Cryptography and Network Security: Principles and Practice

Eighth Edition, Global Edition

Chapter 3

Classical Encryption Techniques

Hemant Ghayvat

ChatGPT and Open-AI

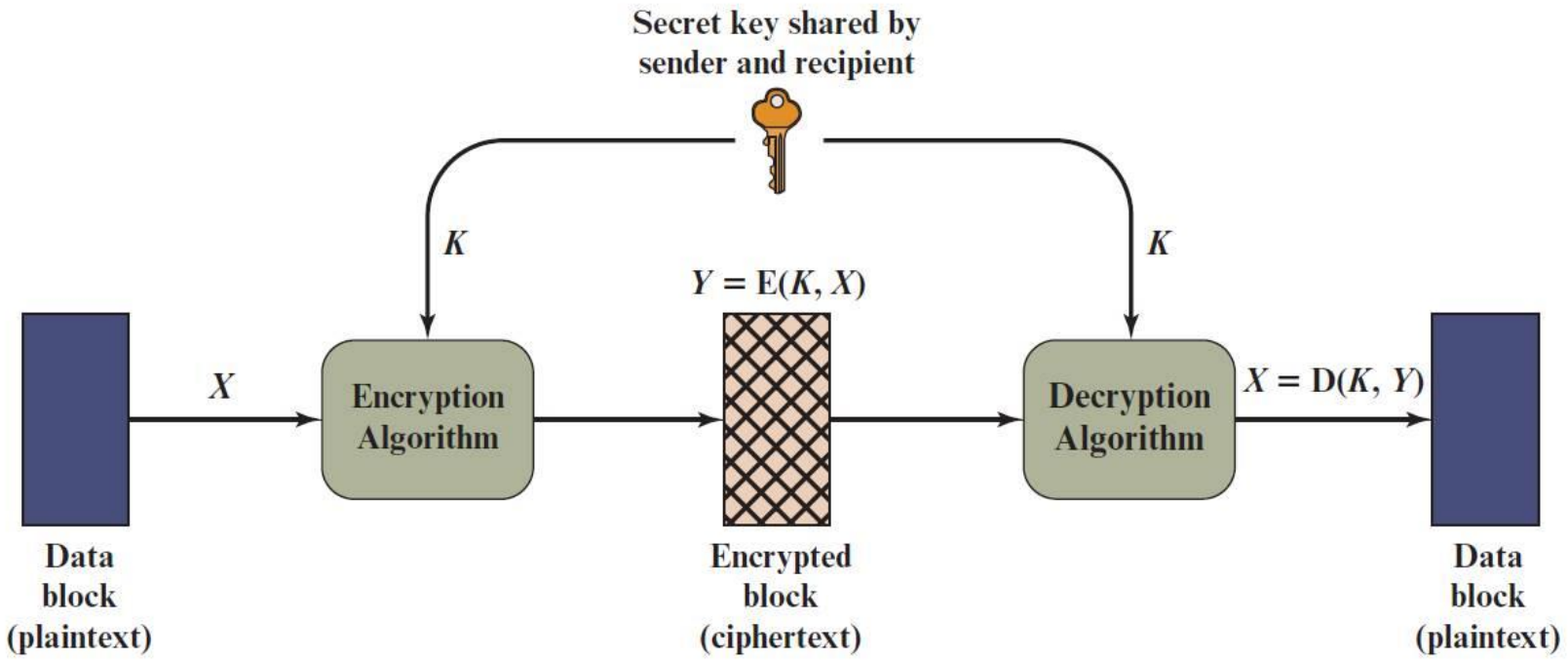
Definitions (1 of 2)

- Plaintext
  - An original message
- Ciphertext
  - The coded message
- Enciphering/encryption
  - The process of converting from plaintext to ciphertext
- Deciphering/decryption
  - Restoring the plaintext from the ciphertext

Definitions (2 of 2)

- Cryptography
  - The area of study of the many schemes used for encryption
- Cryptographic system/cipher
  - A scheme
- Cryptanalysis
  - Techniques used for deciphering a message without any knowledge of the enciphering details
- Cryptology
  - The areas of cryptography and cryptanalysis

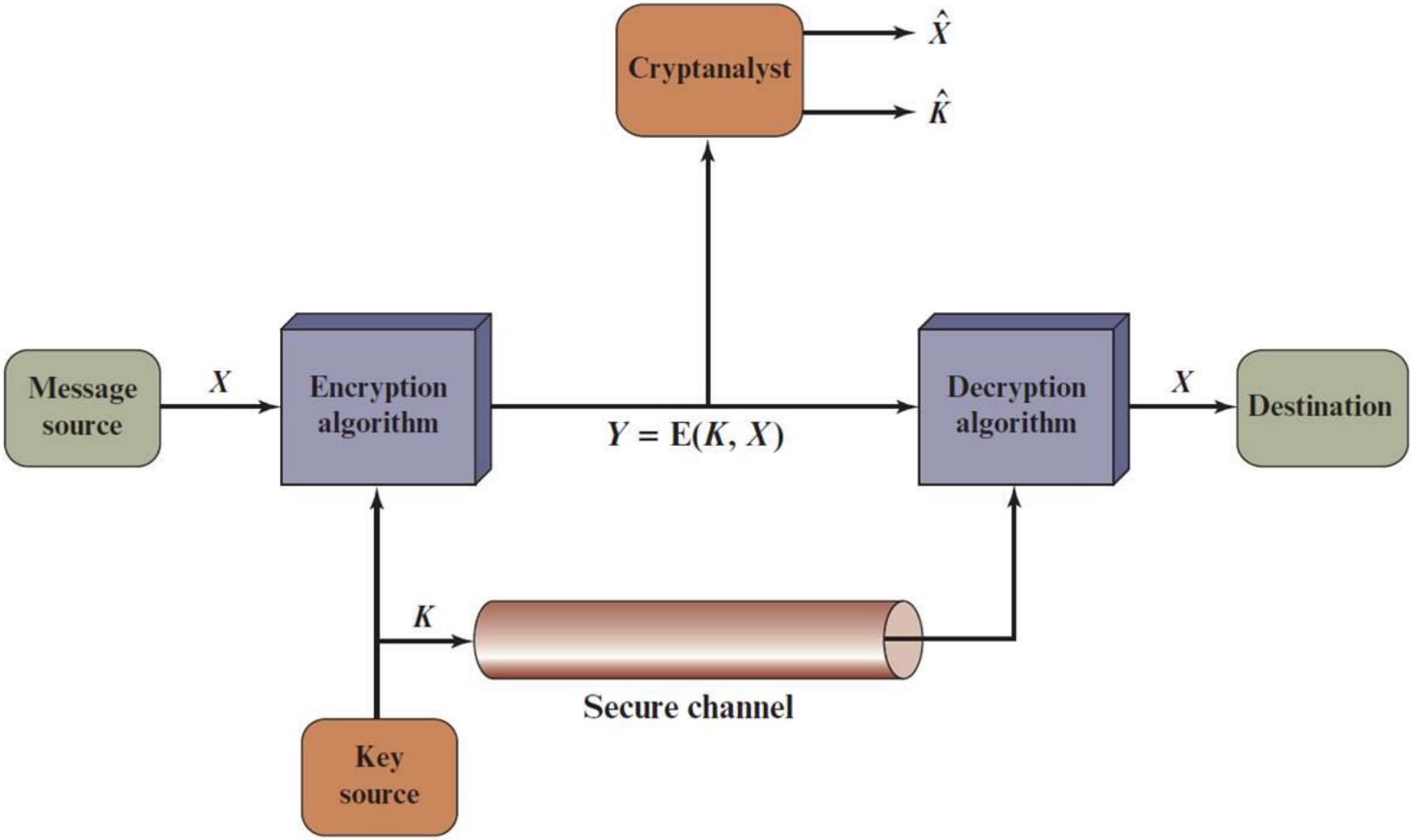
Figure 3.1 Simplified Model of Symmetric Encryption



Symmetric Cipher Model

- There are two requirements for secure use of conventional encryption:
  - A strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

Figure 3.2 Model of Symmetric Cryptosystem



Cryptographic Systems

- Characterized along three independent dimensions:
- The type of operations used for transforming plaintext to ciphertext
  - Substitution
  - Transposition
- The number of keys used
  - Symmetric, single-key, secret-key, conventional encryption
  - Asymmetric, two-key, or public-key encryption
- The way in which the plaintext is processed
  - Block cipher
  - Stream cipher

Cryptanalysis and Brute-Force Attack

- Cryptanalysis
  - Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext
  - Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used
- Brute-force attack
  - Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
  - On average, half of all possible keys must be tried to achieve success

Encryption Scheme Security

- Unconditionally secure
  - No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there
- Computationally secure apply two rules of cost and time
  - 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

Brute-Force Attack

- Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success
- To supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed

Strong Encryption

- The term *strong encryption* refers to encryption schemes that make it impractically difficult for unauthorized persons or systems to gain access to plaintext that has been encrypted
- Properties that make an encryption algorithm strong are:
  - Appropriate choice of cryptographic algorithm
  - Use of sufficiently long key lengths
  - Appropriate choice of protocols
  - A well-engineered implementation
  - Absence of deliberately introduced hidden flaws

Substitution Technique

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

Caesar Cipher

- Simplest and earliest known use of a substitution cipher
- Used by Julius Caesar
- Involves replacing each letter of the alphabet with the letter standing three places further down the alphabet
- Alphabet is wrapped around so that the letter following Z is A

plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

Caesar Cipher Algorithm

- 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
- Mathematically give each letter a number

  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
- Algorithm can be expressed as:

  $$c = E(3, p) = (p + 3) \bmod (26)$$
- A shift may be of any amount, so that the general Caesar algorithm is:

  $$C = E(k, p) = (p + k) \bmod 26$$
- Where k takes on a value in the range 1 to 25; the decryption algorithm is simply:

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

Caeser Cipher

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

P(Plain text) = HELLO WORLD

K = 3 ; formula for ciphertent (C)

Encryption:-

$C = (P+K) \bmod 26$

Let's begin with H

$C_H = (7+3) \bmod 26 \Rightarrow 10 \bmod 26$

↑ weight value from above

$C_H \Rightarrow 10$ [we did't perform modulas operation as remainder would be in decimal point]

$C_H = K$

Similarely P⇒ HELLO WORLD

↓ ↓ ↓ ↓ ↓ ↓

C ⇒ K HOOR

Decryption

$P = (C-K) \bmod 26$

$P_K = (10-3) \bmod 26 \Rightarrow 7 \bmod 26 \Rightarrow 7$

$P_K = H$

Let's take some more examples

Plaintext (P) = XYZ

$C_{XY} = (24+3) \bmod 26$

$= 27 \bmod 26$

$= 1 \Rightarrow B$

$C_X = (23+3) \bmod 26$

$= 26 \bmod 26 \Rightarrow 0 \Rightarrow A$

$C_Z = (25+3) \bmod 26 \Rightarrow 28 \bmod 26$

$\Rightarrow 2 \Rightarrow C$

$C_{(XYZ)} \Rightarrow ABC$

Now let's calculate plaintext (Decry

$P_A = (C-K) \bmod 26 \Rightarrow (0-3) \bmod 26$

$\Rightarrow -3 \bmod 26$ [In mod negative ⊗ not allowed]

$\Rightarrow (26-3) \bmod 26 \Rightarrow 23 \bmod 26$

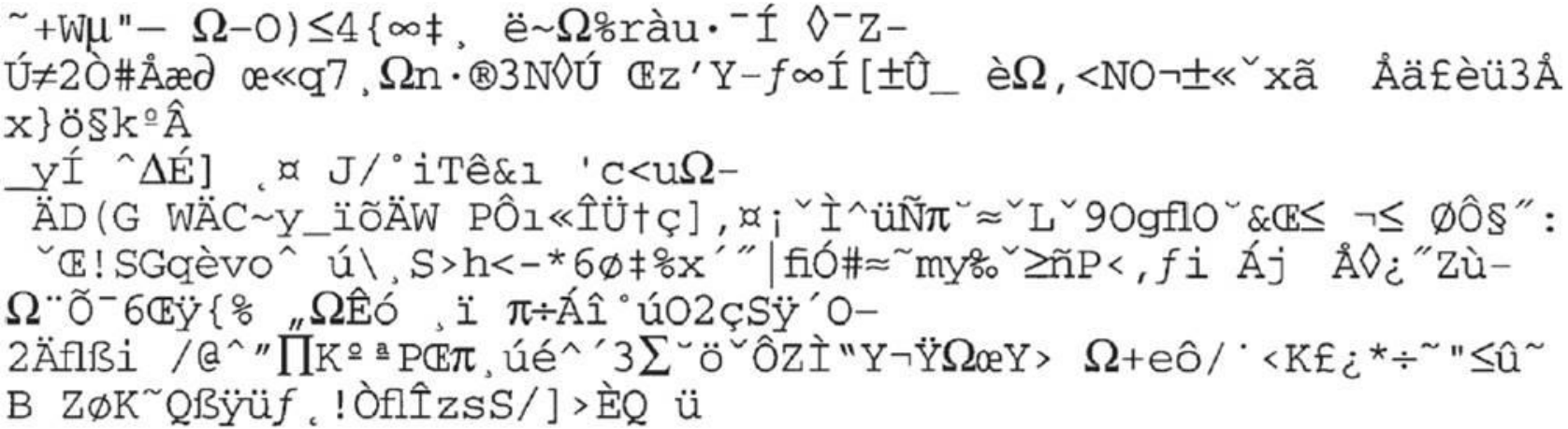
$\Rightarrow 23 \Rightarrow X$

Figure 3.3 Brute-Force Cryptanalysis of Caesar Cipher

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

Sample of Compressed Text

Figure 3.4 Sample of Compressed Text

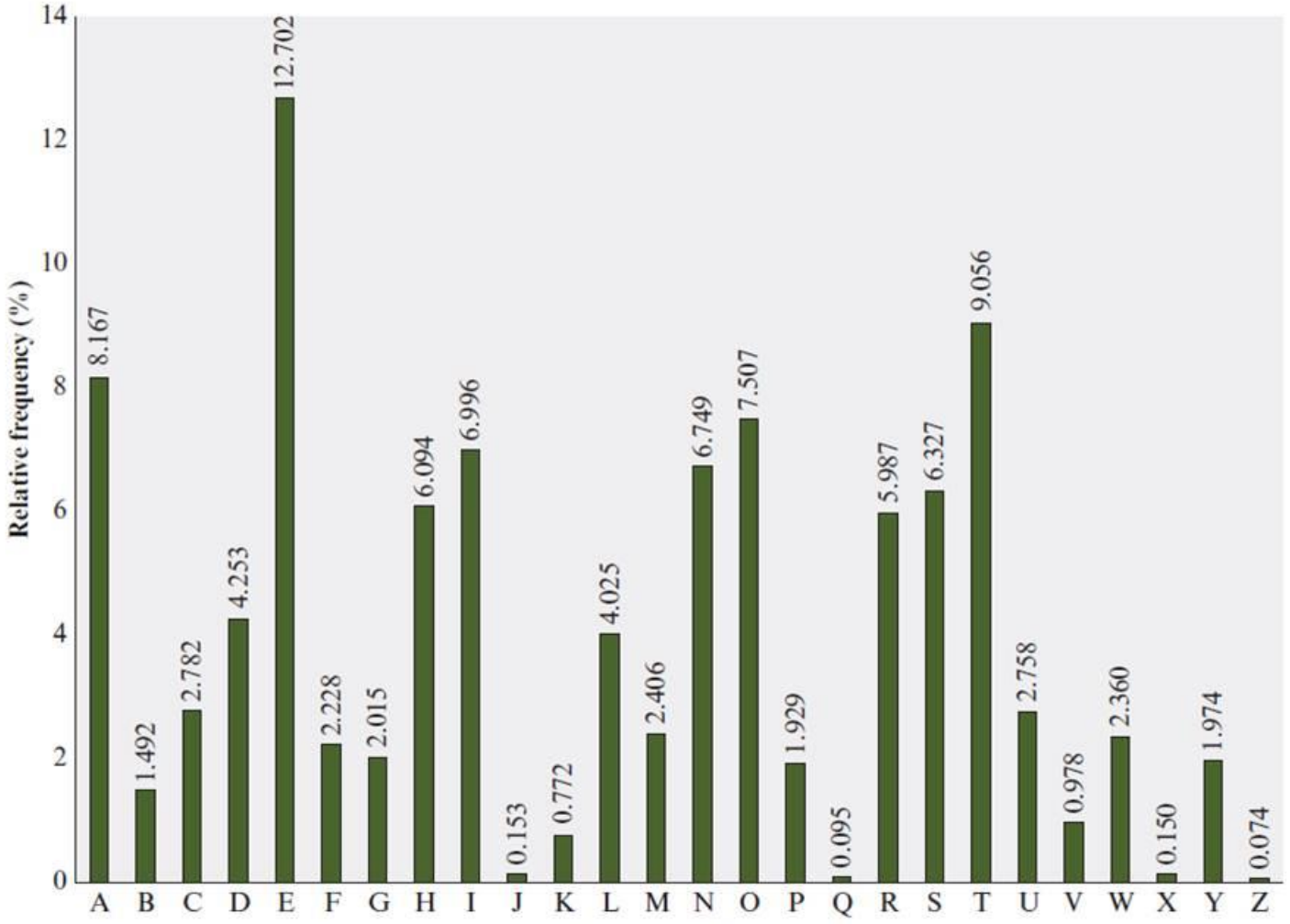


Monoalphabetic Cipher

- Rather than just shifting the alphabet could shuffle (jumble) the letters arbitrarily
-
- Each plaintext letter maps to a different random cipher text letter hence key is 26 letters long (26! (permutation combination)

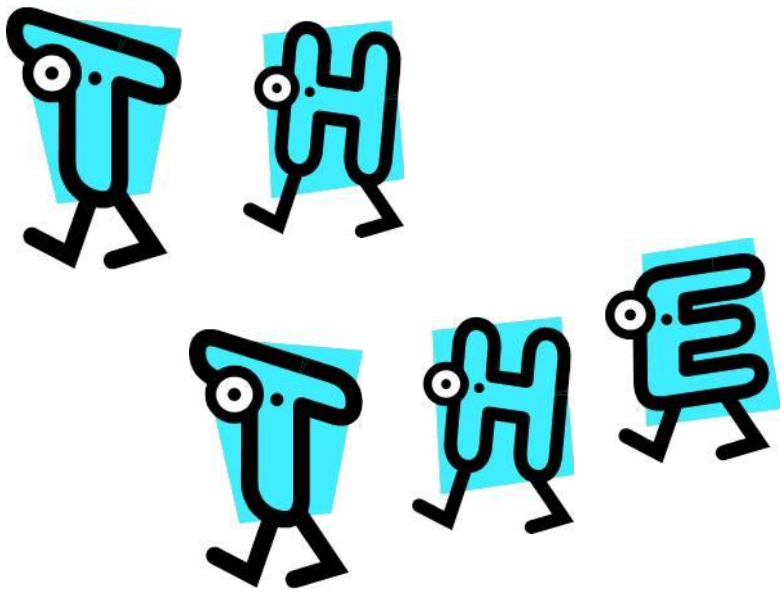
Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext: ifwewishtoreplaceletters
Cipher text: WIRFRWAJUHYFTSDVFSFUUFYA

Figure 3.5 Relative Frequency of Letters in English Text



Monoalphabetic Ciphers

- Easy to break because they reflect the frequency data of the original alphabet
- Countermeasure is to provide multiple substitutes (homophones) for a single letter
- Digram
  - Two-letter combination
  - Most common is *th*
- Trigram
  - Three-letter combination
  - Most frequent is *the*



Playfair Cipher

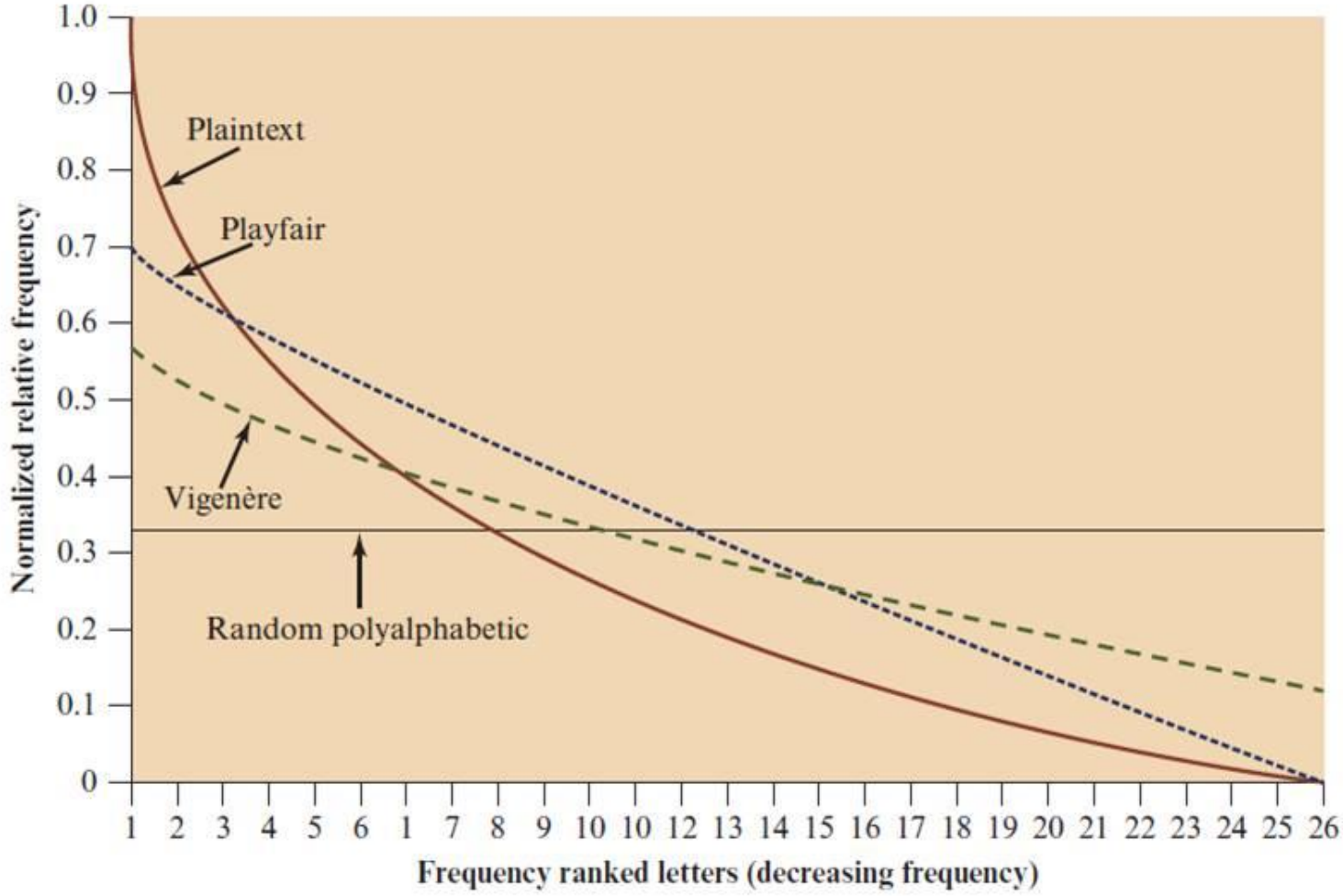
- Best-known multiple-letter encryption cipher
- Treats digrams in the plaintext as single units and translates these units into ciphertext digrams
- Based on the use of a 5 × 5 matrix of letters constructed using a keyword
- Invented by British scientist Sir Charles Wheatstone in 1854
- Used as the standard field system by the British Army in World War I and the U.S. Army and other Allied forces during World War II

Playfair Key Matrix

- Fill in letters of keyword (minus duplicates) from left to right and from top to bottom, then fill in the remainder of the matrix with the remaining letters in alphabetic order
- 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

Figure 3.6 Relative Frequency of Occurrence of Letters



Polyalphabetic Cipher

- Instead of having one key (table) that is used to encrypt each block of plaintext, we use several different keys.

- The Vigenère cipher is the classical example.

Vigenère Cipher

- Best known and one of the simplest polyalphabetic substitution ciphers
- In this scheme the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25
- Each cipher is denoted by a key letter which is the ciphertext letter that substitutes for the plaintext letter a

Vigenere Cipher

Step 1: Make a table with alphabets in the very first row and column

	A	B	C	D	E	F	...	Z
A	A	B	C	D	E	F		
B	B	C	D	E	F	G		
C	C	D	E	F	G	H		
D	D	E	F	G	H	I		
E	E	F	G	H	I	J		
F	F	G	H	I	J	H		
⋮								
Z								

Step 2: Follow RHS row to fill the table.

Example:

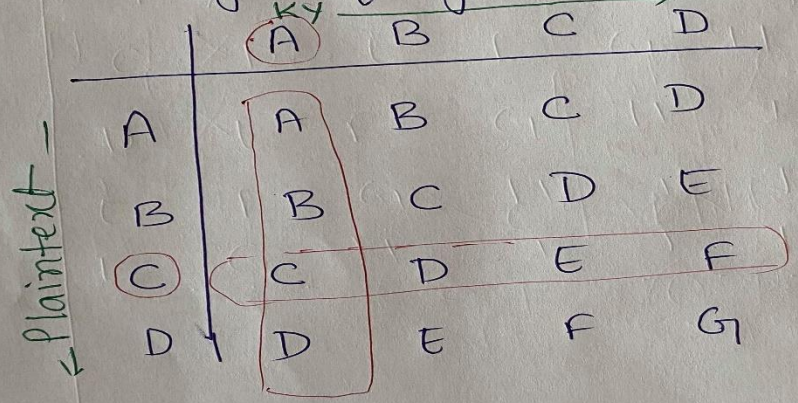
P = CAD

K = ADD

C = ?

Encryption

Step I: Make a table within the range of given alphabets (A to D)



Step II: Find the plaintext row and fix it. followed fixing the column of corresponding key alphabet.

Step III: Intersaction of row (from plaintext) with column (from key)

C = C D G

Now from ciphertext (C), we have to find plaintext

C = CDG

K = ADD

P = ?

Decryption

Step I: Draw the table, now (vertical) extreme column would be key

	A	B	C	D
A	A	B	C	D
B	B	C	D	E
C	C	D	E	F
D	D	E	F	G

Key ↓

Step II: First letter of ciphertext is C for this we need to find P. Now the corresponding key is A. The row in the key A is marked (fixed).

Step III: Now we have to find the corresponding ciphertext C in the marked row. The letter in the top most row of ciphertext alphabet is C

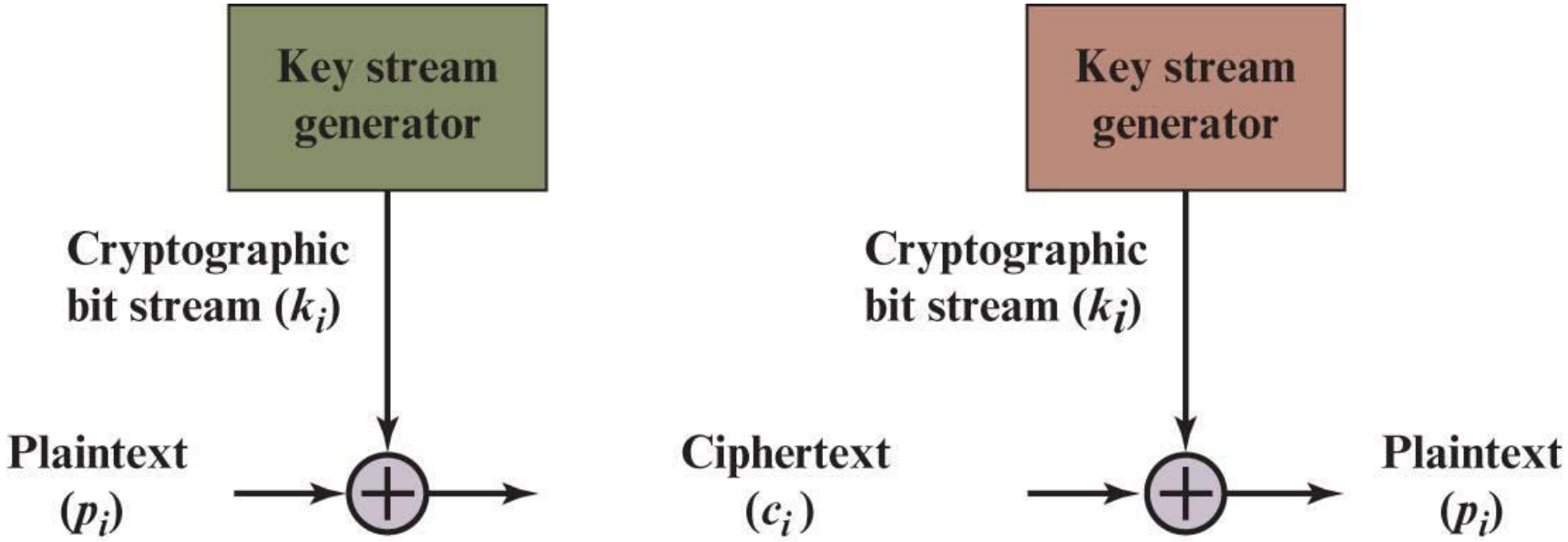
So:- P = CAD

One-Time Pads (Vernam Cipher)

- Extended from Vigenere cipher
- This is one type of substitution cipher that is absolutely unbreakable.
- The message is represented as a binary string (a sequence of 0's and 1's using a coding mechanism such as ASCII coding.
- The key is a truly random sequence of 0's and 1's of the same length as the message.
- The encryption is done by adding the key to the message modulo 2, bit by bit. This process is often called *exclusive or*, and is denoted by *XOR.* The symbol $\oplus$ is used
- Length of the key should be equal to plaintext

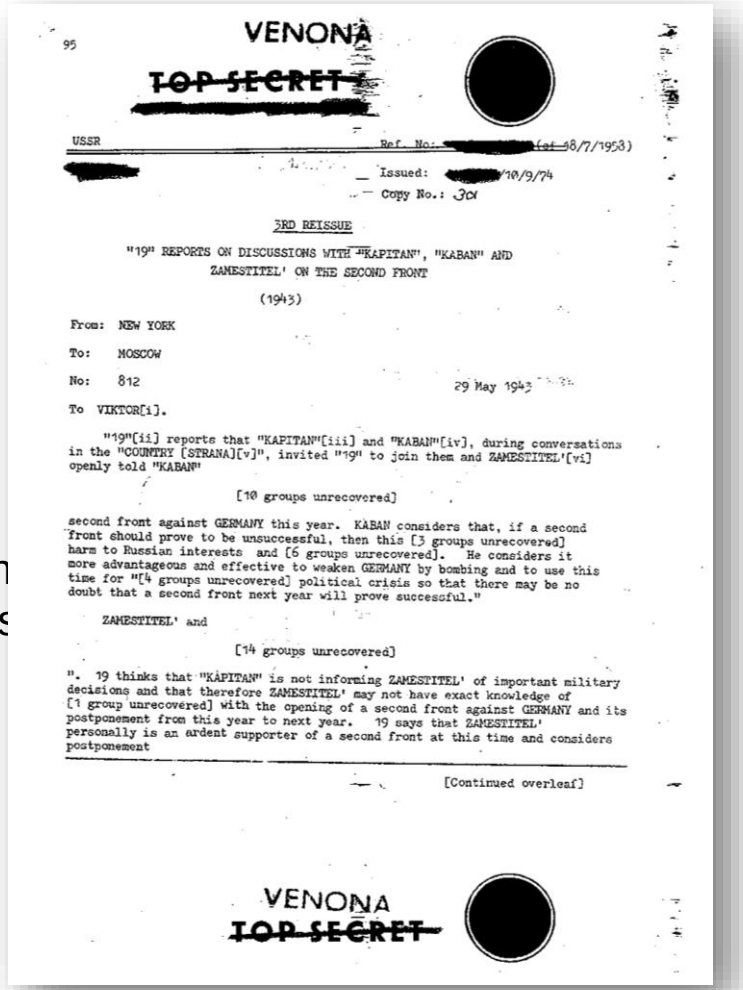
Vernam Cipher

Figure 3.7 Vernam Cipher



Weaknesses of the One-Time Pad (1943)

- In spite of their perfect security, one-time pads have some weaknesses
- The key has to be as long as the plaintext
- Keys can never be reused
  - Repeated use of one-time pads allowed th U.S. to break some of the communications of Soviet spies during the Cold War.



95

VENONA

~~TOP SECRET~~

USSR

Ref. No.: [redacted] (of 8/7/1958)

[redacted] Issued: [redacted] 10/9/74

Copy No.: 301

3RD REISSUE

"19" REPORTS ON DISCUSSIONS WITH "KAPITAN", "KABAN" AND ZAMESTITEL' ON THE SECOND FRONT

(1943)

From: NEW YORK

To: MOSCOW

No: 812

29 May 1943

To VIKTOR[i].

"19"[ii] reports that "KAPITAN"[iii] and "KABAN"[iv], during conversations in the "COUNTRY [STRANA][v]", invited "19" to join them and ZAMESTITEL'[vi] openly told "KABAN"

[10 groups unrecovered]

second front against GERMANY this year. KABAN considers that, if a second front should prove to be unsuccessful, then this [3 groups unrecovered] harm to Russian interests and [6 groups unrecovered]. He considers it more advantageous and effective to weaken GERMANY by bombing and to use this time for "[4 groups unrecovered] political crisis so that there may be no doubt that a second front next year will prove successful."

ZAMESTITEL' and

[14 groups unrecovered]

". 19 thinks that "KAPITAN" is not informing ZAMESTITEL' of important military decisions and that therefore ZAMESTITEL' may not have exact knowledge of [1 group unrecovered] with the opening of a second front against GERMANY and its postponement from this year to next year. 19 says that ZAMESTITEL' personally is an ardent supporter of a second front at this time and considers postponement

[Continued overleaf]

VENONA

~~TOP SECRET~~

Transposition ciphers

- An alternative to substitution ciphers
- Instead of changing the coding of the characters (blocks) in the plaintext, we rearrange the text.
- The effect is that the cipher text and the plaintext contains the same symbols.
- Algorithm

➢ Divide to plaintext into blocks

➢ Decide on a permutation order

➢ Rearrange the blocks according to this

Rail Fence Cipher

- Simplest transposition cipher
- Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
- To encipher the message “meet me after the toga party” with a rail fence of depth 2, we would write:

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

e t e f e t e o a a t

Encrypted message is:

MEMATRHTGPRYETEFETEOAAT

Summary: Problems with classical ciphers

- Neither substitution nor transposition ciphers are secure enough today
- They also often have problems with complex keys that are hard to remember
- Solution?
- Hybrid approach by combining both methods
- Combine both methods!
- Simple ciphers can be implemented in hardware
- S-box = substitution cipher
- P-box = transposition cipher

Chapter 4

Block Ciphers and the Data Encryption Standard

Stream Cipher (1 of 2)

- 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

Stream Cipher (2 of 2)

- 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

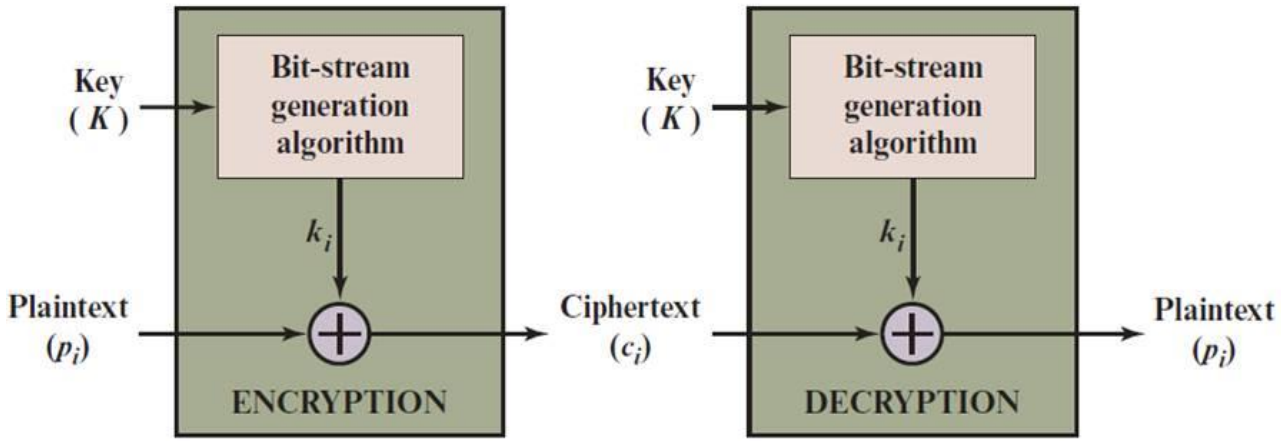
Stream and Block Ciphers

- Both uses symmetric encryption key
- Stream Cipher: It encrypts a digital data stream one bit or 1 byte at a time
- Block Ciphers: In this a block of plain text is treated as a whole and used to produce the ciphertext of equal length, Typically a block size of 64 or 128 bits.

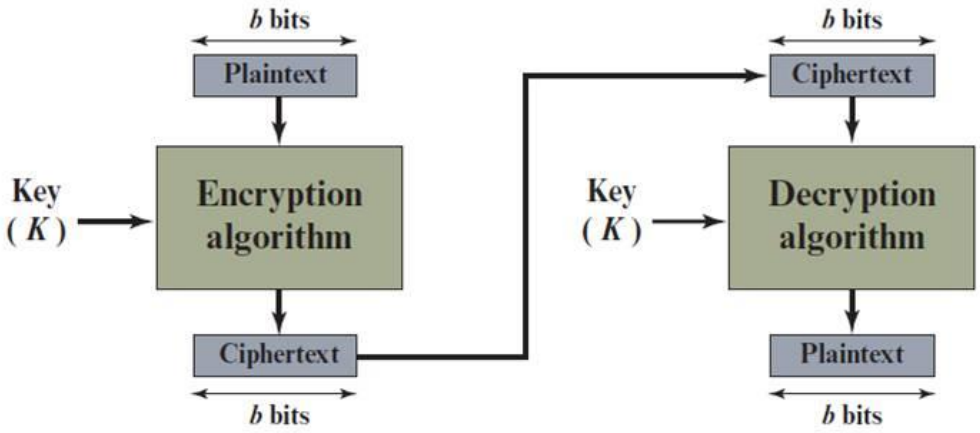
Block Cipher

- 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

Figure 4.1 Stream Cipher and Block Cipher



(a) Stream cipher using algorithmic bit-stream generator



(b) Block cipher

Feistel Cipher (1973 IBM)

- Feistel cipher refers to splitting the plaintext into two equal parts
- Split plaintext block into left and right halves: Plaintext = (L_0,R_0)
- These two halves of the data pass through n rounds of processing and then combine to produce the ciphertext block
- For each round i=1,2,...,n, compute

  $L_i = R_{i-1}$

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

  where F is round function and K_i is subkey (*refer next slide for it*)
- Ciphertext = (L_n,R_n)

Feistel Cipher

- Decryption: Ciphertext = (L_n, R_n)

- For each round $i = n, n-1, \ldots, 1$, compute

  $R_{i-1} = L_i$

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

  where F is round function and K_i is subkey

- Plaintext = (L_0, R_0)

- Formula "works" for any function F

- But only secure for certain functions F

Feistel Cipher: Sub Key

- On the right half we apply a function and in the function, we use a subkey generated from the master key (main key)
- A substitution is performed on the left half of the data. This is done by applying a round function to the right half of the data followed by the XOR of the output of that function and the left half of the data.
- That's count the first round.
- All rounds have the same structure
- All conventional block encryption algorithms including data encryption standard (DES) are based on Feistel Cipher Structure.

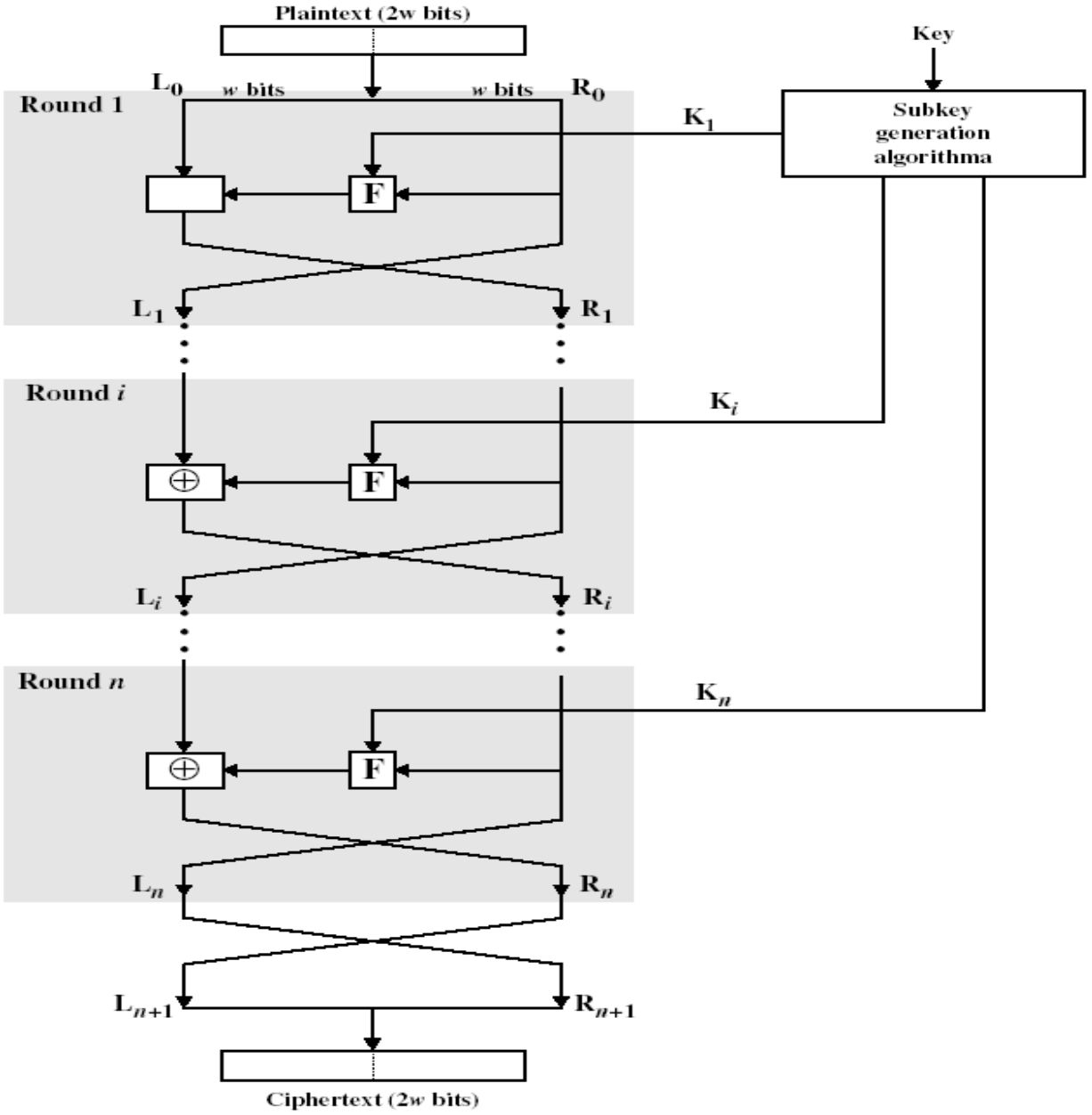
Block Cipher Design Principles: Design of Function F

- 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

Classical Feistel Network



Design Features of Feistel Network

- **Block Size:** (larger block means greater security).
- **Key Size**:56-128 bits.
- **Number of Rounds**: a single round offers inadequate security; a typical size is 16 rounds.
- **Sub-key Generation Algorithms**: greater complexity should lead to a greater difficulty of cryptanalysis.
- **Round function:** Again, greater complexity generally means greater resistance to cryptanalysis.

Block Ciphers in Practice

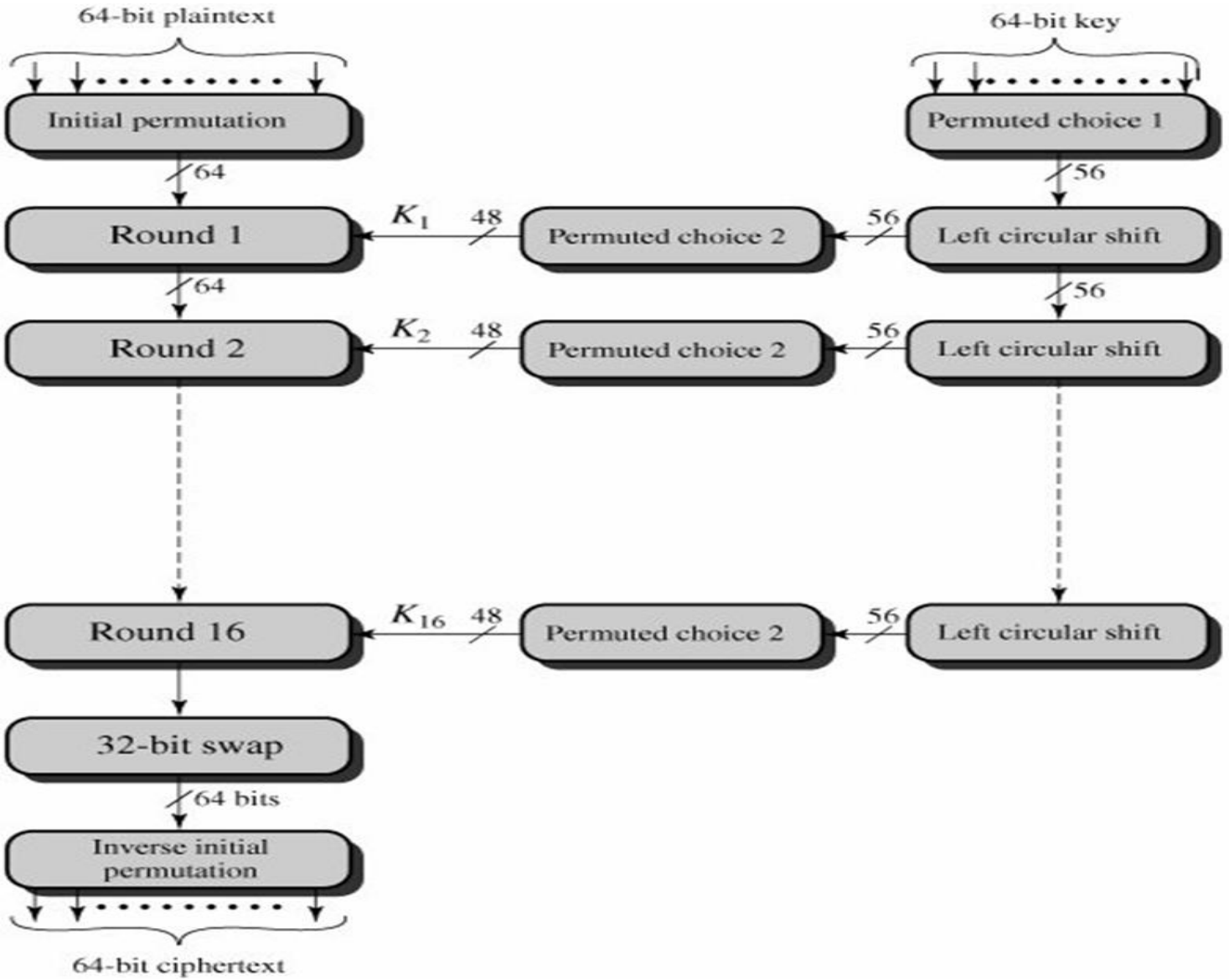
- Data Encryption Standard (DES)
  - Developed by IBM and adopted by NIST in 1977
  - 64-bit blocks and 56-bit keys
  - Small key space makes exhaustive search attack feasible since late 90s
- Triple DES (3DES)
  - Nested application of DES with three different keys KA, KB, and KC
  - Effective key length is 168 bits, making exhaustive search attacks unfeasible
  - $C = E_{KC}(D_{KB}(E_{KA}(P)))$; $P = D_{KA}(E_{KB}(D_{KC}(C)))$
  - Equivalent to DES when KA=KB=KC (backward compatible)
- Advanced Encryption Standard (AES)
  - Selected by NIST in 2001 through open international competition and public discussion
  - 128-bit blocks and several possible key lengths: 128, 192 and 256 bits
  - Exhaustive search attack not currently possible
  - AES-256 is the symmetric encryption algorithm of choice

Data Encryption Standard (DES)

- Block Cipher
- Symmetric cipher (same Keys for the encryption and decryption)
- 64 bit plaintext block
- 16 fiestel round

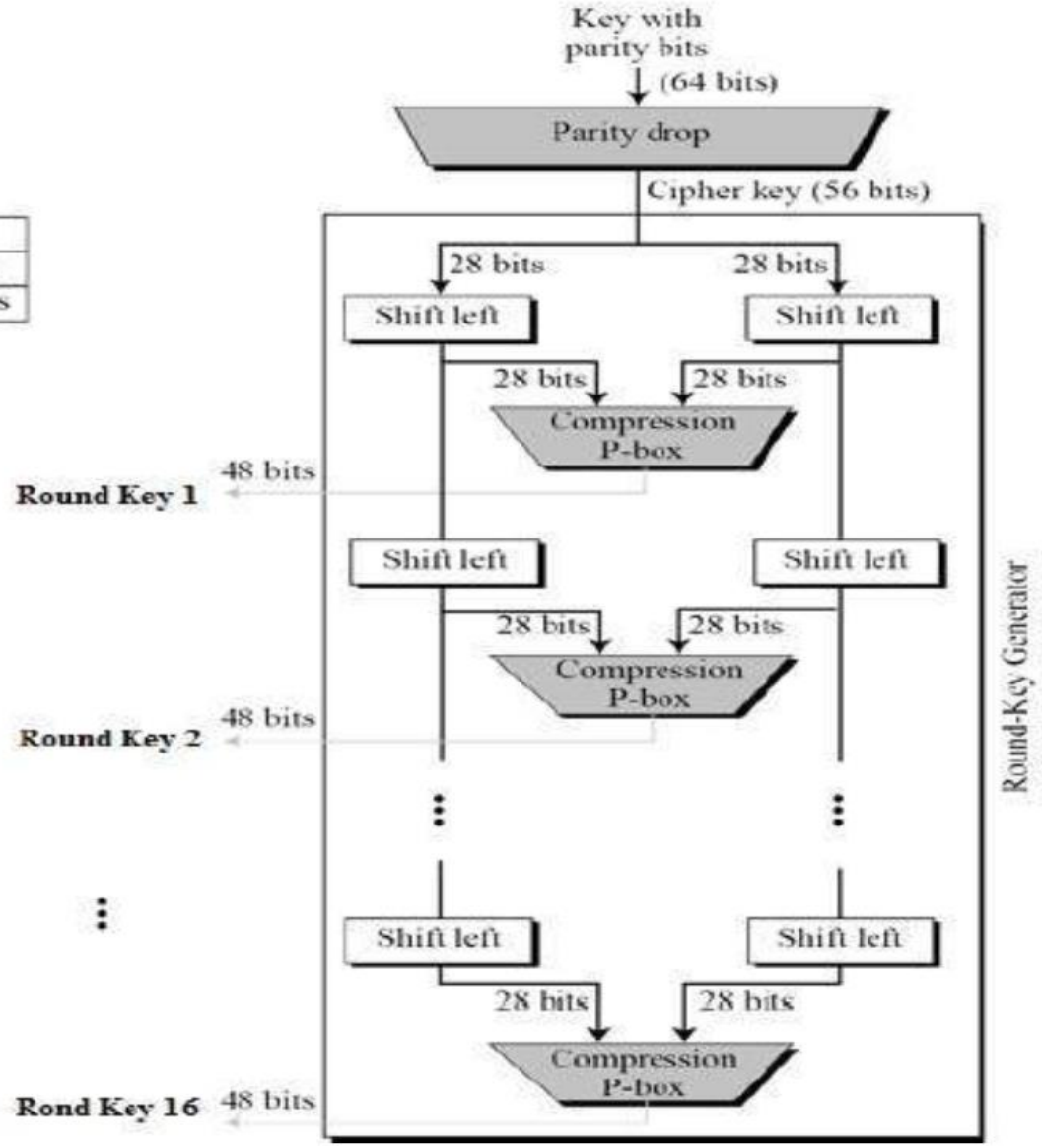
- Steps in DES:

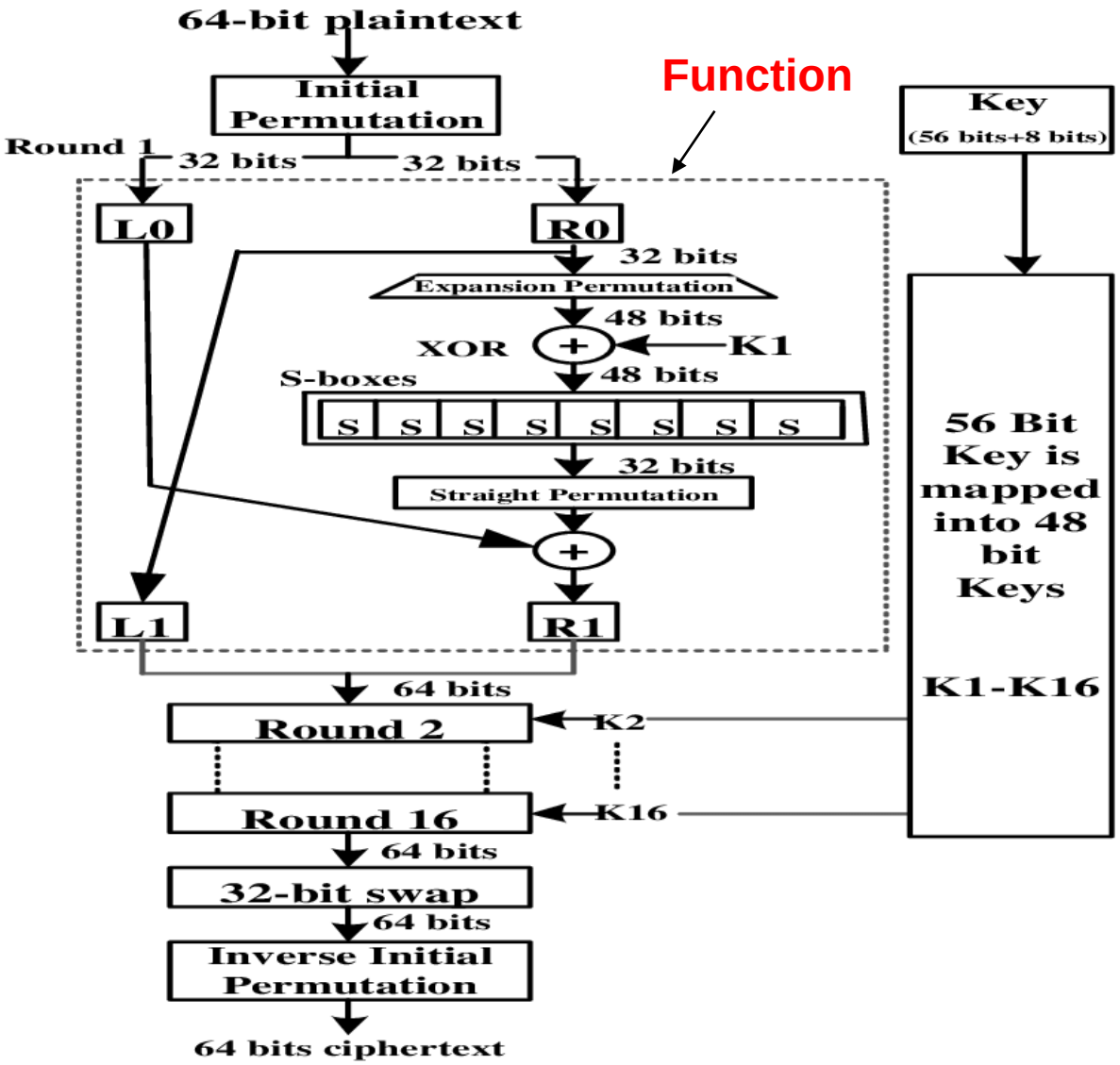
I. Initial Permutation
II. 16 Feistel rounds
III. Swapping or left right swap
IV. Final permutation or inverse initial permutation



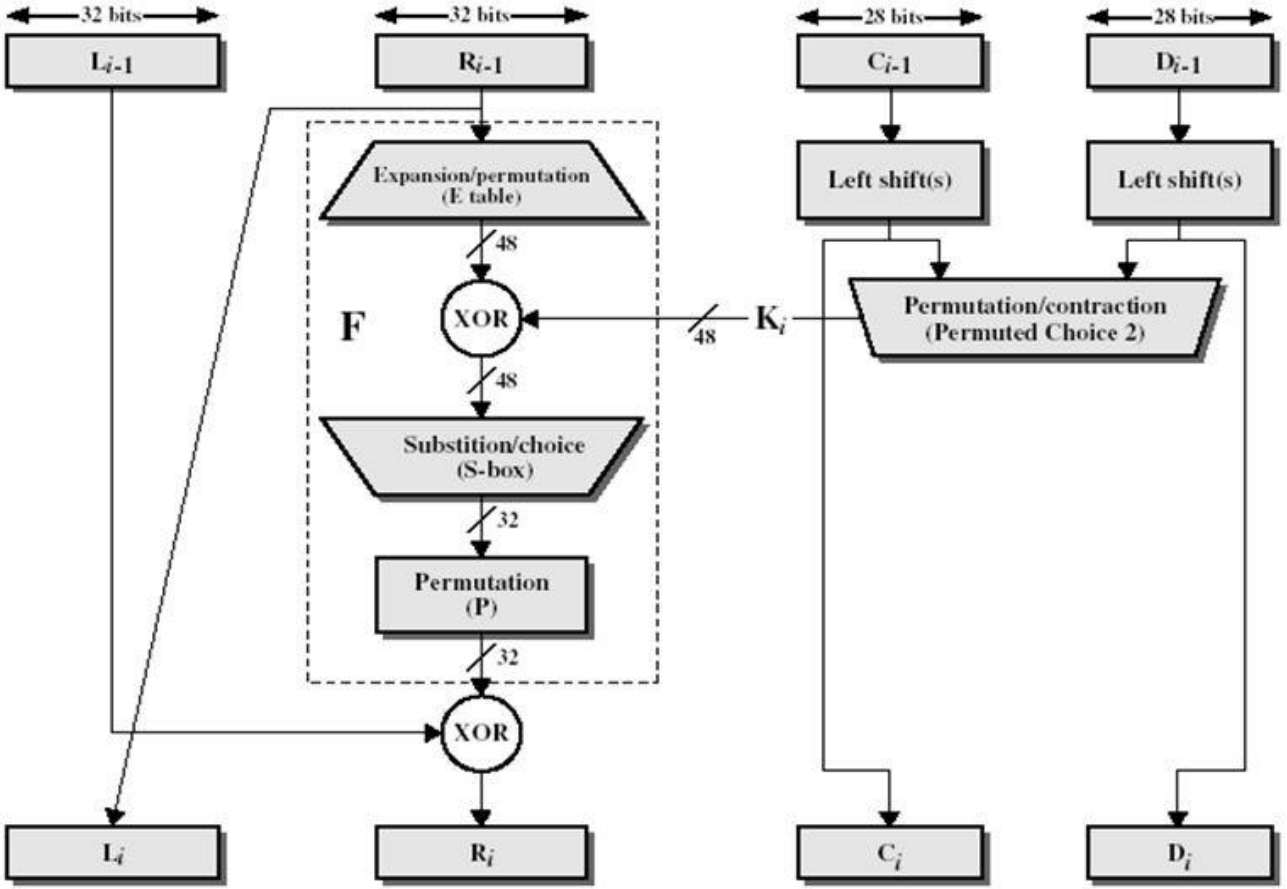
Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits





Single Iteration of DES Algorithm



DES Analysis

- The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.
- **Avalanche effect** − A small change in plaintext results in the very great change in the ciphertext.
- **Completeness** − Each bit of ciphertext depends on many bits of plaintext.

- Problem with DES

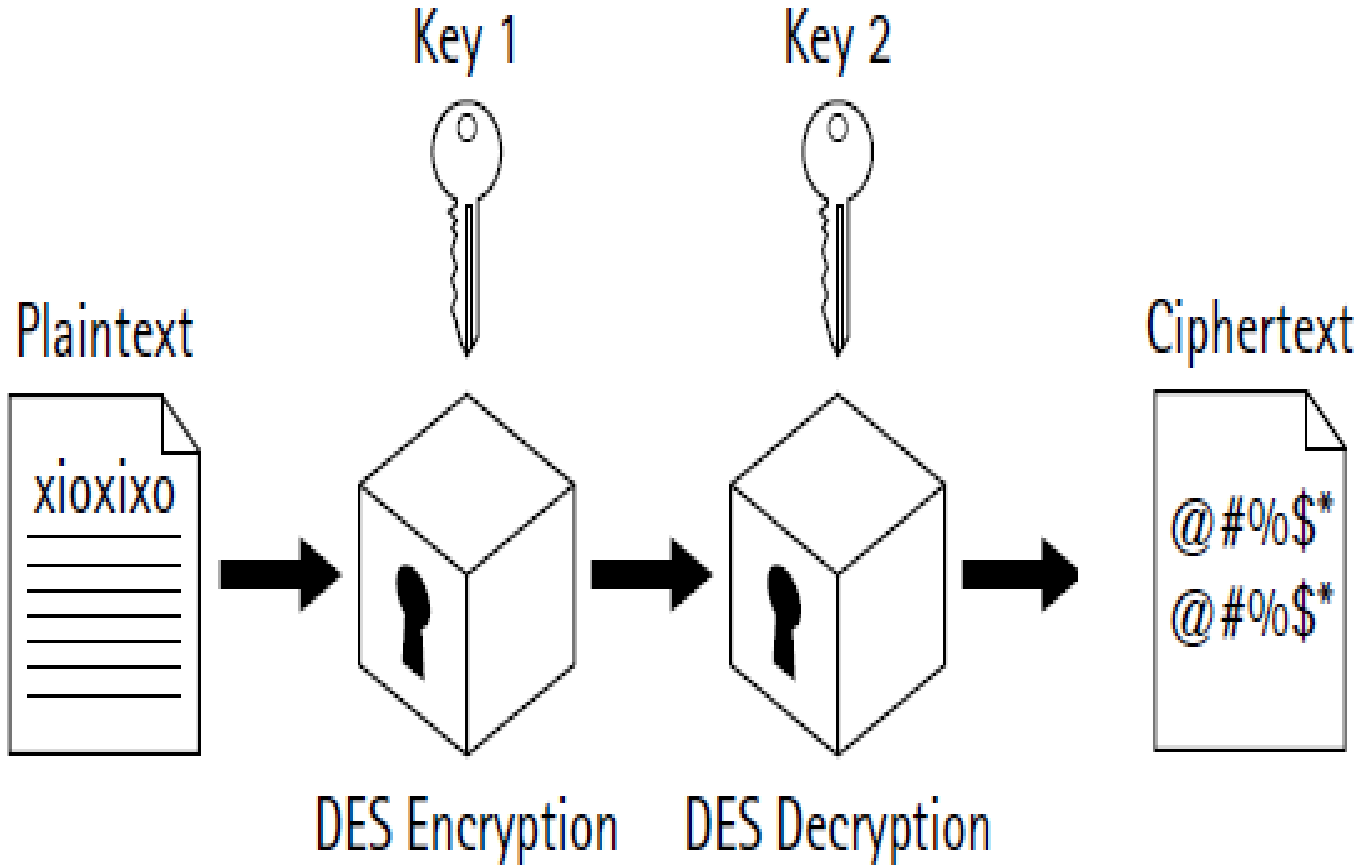
➢ Broken in 1998 by Electronic Frontier Foundation

➢ Used special purpose machine - $250,000 [a]Took less than three days

Double DES

- DES uses a 56-bit key, this raised concerns about brute force attacks.
- Double uses two keys, K1 and K2
- Perform DES on the plaintext using K1 to get encrypt text.
- Again, perform DES on the encrypt text using K2.
- The final output is the encryption of the encrypted text.
- his leads to a 2x56=112 bit key, so it is more secure than DES. Is it?
- $$p \rightarrow E(k_1, \quad p) \rightarrow E(k_2, \quad E(k_1, \quad p)) = C$$

Double DES



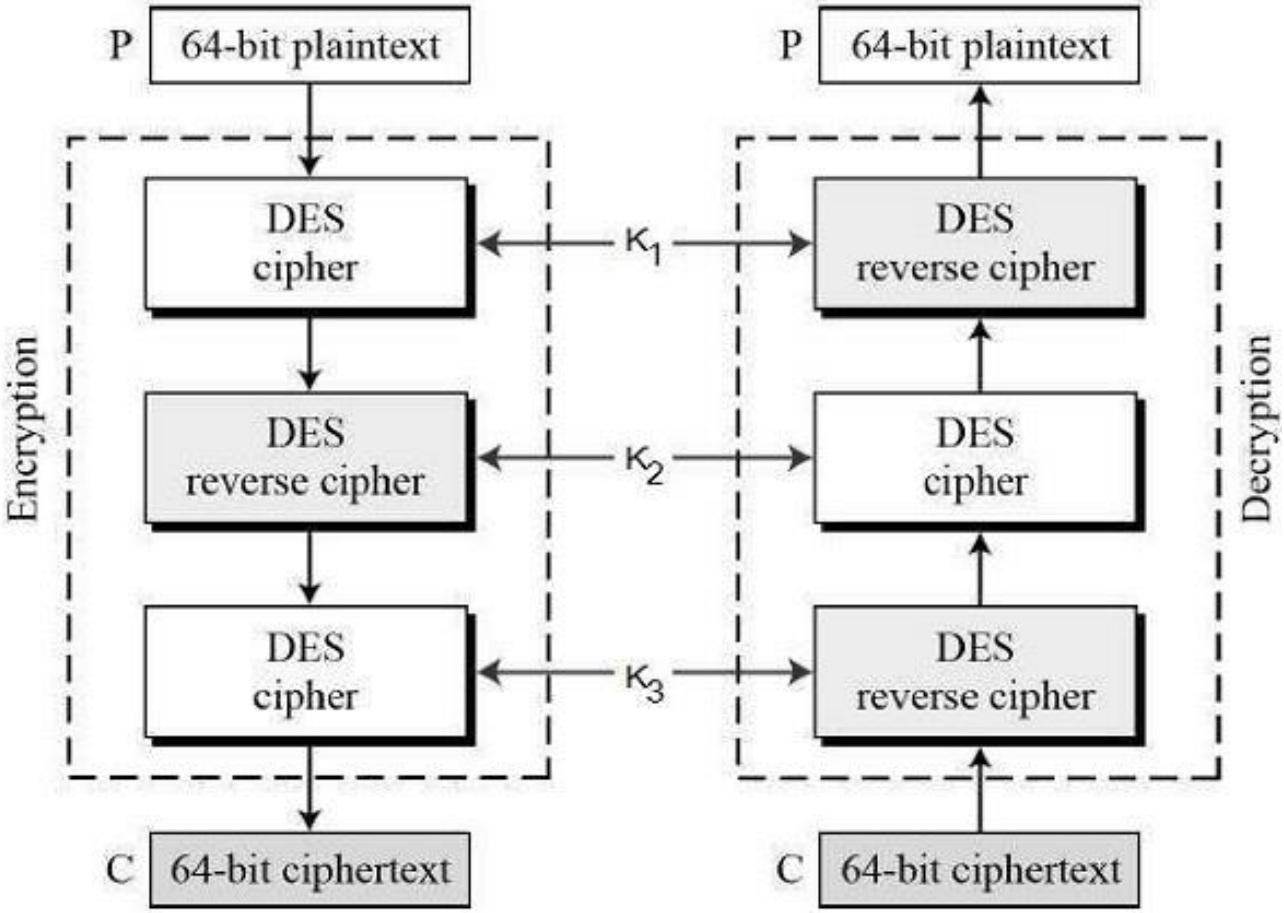
Meet-in-the-Middle Attack (MIM Attack)

- To improve the security of a block cipher, one might get the (naive) idea to simply use two independent keys to encrypt the data twice.
- In fact, an exhaustive search of all possible combinations of keys would take 2^{2n} attempts (if each key K1, K2 is n bits long), compared to the 2^n attempts required for searching a single key.
- The attacker can first compute E_{K1}(P) for all possible keys K1 and store the results in memory (in a lookup table).
- Afterwards he can decrypt the ciphertext by computing D_{K2}(C) for each K2.
- Any matches between these two resulting sets are likely to reveal the correct keys. (To speed up the comparison, the E_{K1}(P) set is stored in an in memory lookup table, then each D_{K2}(C) can be matched against the values in the lookup table to find the candidate keys.)
- Once the matches are discovered, they can be verified with a second testset of Plaintext and Ciphertext.

Triple DES

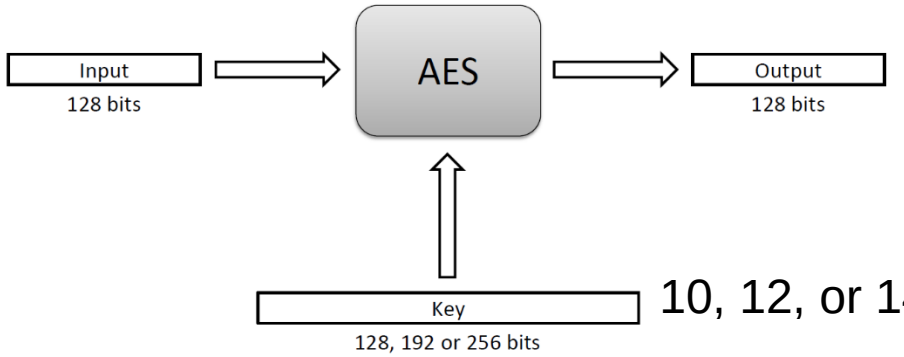
- 3DES was developed in 1999 by IBM – by a team led by Walter Tuchman. 3DES prevents a meet-in-the-middle attack. 3DES has a 168-bit key and ciphers blocks of 64 bits
- The plain text block is first encrypted with k1, then encrypted with k2 and finally with the k3
- All three keys would be different from each other
- Its three times slower than DES

Triple DES



The Advanced Encryption Standard (AES)

- clear a replacement for DES was needed
  - ➢ have theoretical attacks that can break it
  - ➢ have demonstrated exhaustive key search attacks
- can use Triple-DES – but slow, has small blocks
- US NIST issued call for ciphers in 1997
- 15 candidates accepted in Jun 98
- 5 were shortlist
- Rijndael was se



01

10, 12, or 14 round

The AES Cipher - Rijndael

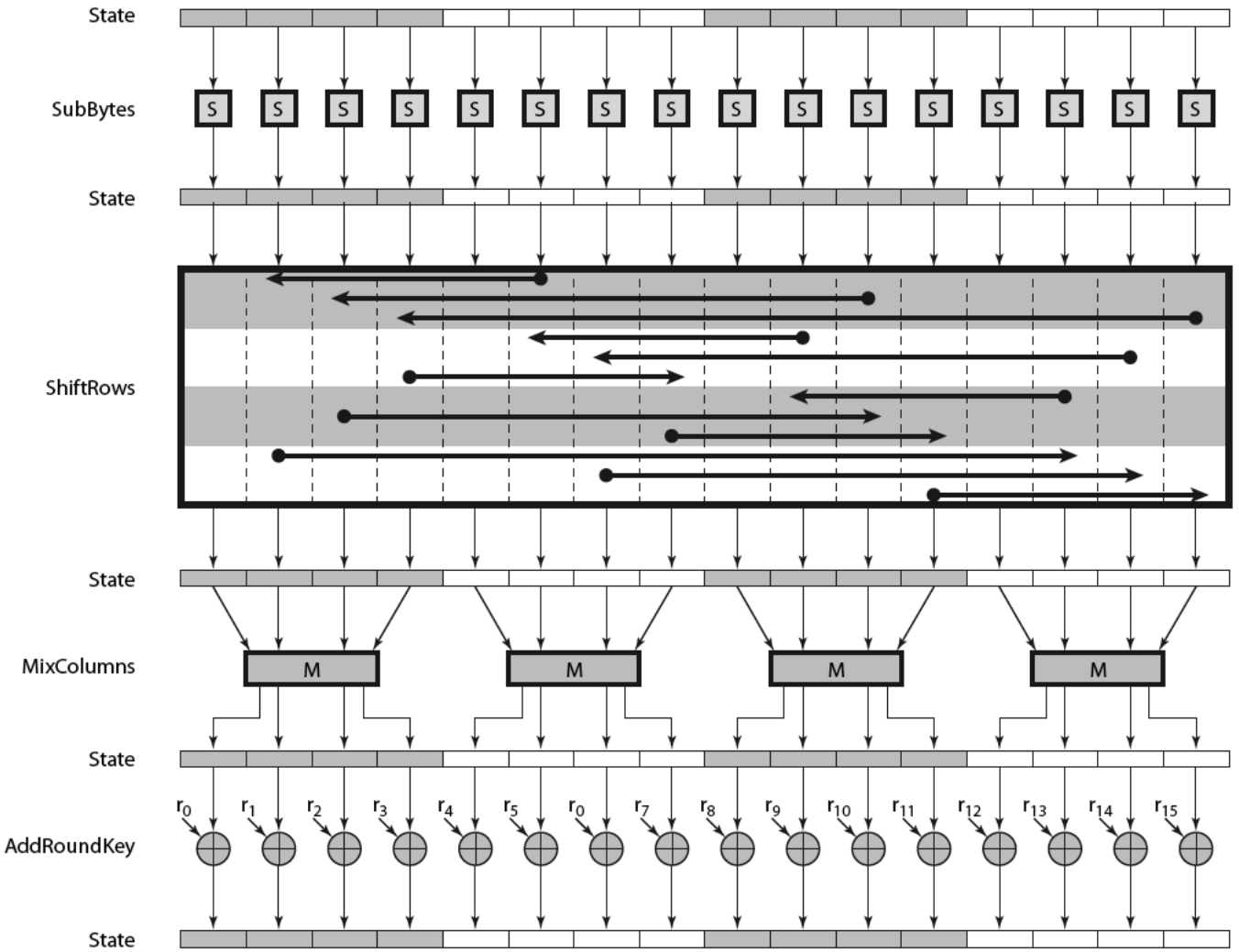
- has 128/192/256 bit keys, 128 bit data
- an **iterative** rather than **Feistel** cipher
  - processes data as block of 4 columns of each 4 bytes
  - operates on entire data block in every round
- designed to have:
  - resistance against known attacks
  - speed and code compactness on many CPUs
  - design simplicity

AES Rounds

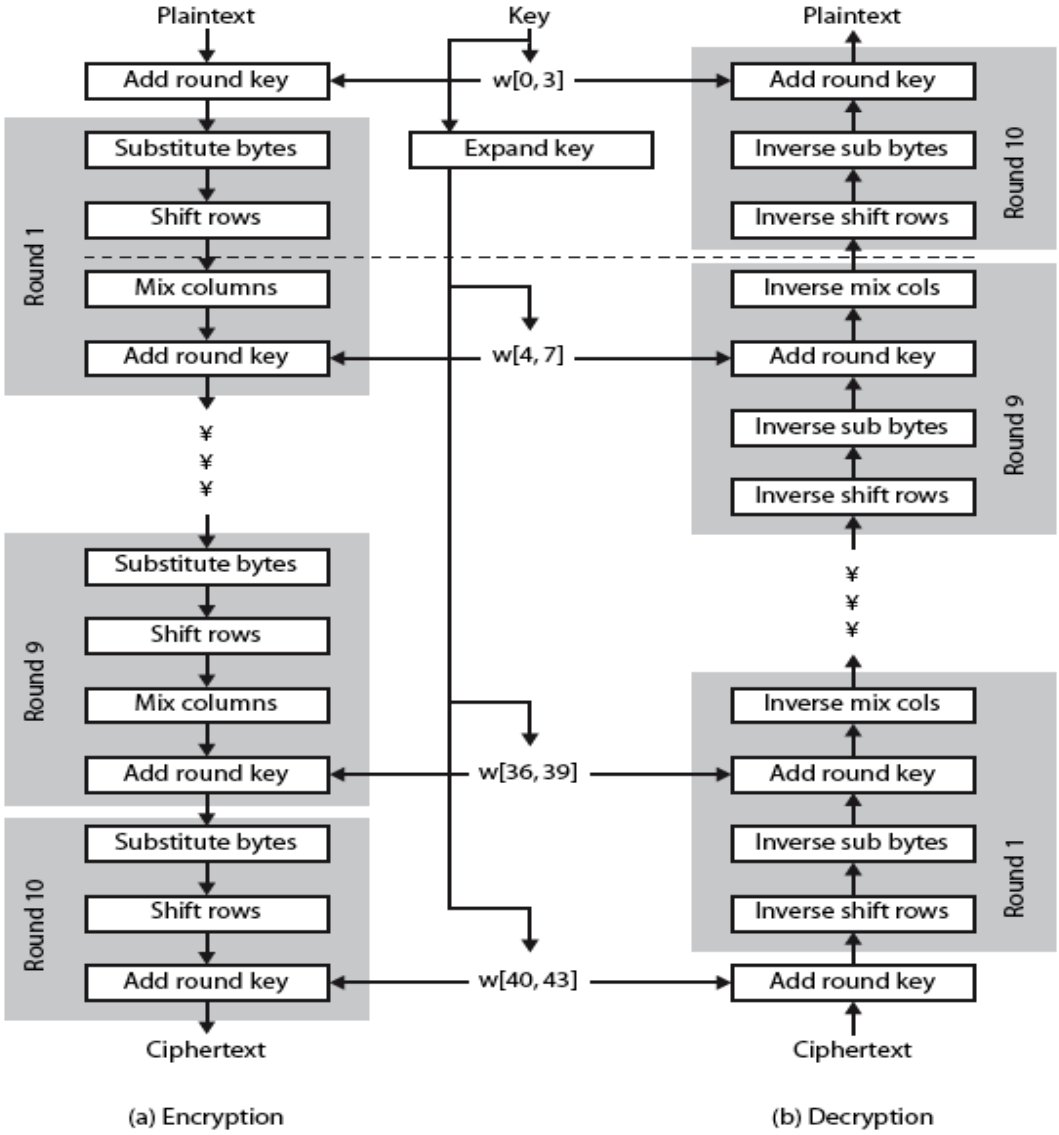
- Each round is built from four basic steps:

1. SubBytes step: an S-box substitution step
2. ShiftRows step: a permutation step
3. MixColumns step: a matrix multiplication step
4. AddRoundKey step: an XOR step with a round key derived from the 128/192/256-bit encryption key

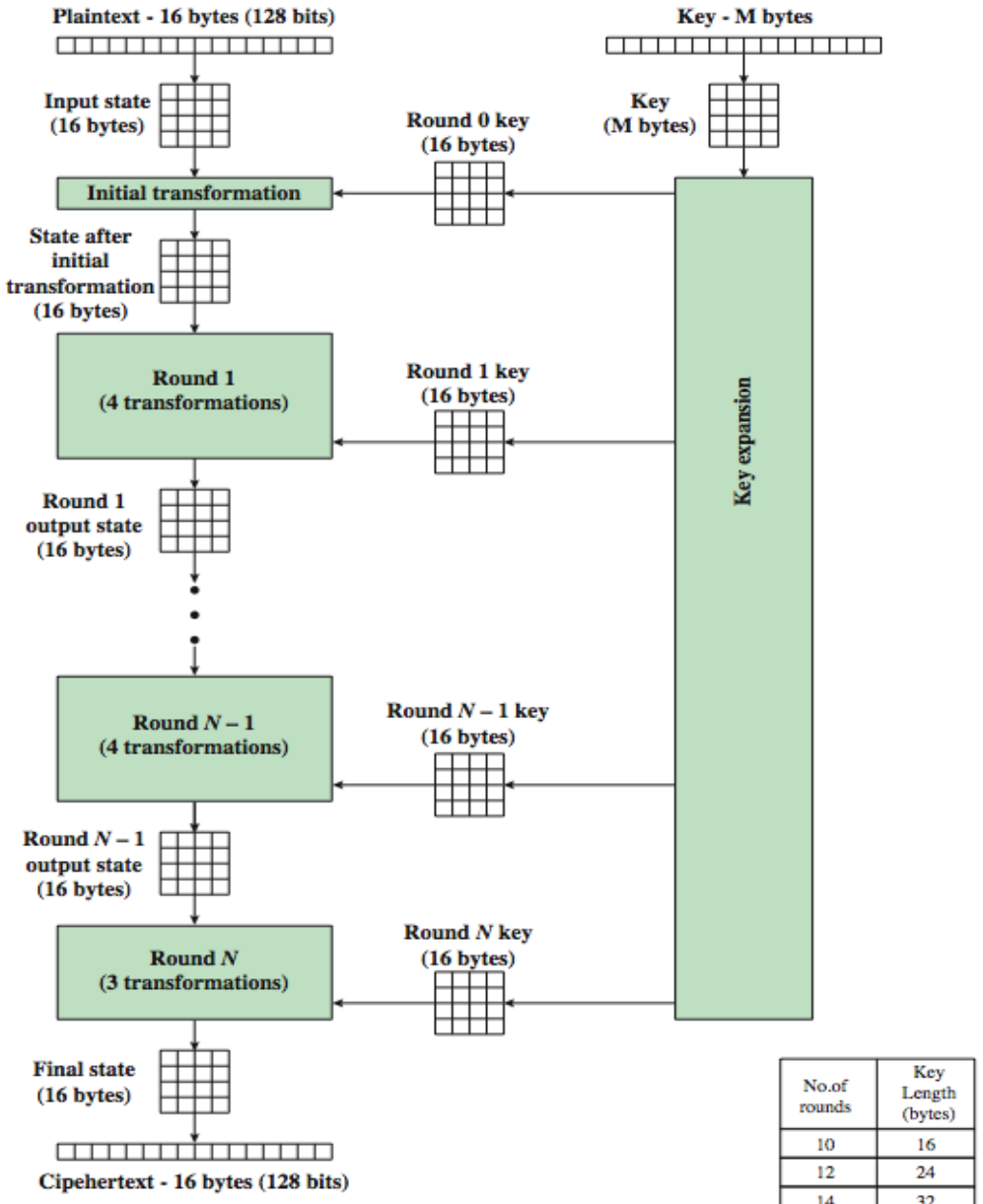
AES Round



AES Structure



AES Encryption Process



No.of rounds	Key Length (bytes)
10	16
12	24
14	32

AES Key Expansion

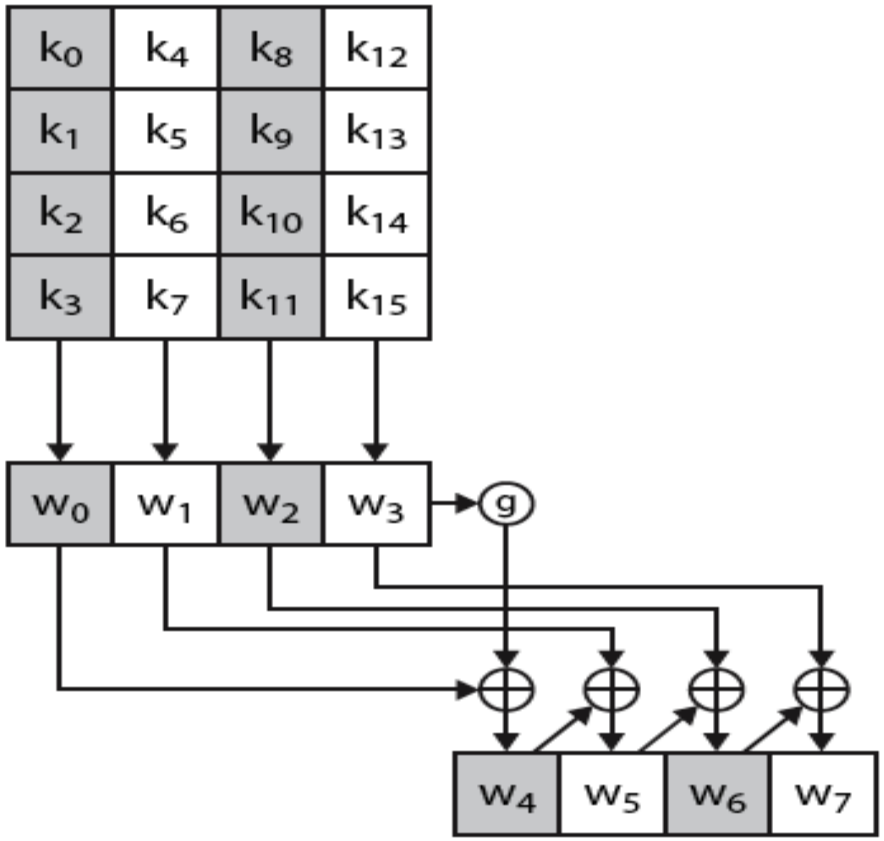


Table 4.5 Average Time Required for Exhaustive Key Search

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

How to use a block cipher?

- Block ciphers encrypt fixed-size blocks
  - e.g. DES encrypts 64-bit blocks
- We need some way to encrypt a message of arbitrary length
  - e.g. a message of 1000 bytes
- NIST defines several ways to do it
  - called **modes of operation**

Modes of Operation

- Block ciphers encrypt fixed size blocks
  - eg. DES encrypts 64-bit blocks, with 56-bit key
- Need way to use in practise, given usually have arbitrary amount of information to encrypt
  - Partition message into separate block for ciphering
-
- A mode of operation describes the process of encrypting each of these blocks under a single key
- Some modes may use randomized addition input value

Table 7.1 Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 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 block of plaintext and the preceding block of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• 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 Code Book (ECB)

- The plaintext is broken into blocks, P_1, P_2, P_3, ...
- Each block is encrypted independently:

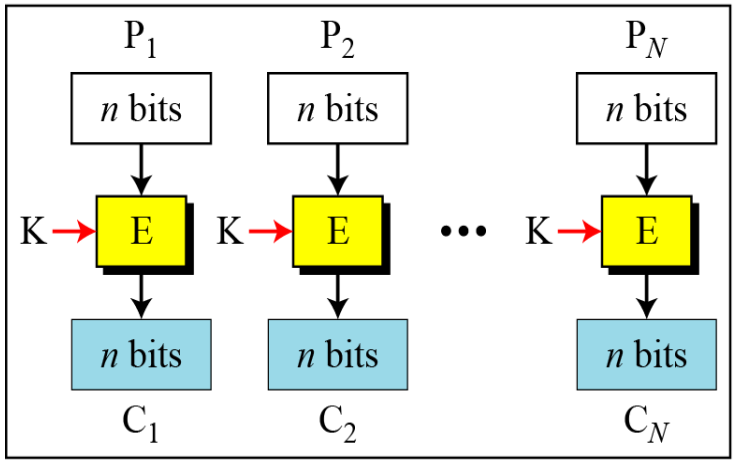
$$C_i = E_K(P_i)$$

- For a given key, this mode behaves like we have a gigantic codebook, in which each plaintext block has an entry, hence the name Electronic Code Book

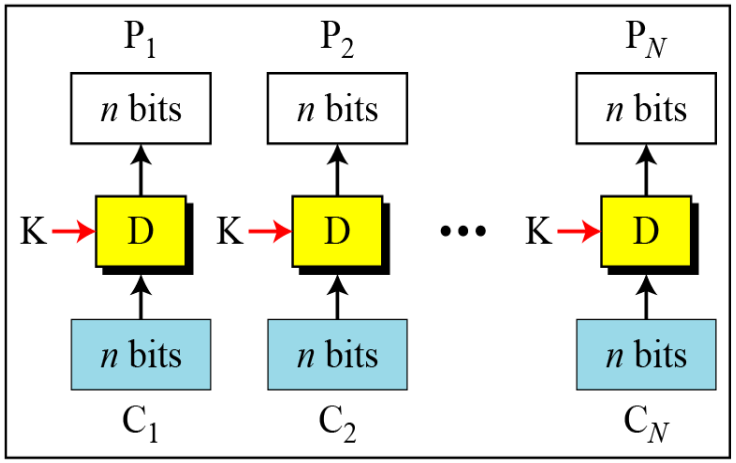
ECB Scheme

Encryption: $C_i = E_K (P_i)$ Decryption: $P_i = D_K (C_i)$

E: Encryption D: Decryption
P_i: Plaintext block i C_i: Ciphertext block i
K: Secret key



Encryption



Decryption

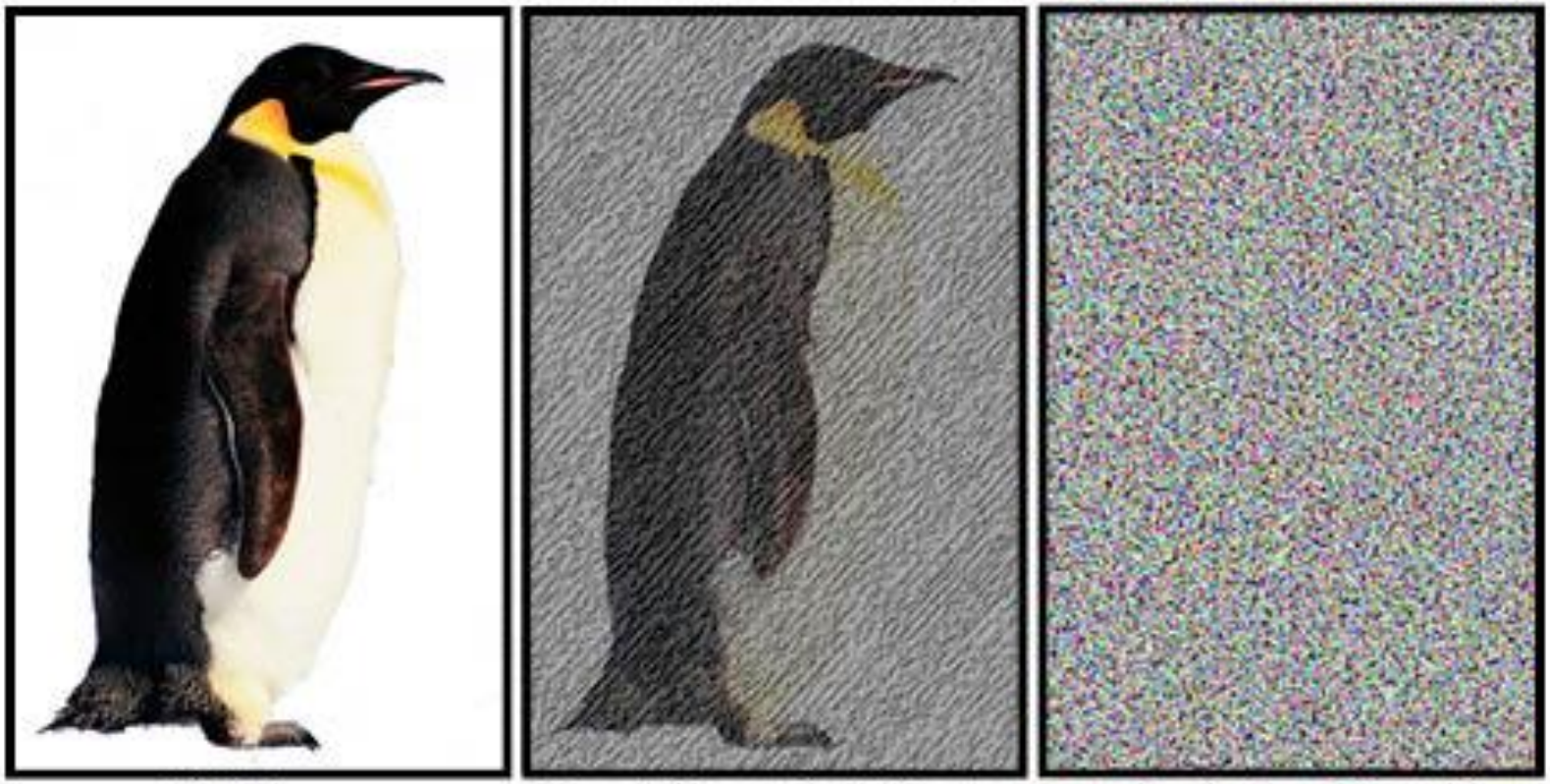
Remarks on ECB

- Strength: it's simple.

- Weakness:
  - Repetitive information contained in the plaintext may show in the ciphertext, if aligned with blocks.
  - If the same message (e.g., an SSN) is encrypted (with the same key) and sent twice, their ciphertexts are the same.

- Typical application: secure transmission of short pieces of information (e.g. a temporary encryption key)

Example of ECB



Original Image

Encrypted image using ECB mode

Modes other than ECB result in pseudorandomness

Cipher Block Chaining (CBC)

- Solve security deficiencies in ECB
  - Repeated same plaintext block result different ciphertext block

- Each previous cipher blocks is chained to be input with current plaintext block, hence name

- Use Initial Vector (IV) to start process

$$C_i = E_K (P_i \text{ XOR } C_{i-1})$$

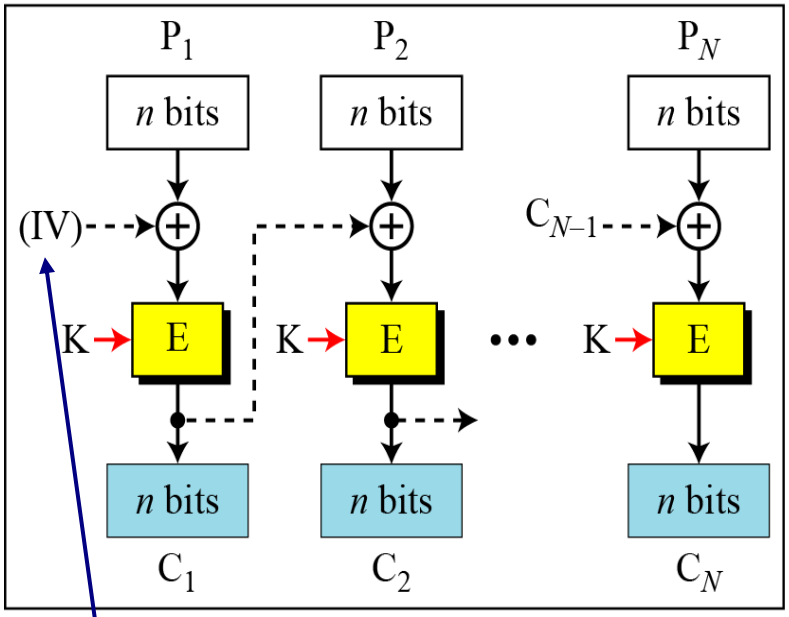
$$C_0 = IV$$

- Uses: bulk data encryption, authentication

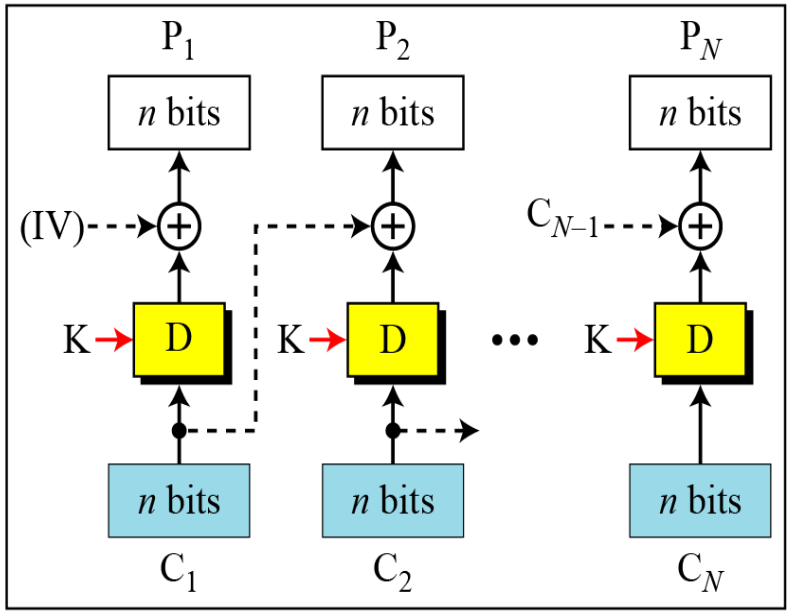
CBC scheme

E: Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i : Ciphertext block i
IV: Initial vector (C_0)



Encryption



Decryption

Encryption:

$C_0 = IV$

$C_i = E_K (P_i \oplus C_{i-1})$

Decryption:

$C_0 = IV$

$P_i = D_K (C_i) \oplus C_{i-1}$

Cipher feedback mode (CFB) Scheme

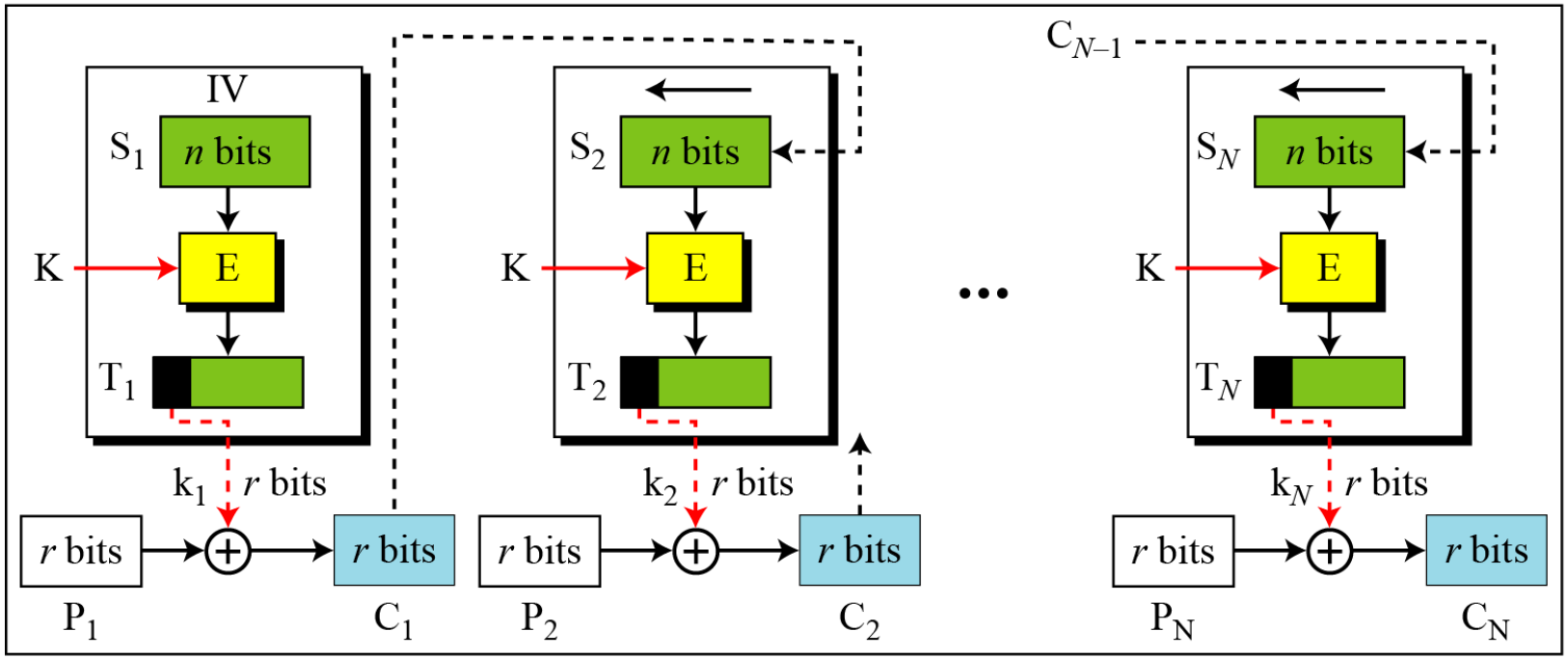
Encryption: $C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

Decryption: $P_i = C_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

E : Encryption
P_i: Plaintext block i
K: Secret key

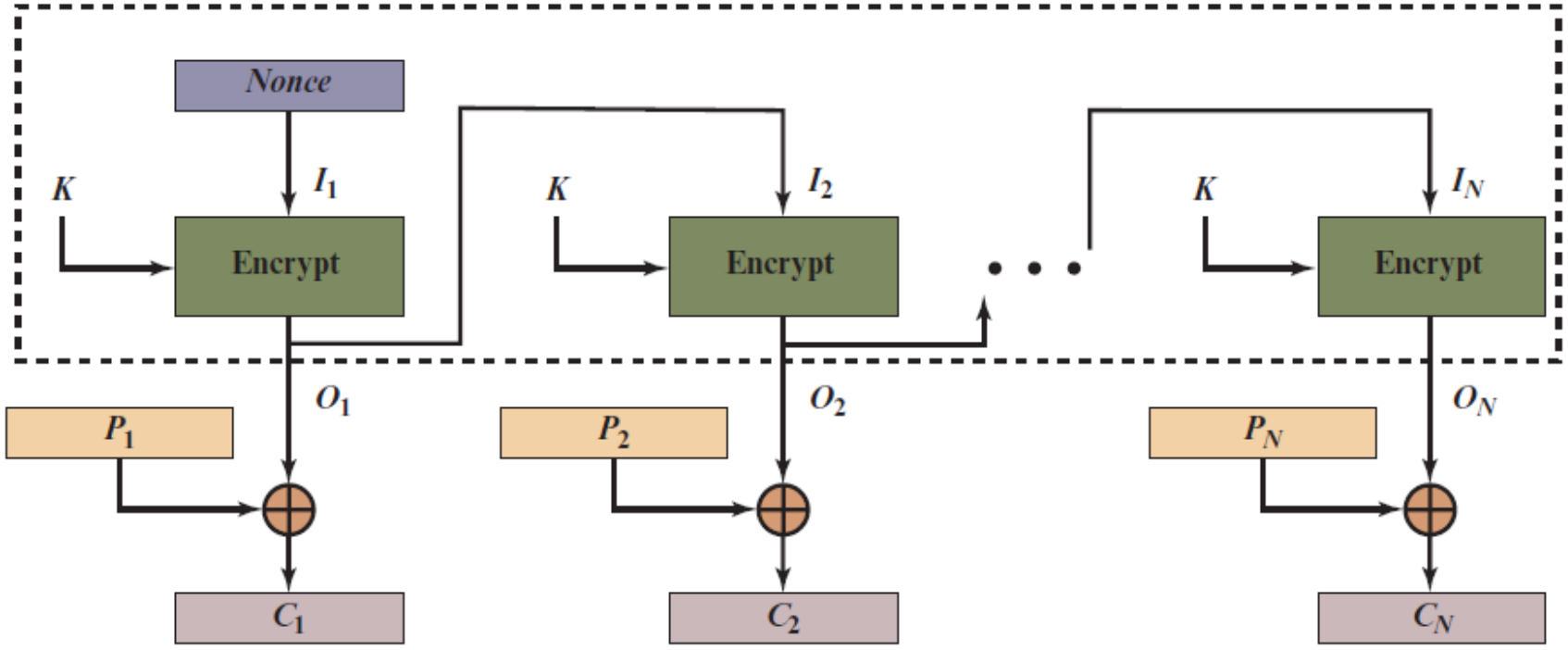
D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



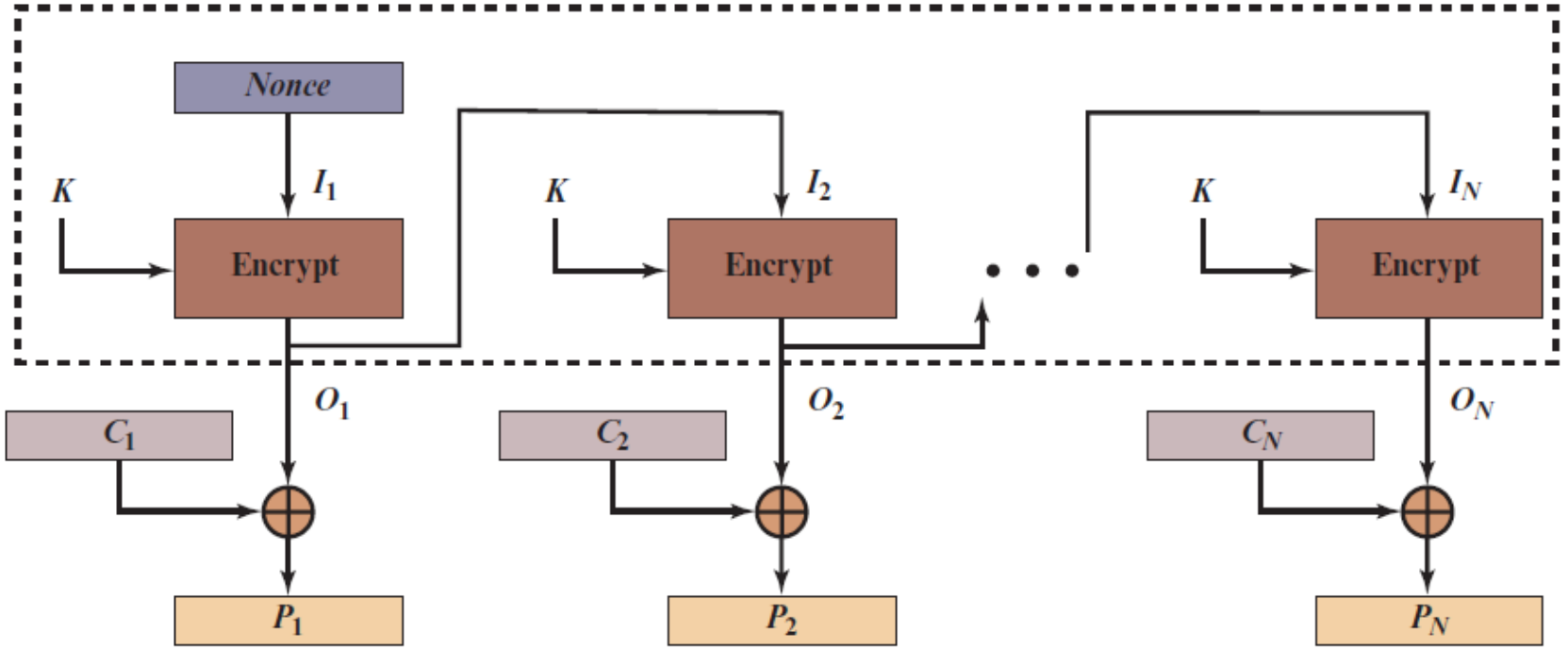
Encryption

Figure 7.6 Output Feedback (OFB) Mode (1 of 2)



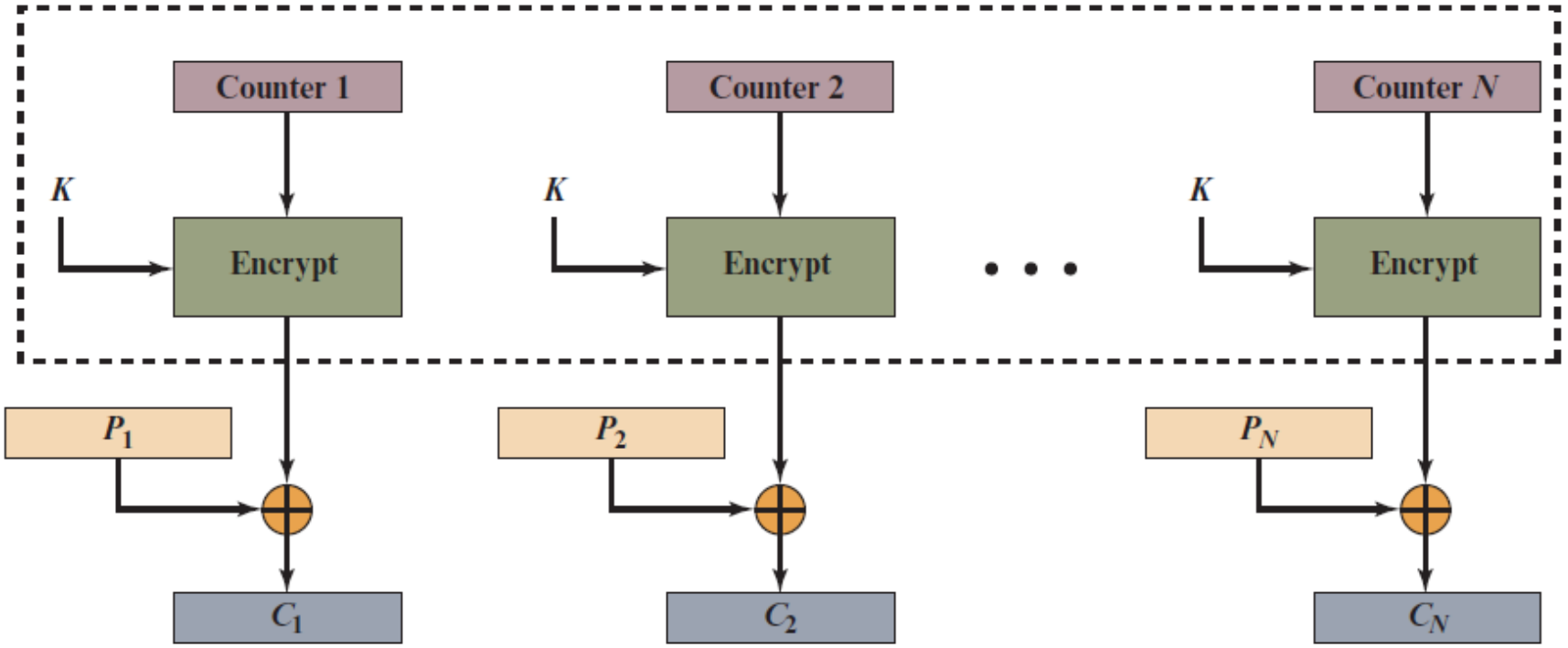
(a) Encryption

Figure 7.6 Output Feedback (OFB) Mode (2 of 2)



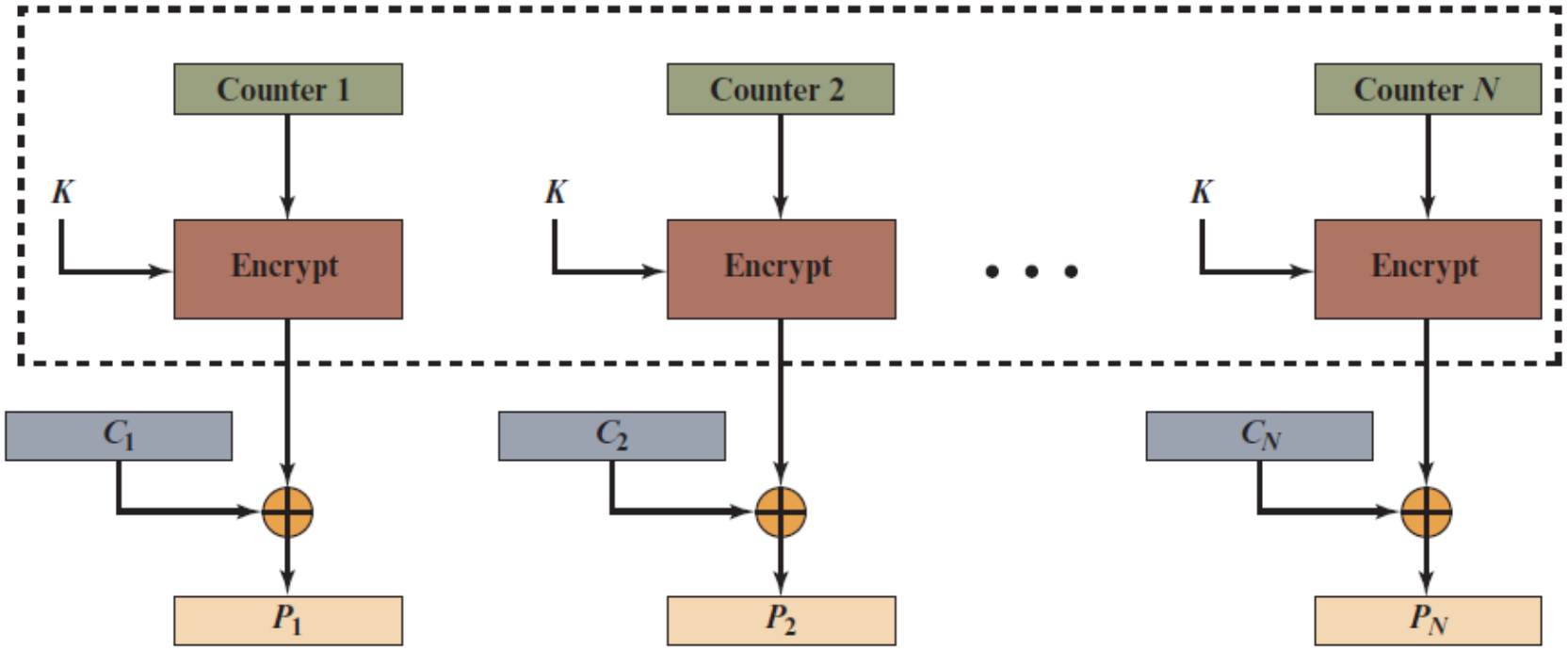
(b) Decryption

Figure 7.7 Counter (CTR) Mode (1 of 2)



(a) Encryption

Figure 7.7 Counter (CTR) Mode (2 of 2)



(b) Decryption