *n*-bit-*n*-bit Block Substitution (shown with $n = 4$)

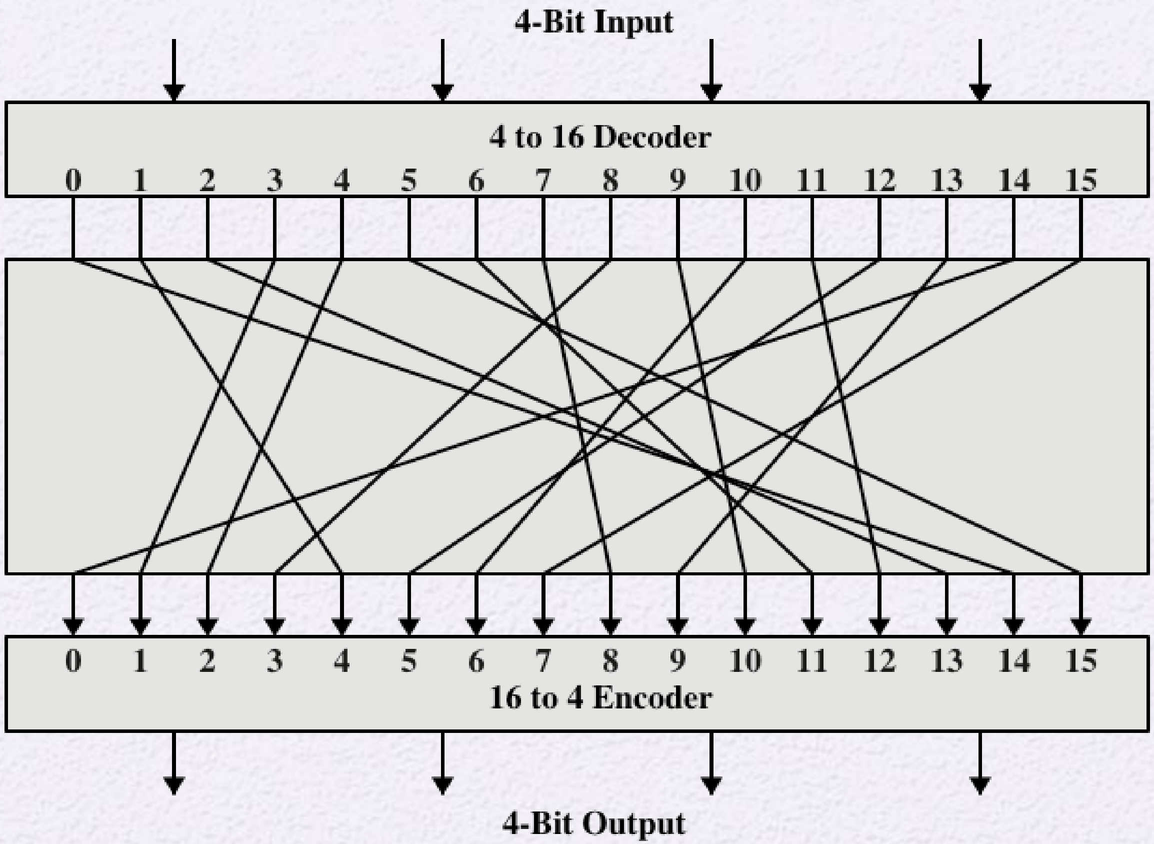


Figure 4.2 General *n*-bit-*n*-bit Block Substitution (shown with *n* = 4)

- Question:
  - What is the key in this diagram?
- Answer:
  - The mapping between the input and output bits

Table 4.1

Encryption and Decryption Tables for Substitution Cipher of Figure 4.2

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

2^n

$2^n! =$

key

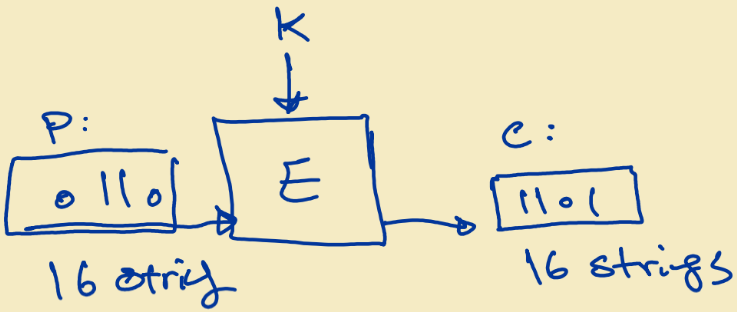
Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

P: 0110 0011 1010 12 bits

1	2	3

length of each block = 4

$n = 4$

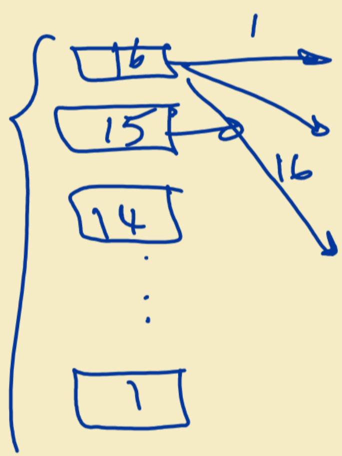


$\underline{2} \cdot \underline{2} \cdot \underline{2} \cdot \underline{2} = 16$ possible strings of length 4.

$2^4 : 16$

#key : # Mappings

$= 2^n! = 16!$



P: k: C:

$0110 \oplus 1011 = 1101$

$16 \cdot 4 = |k|$

$2^n \cdot n = |k|$

Ideal Block Cipher

- The scheme on the previous slides is described as an ideal block cipher
  - Because it allows the maximum number of possible encryptions
- For a block cipher with n-bit block
  - A single key will require **2^n mappings** to be stored
  - $2^n!$ possible keys are possible
- The size of the key ($n.2^n$) can be very large for large values of n and this makes the ideal block cipher impractical for real world use

Product Cipher

- For this reason, most modern block ciphers are product ciphers
- The product cipher combines a sequence of simple transformations to complete an encryption
- Each transformation is weak but the repeated application of simple transformations in the sequence leads to strong encryption

Diffusion and Confusion

- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
  - Shannon's concern was to thwart cryptanalysis based on statistical analysis

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

Confusion

- Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
- 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

Diffusion and Confusion

- From a cryptanalysis point of view, we can define diffusion and confusion as follows:

Diffusion

- Prevent the prediction of the key by analyzing the relationship between the plaintext and the cipher text

Confusion

- Prevent the prediction of the key by analyzing the relationship between the ciphertext and the key

- A good product cipher should therefore contain transformations that enhance both confusion and diffusion

Feistel Cipher

- Feistel proposed the use of a cipher that alternates substitutions and permutations

Substitutions

- Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements

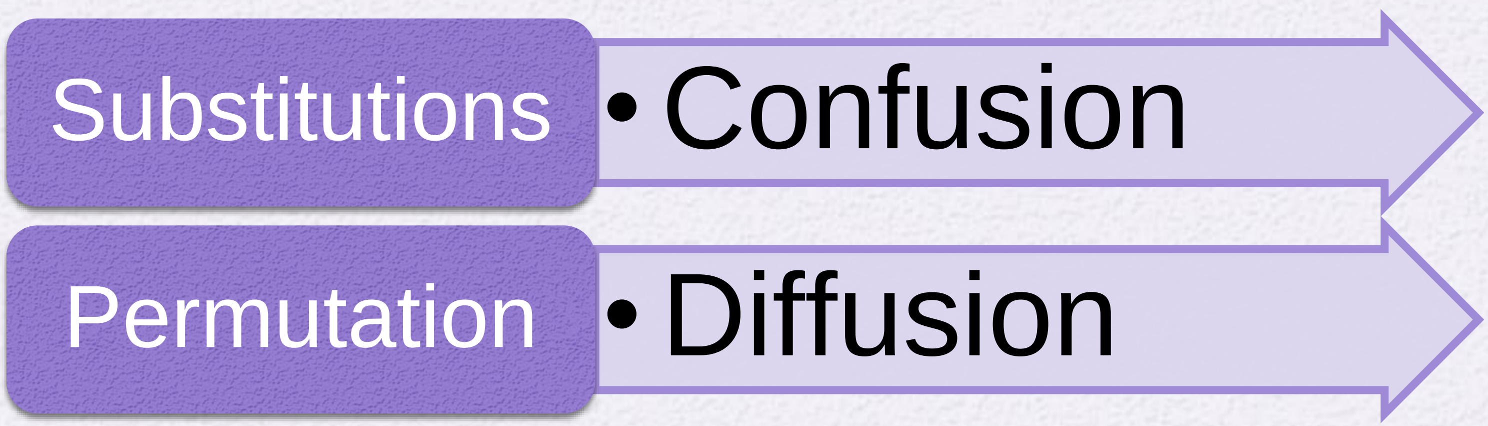
Permutation

- No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

- Is a practical application of a proposal by Claude Shannon to develop a product cipher that alternates confusion and diffusion functions
- Is the structure used by many significant symmetric block ciphers currently in use

Feistel Cipher

- In a Feistel Cipher substitutions provide confusion and permutations provide diffusion



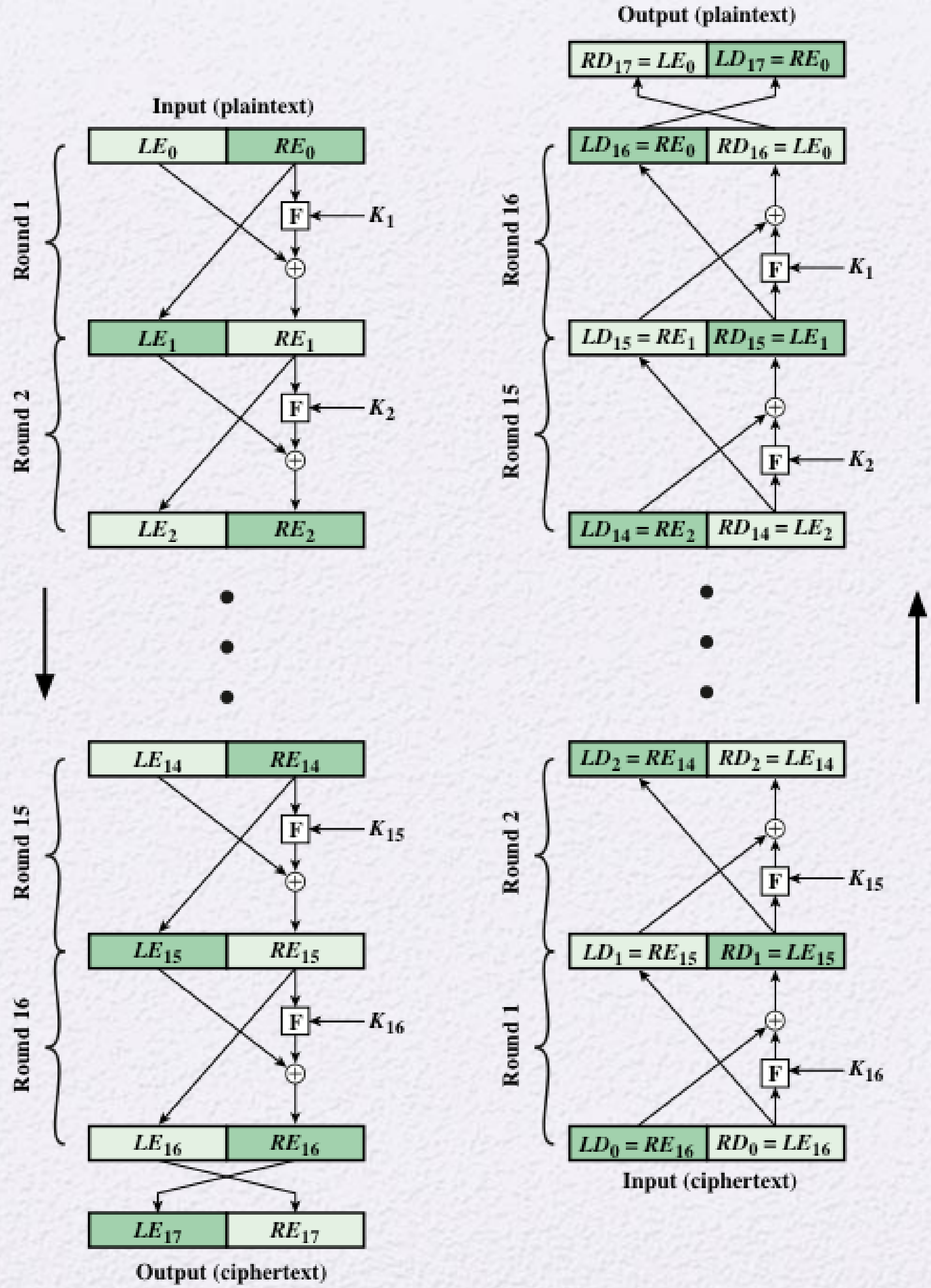
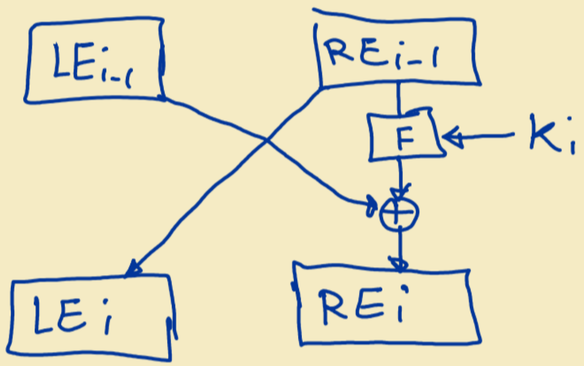
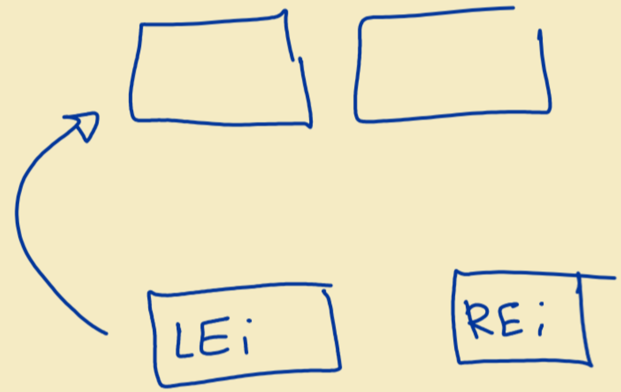


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

Feistel cipher:





$$LE_i = RE_{i-1}$$

$$RE_i = F(RE_{i-1}, k_i) \oplus \boxed{LE_{i-1}}$$

$A \oplus B = C \qquad B = A \oplus C$

$$RE_{i-1} = LE_i$$

$$LE_{i-1} = F(RE_{i-1}, k_i) \oplus RE_i$$

Feistel Cipher Design Features

- Block size
  - Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm
- Key size
  - Larger key size means greater security but may decrease encryption/decryption speeds
- Number of rounds
  - The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security
- Subkey generation algorithm
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis
- Round function F
  - Greater complexity generally means greater resistance to cryptanalysis
- Fast software encryption/decryption
  - In many cases, encrypting is embedded in applications or utility functions in such a way as to preclude a hardware implementation; accordingly, the speed of execution of the algorithm becomes a concern
- Ease of analysis
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

Feistel Example

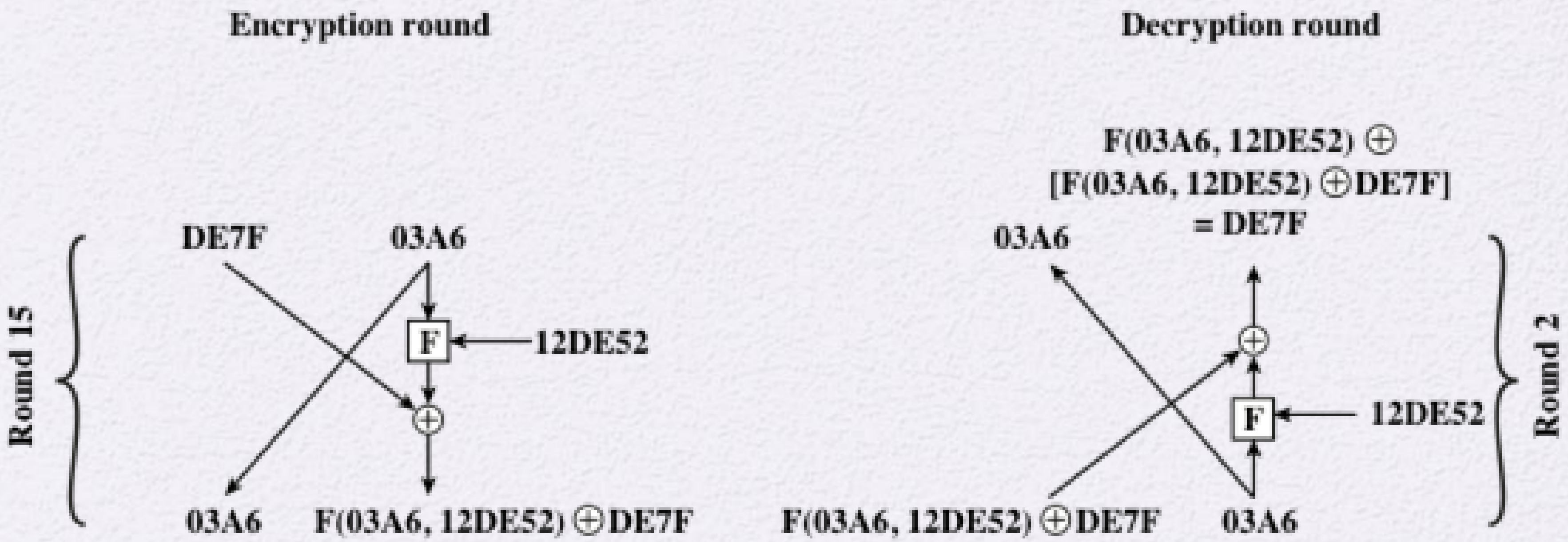


Figure 4.4 Feistel Example

Data Encryption Standard (DES)

- Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46
- Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
- Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  - Data are encrypted in 64-bit blocks using a 56-bit key
  - The algorithm transforms 64-bit input in a series of steps into a 64-bit output
  - The same steps, with the same key, are used to reverse the encryption

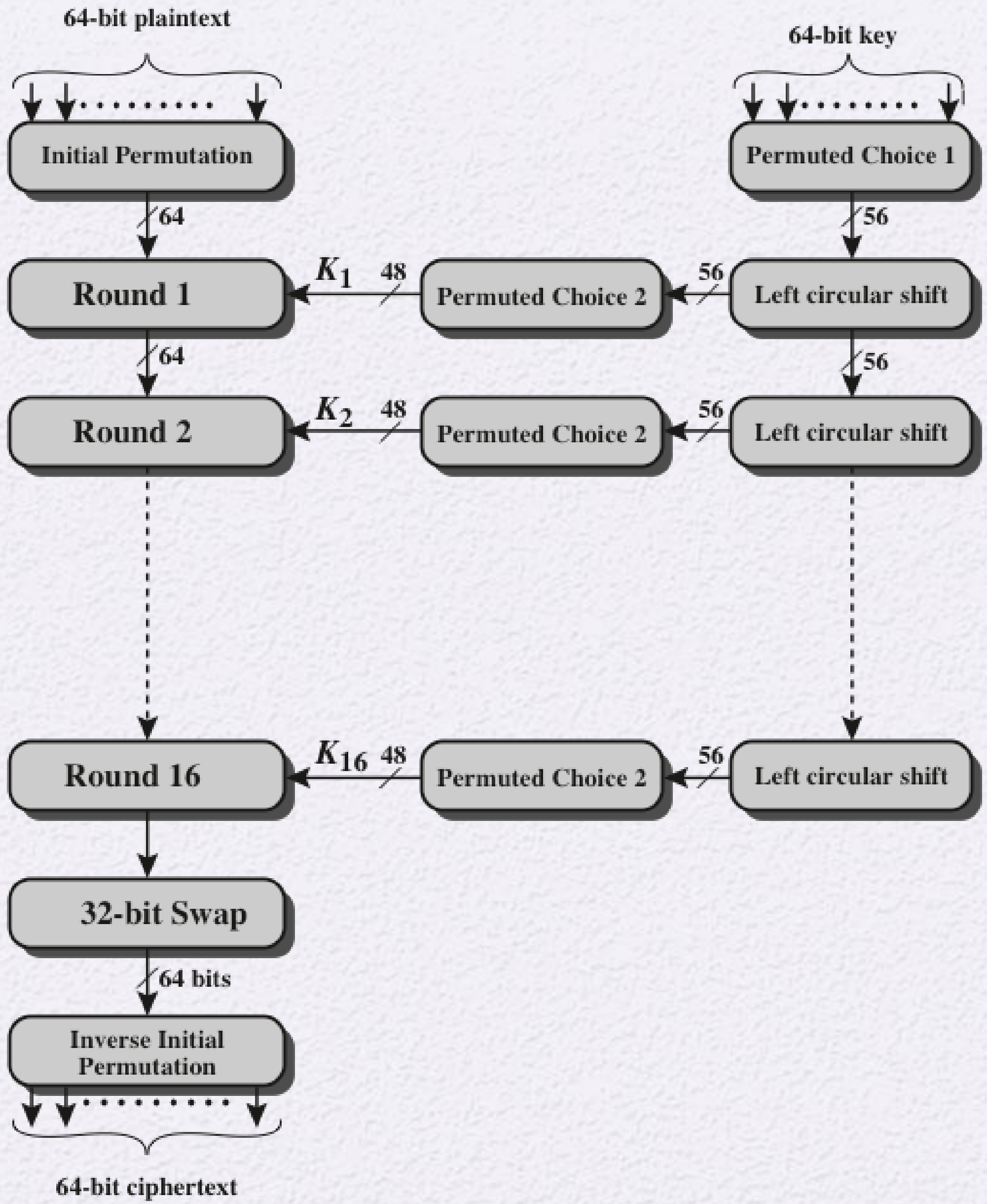


Figure 4.5 General Depiction of DES Encryption Algorithm

Table 4.2

DES Example

(Table can be found on page 106 in the textbook)

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

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

Round		δ
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 4.3 Avalanche Effect in DES: Change in Plaintext

(Table can be found on page 107 in the textbook)

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

Table 4.4 Avalanche Effect in DES: Change in Key

(Table can be found on page 107 in the textbook)

Table 4.5
Average Time Required for Exhaustive Key Search

Key Size (Bits)	Cipher	# of Alternative Keys	Time (10^9 decryptions/sec)	Time (10^{13} decryptions/sec)
56	DES	$2^{56} \approx 7.2 * 10^{16}$	2^{55} ns ≈ 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 * 10^{38}$	2^{127} ns ≈ $5.3 * 10^{21}$ years	$5.3 * 10^{17}$ years
168	Triple DES	$2^{168} \approx 3.7 * 10^{50}$	2^{167} ns ≈ $5.8 * 10^{33}$ years	$5.8 * 10^{29}$ years
192	AES	$2^{192} \approx 6.3 * 10^{57}$	$2^{191} \approx 9.8 * 10^{40}$ years	$9.8 * 10^{36}$ years
256	AES	$2^{256} \approx 1.2 * 10^{77}$	2^{255} ns ≈ $1.8 * 10^{60}$ years	$1.8 * 10^{56}$ years
26 Characters (Permutation)	Monoalphabetic	$26! = 4 * 10^{26}$	$2 * 10^{26}$ ns ≈ $6.3 * 10^{9}$ years	$6.3 * 10^{6}$ years

(Table can be found on page 109 in the textbook)

Strength of DES

- Timing attacks
  - One in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts
  - Exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs
  - So far it appears unlikely that this technique will ever be successful against DES or more powerful symmetric ciphers such as triple DES and AES

Block Cipher Design Principles: Number of Rounds

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

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

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

Block Cipher Design Principles: Design of Function F

- The heart of a Feistel block cipher is the function F
- The more nonlinear F, the more difficult any type of cryptanalysis will be
- The SAC and BIC criteria appear to strengthen the effectiveness of the confusion function

The algorithm should have good avalanche properties

Strict avalanche criterion (SAC)

States that any output bit j of an S-box should change with probability 1/2 when any single input bit i is inverted for all i , j

Bit independence criterion (BIC)

States that output bits j and k should change independently when any single input bit i is inverted for all i , j , and k

Block Cipher Design Principles: Key Schedule Algorithm

- With any Feistel block cipher, the key is used to generate one subkey for each round
- In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key
- It is suggested that, at a minimum, the key schedule should guarantee key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion

Summary

- Understand the distinction between stream ciphers and block ciphers
- Present an overview of the Feistel cipher and explain how decryption is the inverse of encryption
- Present an overview of Data Encryption Standard (DES)

- Explain the concept of the avalanche effect
- Discuss the cryptographic strength of DES
- Summarize the principal block cipher design principles