



# Distributed and Edge Computing

# Distributed Systems Security

**Sharad K. Ghimire**

Department of Electronics and Computer Engineering  
Pulchowk Campus  
Institute of Engineering  
Tribhuvan University

# Contents

Overview of cryptography and data privacy

Modern Ciphers

Confusion and Diffusion

Block and Stream Ciphers

Block Ciphers: DES, AES

Block Cipher modes of operation



# Modern Ciphers

# Confusion and Diffusion

Claude Shannon defined these properties that a good cryptosystem should have to hinder statistical analysis

# Confusion

Confusion means that the key does not relate in a simple way to the cipher-text, i.e., **each character** of the **ciphertext** should depend on **several parts** of the key

Defines making the relationship between the key and the cipher as difficult and as possible

Specifies that the ciphertext provides no clue about the plaintext

# Confusion

The relationship between the data of the cipher text and the value of the encryption has to remain as difficult as applicable

The main goal of confusion is to create it very complex to find the key even if one has most of the plaintext-ciphertext pairs produced with the similar key

Each bit of the Ciphertext should be based on the complete key and in several ways on multiple bits of the key, changing one bit of the key must change the Ciphertext completely

# Diffusion

If we change a character of plaintext, then several characters of the ciphertext should change, similarly, if we change a character of the ciphertext, then several characters of the plaintext should change

Diffusion can define to the property that the repetition in the statistics of the plaintext is “dissipated” in the statistics of the Ciphertext

The output bits should be based on the input bits in a difficult way so that in case one bit of the plaintext is modified, thus the cipher-text should change completely in an unstable or pseudo-random manner



# Diffusion

The statistical mechanism of the plaintext is used up into a high-range data of the ciphertext  $\Rightarrow$  it is achieved by having each plaintext digit influence the value of some ciphertext digits

Similarly each ciphertext digit be influenced by some plaintext digits

The frequency statistics of characters in the plaintext are diffused over several characters in the ciphertext

# Confusion and Diffusion

Confusion and diffusion are both cryptographic approaches

**Confusion:-** The relationship between the ciphertext statistics and the encryption key value  $\Rightarrow$  as difficult as possible

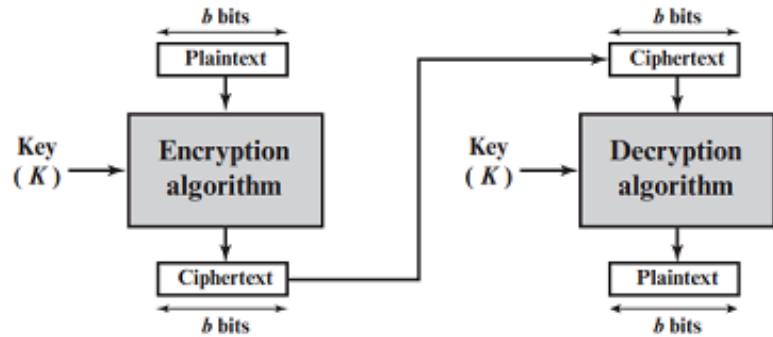
**Diffusion:-** Attempt to disguise the statistical nature of the plaintext by spreading the influence of each individual plaintext digit over a large number of ciphertext digits

Some ciphers can be confusion-only or diffusion-only, but any "reasonable" block cipher uses both confusion and diffusion



# Block & Stream Ciphers

# Block Cipher



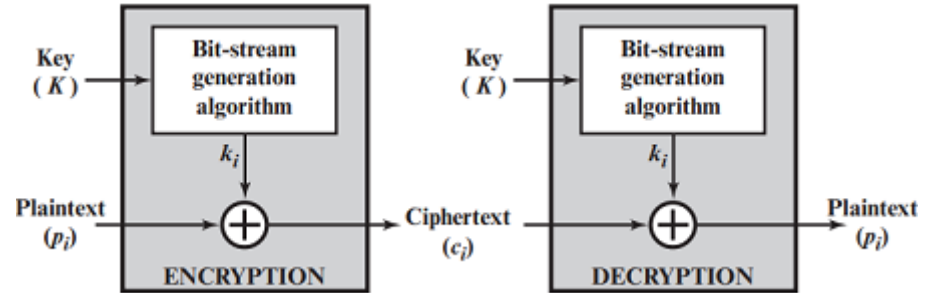
A group of plaintext symbols of size  $m$  ( $m > 1$ ) are encrypted together creating a group of ciphertext of the same size

A single key is used to encrypt the whole block even if the key is made of multiple values

Most modern symmetric encryption algorithms are block ciphers

A block of plaintext is treated as a whole and used to produce a ciphertext block of equal length  $\Rightarrow$  typical block size of 64 or 128 bits

# Stream Cipher



Convert each symbol of plaintext directly into a symbol of cipher-text

Encryption and decryption are done one symbol (such as a character or a bit) at a time  $\Rightarrow$  needs a plaintext stream, a ciphertext stream, and a key stream

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

Call the plaintext stream  $P$ , the ciphertext stream  $C$ , and the key stream  $K$

$$P = P_1 P_2 P_3, \dots$$

$$C = C_1 C_2 C_3, \dots$$

$$K = (k_1, k_2, k_3, \dots)$$

$$C_1 = E_{k1}(P_1) \quad C_2 = E_{k2}(P_2) \quad C_3 = E_{k3}(P_3) \dots$$

# Block & Stream Cipher

## **Block Cipher:**

A block cipher is a deterministic algorithm operating on fixed length groups of bits called blocks

The size of block is defined by the algorithm, such as 64 bits, 128 bits etc.

## **Stream Cipher:**

Each plaintext message is encrypted one at a time with the corresponding digit of the keystream, to give a ciphertext stream

# Stream Cipher

# Stream Cipher

Encrypts plaintext of one byte or one bit or similar unit at a time

More efficient for real-time processing

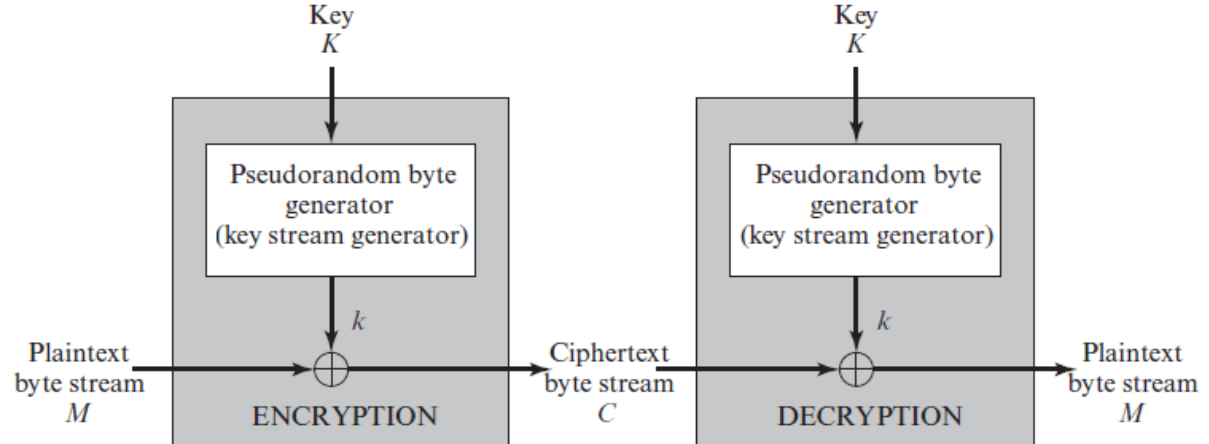
Several stream ciphers have been used in different protocols during the last few decades

E.g. RC4



In stream shown in figure, cipher a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random

The output of the generator, called a keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation



**Stream Cipher Diagram**

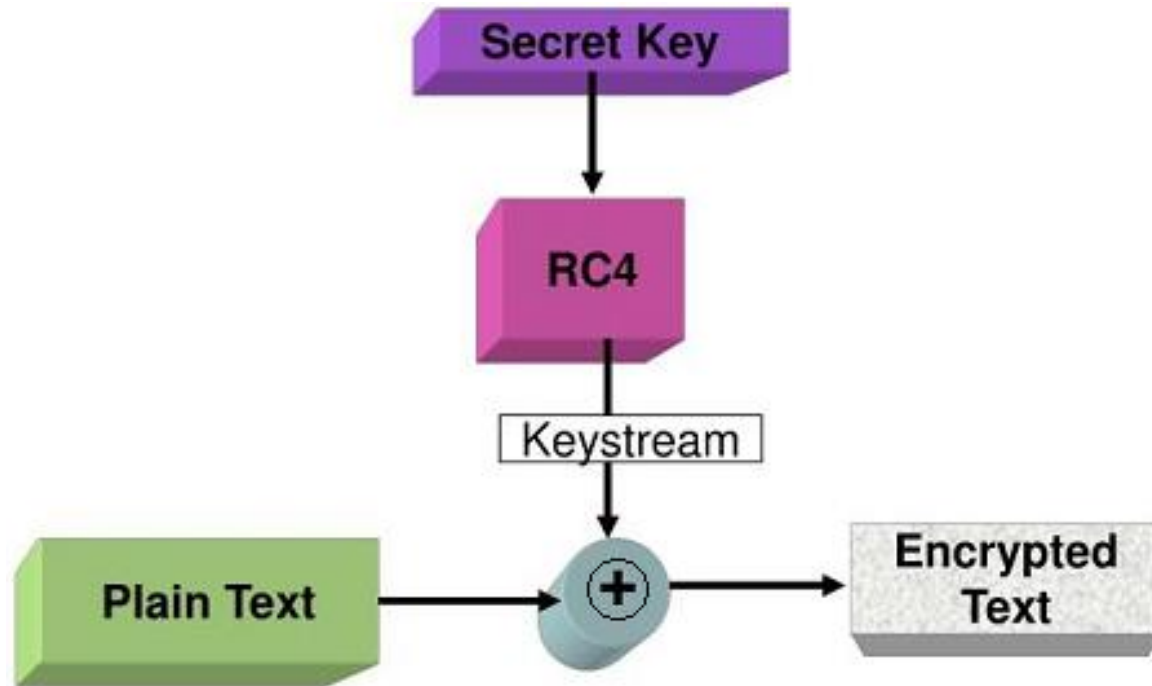
# Example

E.g., let the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is

	11001100	plaintext
$\oplus$	01101100	key stream
	10100000	ciphertext

Decryption requires the use of the same pseudorandom sequence as:

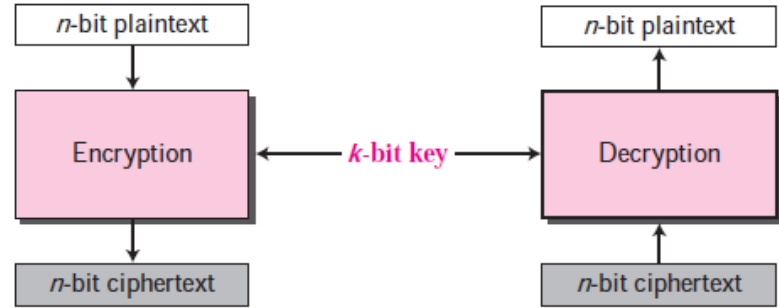
	10100000	ciphertext
$\oplus$	01101100	key stream
	11001100	plaintext



**Block Diagram of RC4 Cipher**

# Block Ciphers

# Block Ciphers



Most common symmetric algorithms are block ciphers

A block cipher encrypts block of predefined number bits and generates same number of bits as ciphertext

Made up of fixed number of bits in a block (typically 64 bits or 128 bits)

Use a key of fixed sized typically 56 bits, 128 bits, 192 bits or 256 bits long

Block ciphers are frequently used to encrypt large amounts of data into data blocks

# Modern Block Ciphers

Most important symmetric algorithms, using block ciphers:

- Data Encryption Standard (DES) and Triple-DES (3DES)
- Advanced Encryption Standard (AES)



# Data Encryption Standard (DES)

# Data Encryption Standard (DES)

Designed by IBM

Adopted by US government as a standard encryption

Dominant encryption since its introduction in 1977

64 bit plaintext blocks with 64 bit ciphertext

56 bit key

Broken in 1998 by Electronic Frontier Foundation



# DES

Based on the Feistel block cipher, called LUCIFER, developed in 1971 by IBM cryptography researcher Horst Feistel

DES was adopted by the National Bureau of Standards (now called the National Institute of Standards and Technology) as Federal Information Processing Standard

Uses 16 rounds of the Feistel structure, using a different key for each round

Over the years, DES became the dominant symmetric encryption algorithm

Plaintext 64, ciphertext 64-bit and key size of 56-bits

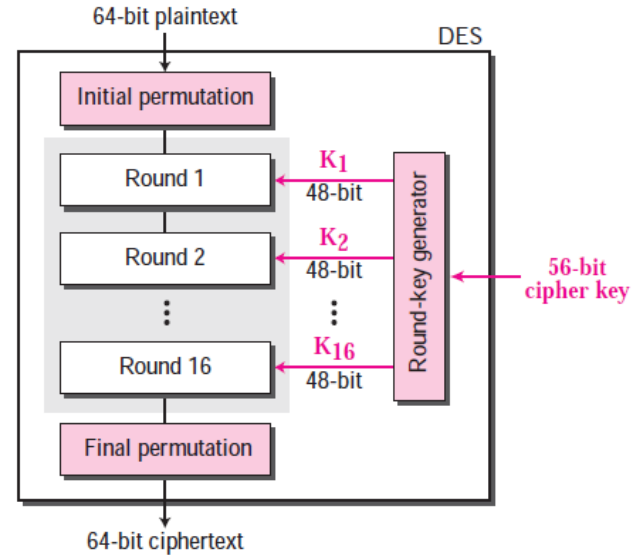
# DES Structure

DES is based on the Feistel Cipher

DES functions

- Round function
- Key schedule
- Additional processing – Initial and final permutation

# DES Structure



**Initial & Final Permutations:** Predefined permutation of 64 bits data

**Rounds:** DES uses 16 rounds, and each round is a Feistel cipher

Each round has a different operations: mixing and swapping

Round-key generator generates different keys for each round from given 56 bit key

# DES Properties

Just only one bit difference in plaintext causes significant differences in the bits of ciphertext

**Avalanche Effect:** A small change in the plaintext (or key) creates a significant change in the ciphertext  $\Rightarrow$  DES has been proved to be with this property

Plaintext: 0000000000000000

Key: 22234512987ABB23

Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001

Key: 22234512987ABB23

Ciphertext: 0A4ED5C15A63FEA3

Each bit of the ciphertext depends on many bits on the plaintext

# DES strength/weakness

The strength of DES lies on following facts:

**The nature of algorithm:** Cryptanalyst can perform cryptanalysis by exploiting the characteristic of DES algorithm but no one has succeeded in finding out the weakness

**The size of key:** 56-bit key is used in encryption, there are  $2^{56}$  possible keys. A brute-force attack on such number of keys is impractical at that time in past, but it is not considered as secure now due to increased processing power of computers  $\Rightarrow$  Key size is the main weakness of DES

# DES

Broken in 1998 by Electronic Frontier Foundation

- Special purpose US\$250,000 machine
- With detailed published description
- Less than three days

Only 56-bit key is considered as not secure

DES is considered as worthless now

# How to Increase the Security?

Modification/Amendment of current algorithm

OR

Switch to a newer suitable algorithm

# Multiple DES

As the major criticism of DES regards its key length  $\Rightarrow$  with available technology and the possibility of parallel processing, a brute-force attack on DES is feasible

In the approach of using multiple (cascaded) instances of DES with multiple keys; does not require an investment in new software and hardware  $\Rightarrow$  one approach is use double DES (**2DES**)

2DES was not found to be effective



# Triple DES

To improve the security of DES, triple DES (3DES) was proposed  
Uses three stages of DES for encryption and decryption

# Triple DES

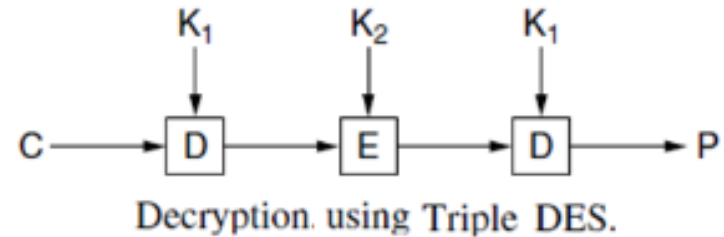
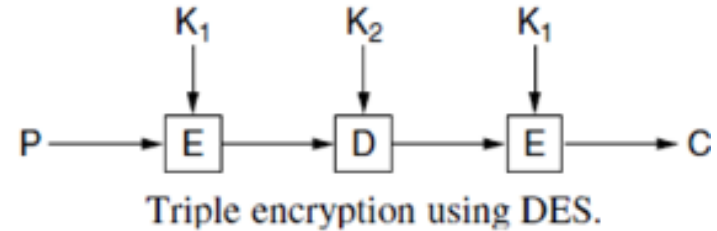
Two versions: using **2 keys** or **3 keys**

3 executions of DEA algorithm

Sequence of E - D - E in encryption

Sequence of D - E - E in decryption

Effective key length 112 or 168 bit



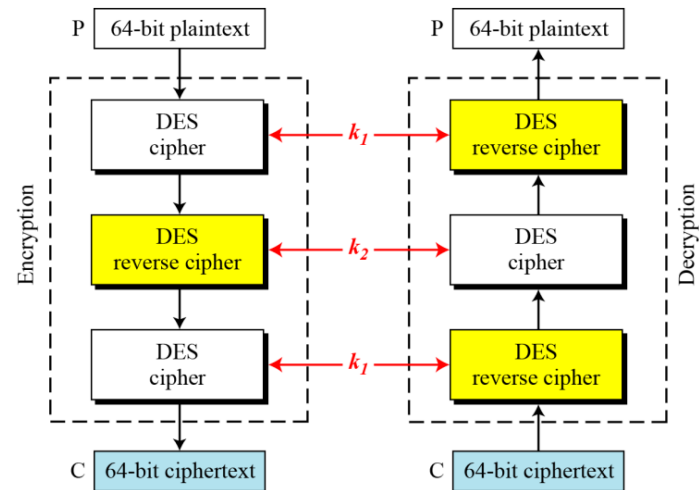
# Triple DES with 2 Keys

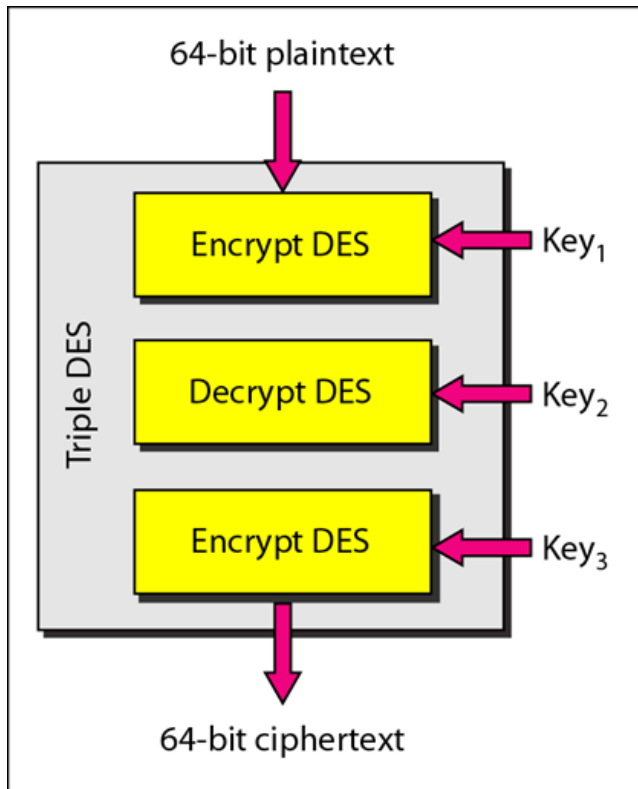
Only two keys:  $k_1$  and  $k_2$

The first and the third stages use  $k_1$ ; the second stage uses  $k_2$

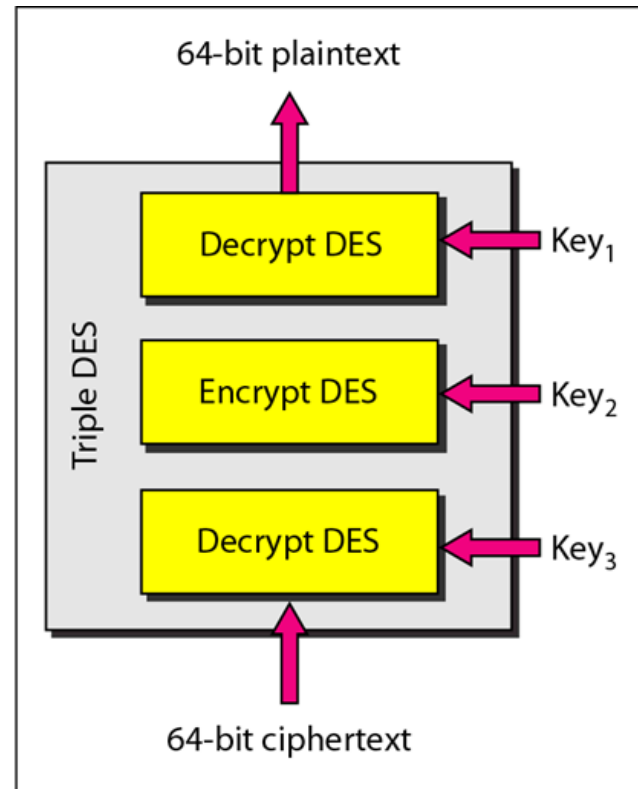
To make it compatible with single DES, the middle stage uses decryption in the encryption site and encryption in the decryption site

A message encrypted with single DES with key  $k$  can be decrypted with triple DES if  $k_1 = k_2 = k \Rightarrow$  much stronger than double DES





a. Encryption Triple DES



b. Decryption Triple DES

## Triple DES with Three Keys

# 3DES Analysis

Breaking 3DES is not practical  $\Rightarrow$  considered as secure

But it is very slow due to three consecutive blocks of DES (typically in software implementation)  $\Rightarrow$  considered as inefficient

Block size (64 bit) is too small

3DES  $\Rightarrow$  Intermediate solution to address the need of high security

$\Rightarrow$  Need of some efficient algorithm



# Advanced Encryption Standard (AES)

# AES

AES is a variant of the Rijndael block cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen

A symmetric block cipher with the block length of 128 bits

The most common / widely used block cipher in current use

Introduced by NIST in 2001

Security strength equal to or better than 3DES

Significantly improved efficiency

Allows for different key lengths: 128, 192, or 256 bits

Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys

# Why AES ?

Developed to provide the highest level of security for the most sensitive data

Uses large data block for encryption i.e. 128 bits data

AES combines speed and security properly

AES is Unbreakable by even Brute Force (till now)

On the basis of required security level either one of 128 bits or 192 bits or 256 bits key can be used



# Open Call for AES

On January 2, 1997, NIST announced that they wished to choose a successor to DES  $\Rightarrow$  an initiation of an effort to develop the new algorithm

NIST asked for input from interested parties on how the successor should be chosen

NIST made a formal call for algorithms on September 12, 1997

# Advanced Encryption Standard

NIST (National Institute of Standards and Technology) issued call for proposals for the newer algorithm (AES) in 1997

The requirements:

- The algorithm must be a symmetric block cipher
- The full design must be public
- Key lengths of 128, 192, and 256 bits must be supported
- Both software and hardware implementations must be possible
- The algorithm must be public or licensed on nondiscriminatory terms

# AES Submission

21 algorithms submitted

After the First AES Candidate Conference, NIST announced that 15 out of 21 received algorithms had met the requirements and been selected as the first candidates (August 1998)

Fifteen designs were from several countries: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish

# AES Evaluation

The advantages and disadvantages were investigated by cryptographers

They were assessed not only on security, but also on performance in a variety of settings (PCs of various architectures, smart cards, hardware implementations) and on their feasibility in limited environments (smart cards with very limited memory, low gate count implementations, FPGAs)

After the Second AES Candidate Conference, NIST announced that 5 out of 15 candidates: **MARS**, **RC6**, **Rijndael**, **Serpent**, and **Twofish**

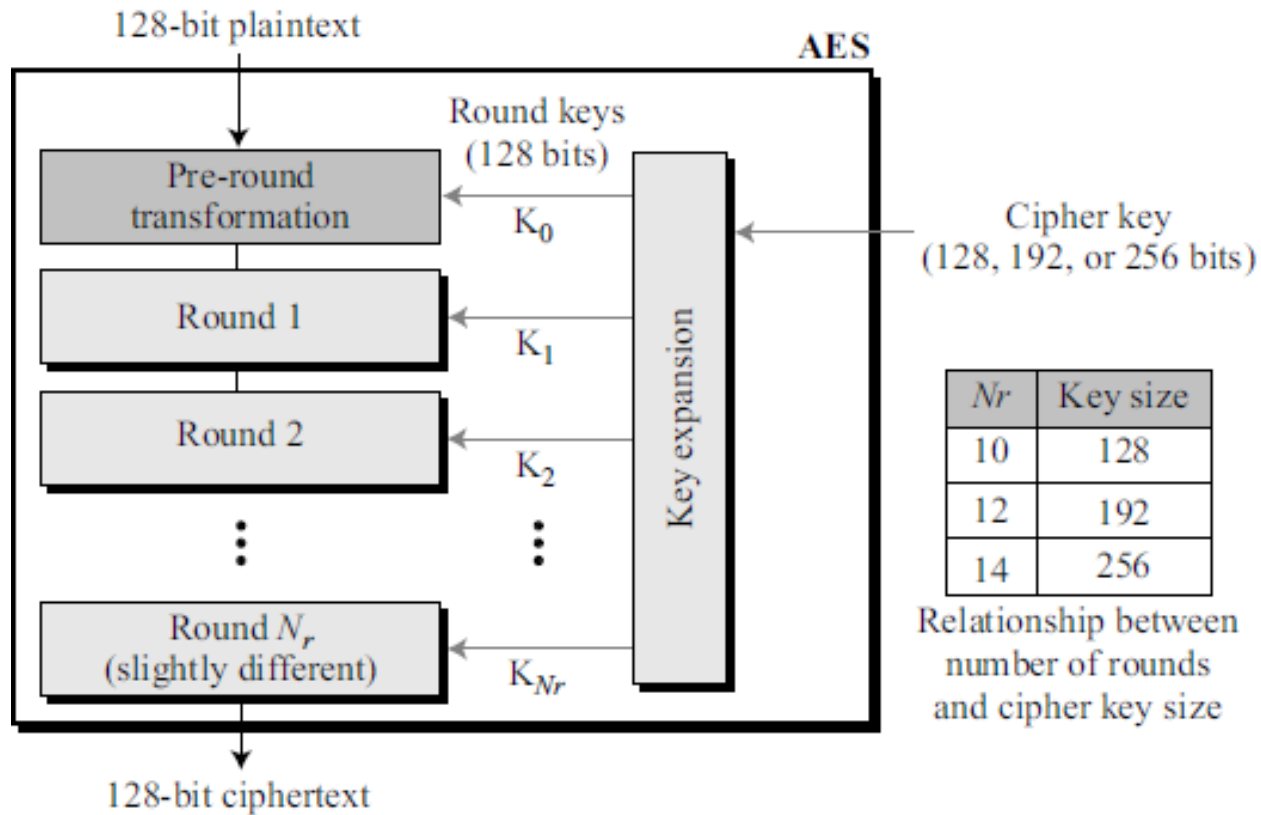
# AES Selection

On October 2, 2000, NIST announced that **Rijndael** had been selected as the proposed AES

In February 2001, NIST announced that a draft of the Federal Information Processing Standard (FIPS) was available for public review and comment

On November 26, 2001, NIST announced that AES was approved as FIPS 197

# AES Structure



## General design of AES encryption cipher

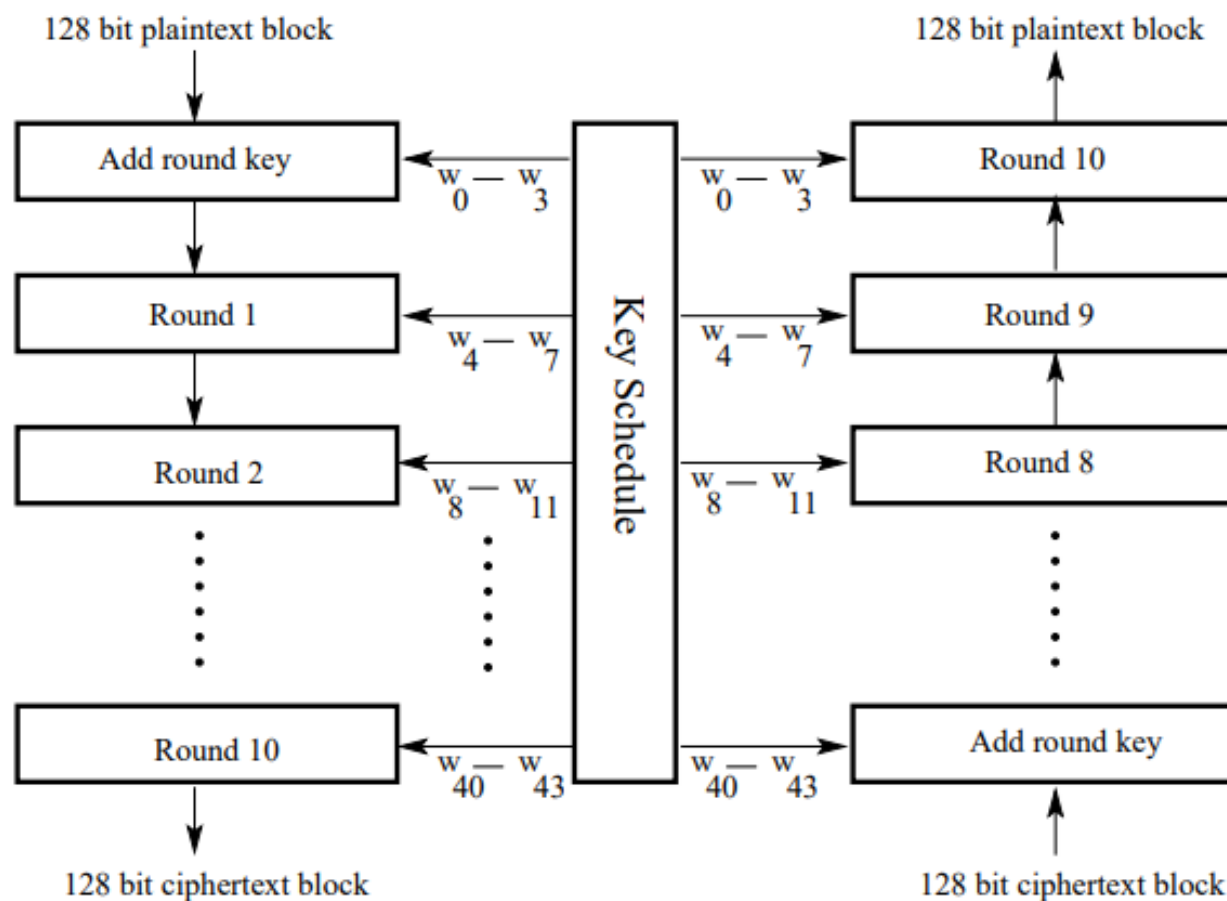
# AES Structure

AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits  
⇒ uses 10, 12, or 14 rounds based on key size that can be 128, 192, or 256 bits

The encryption algorithm (called cipher) and the decryption algorithm (called inverse cipher) is similar, but the round keys are applied in the reverse order

Though the key size can vary i.e., 128, 192 or 256 bits, the round keys, which are created by the key-expansion algorithm are always 128 bits, the same size as the plaintext or ciphertext block

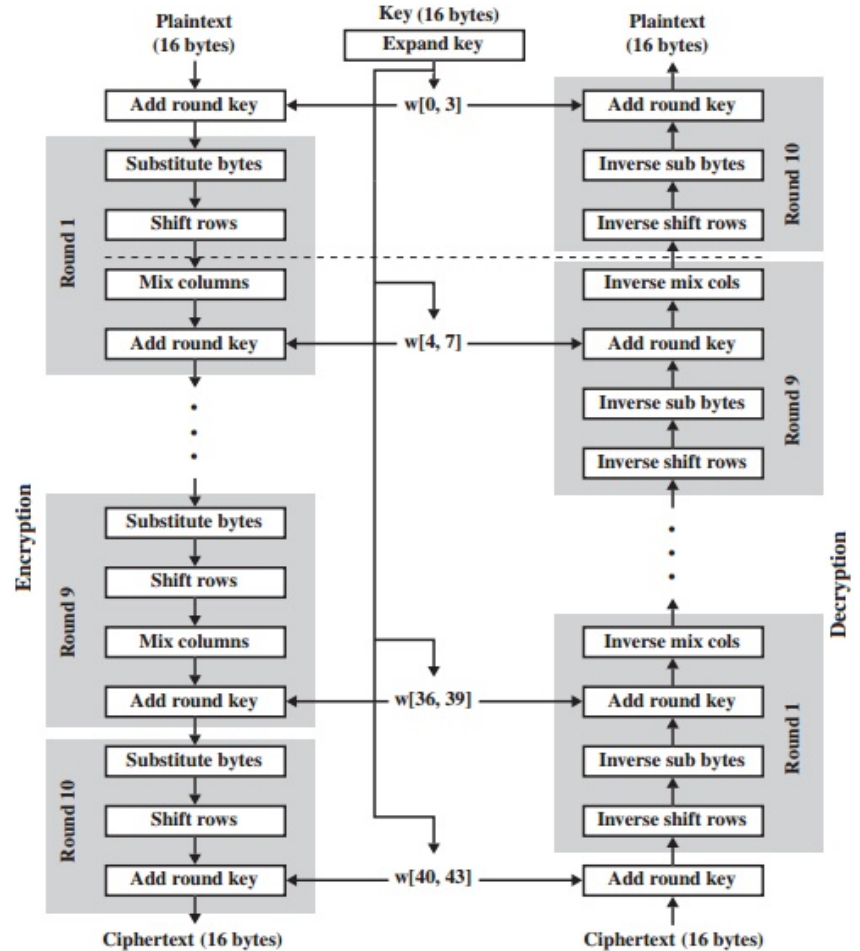




AES Encryption

AES Decryption

# AES Encryption and Decryption





AES has defined three versions, with 10, 12, and 14 rounds.

Each version uses a different cipher key size (128, 192, or 256), but the round keys are always 128 bits.

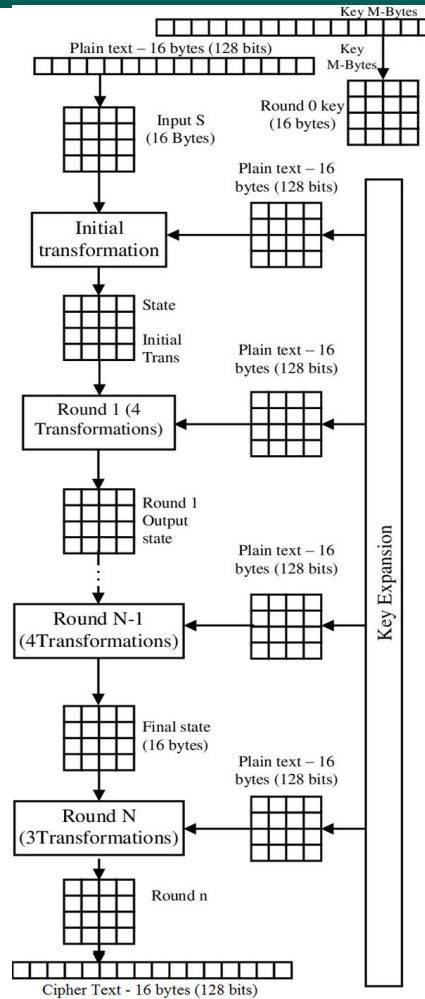


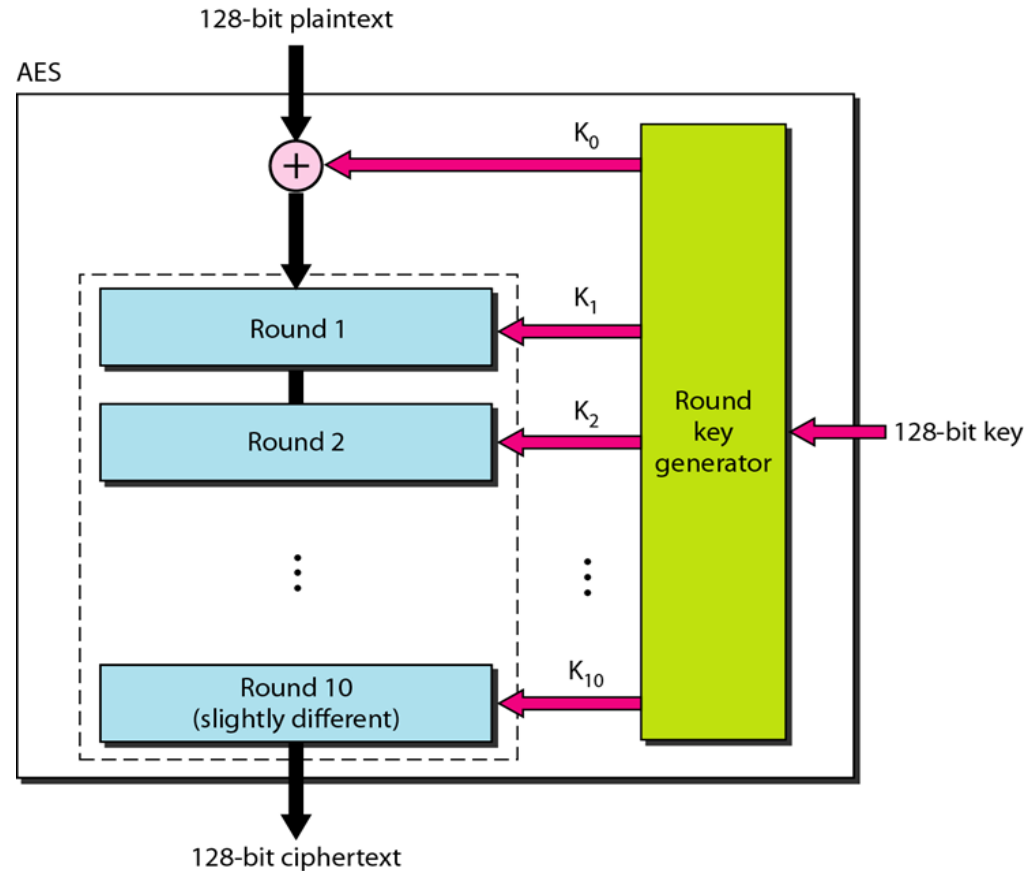
# Transformations

AES uses four types of transformations: substitution, permutation, mixing, and key-adding

- Substitute bytes
- Shift rows  $\Rightarrow$  permutation
- Mix columns
- Add round key

# AES Encryption Process





Structure of AES Cipher with 128 bits key (having 10 rounds only)

# Analysis of AES

Security: AES was designed after DES  $\Rightarrow$  most of the known attacks on DES were already tested on AES; none of them has broken the security of AES so far

AES  $\Rightarrow$  more secure than DES due to the larger-size key (128, 192, 256 bits)

In addition, AES provides two other versions with longer cipher keys

The lack of weak keys is another advantage of AES over DES

### 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	$2^{56} \approx 7.2 \times 10^{16}$	$2^{55} \text{ ns} = 1.125 \text{ years}$	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	$2^{127} \text{ ns} = 5.3 \times 10^{21} \text{ years}$	$5.3 \times 10^{17} \text{ years}$
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	$2^{167} \text{ ns} = 5.8 \times 10^{33} \text{ years}$	$5.8 \times 10^{29} \text{ years}$
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	$2^{191} \text{ ns} = 9.8 \times 10^{40} \text{ years}$	$9.8 \times 10^{36} \text{ years}$
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	$2^{255} \text{ ns} = 1.8 \times 10^{60} \text{ years}$	$1.8 \times 10^{56} \text{ years}$
26 characters (permutation)	Monoalphabetic	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \text{ ns} = 6.3 \times 10^9 \text{ years}$	$6.3 \times 10^6 \text{ years}$





# Ciphering a Long Sequence of Message

# Ciphering a Long Sequence of Message

In real life applications, the text to be enciphered is of variable size and normally much larger than 64 or 128 bits  $\Rightarrow$  **stream cipher** can be used

Block ciphers are strong for encryption  $\Rightarrow$  they are designed to encrypt fixed sized block of message

How to encrypt a **long sequence of message** by using **block ciphers**?

While transmitting a long messages there are some additional issues for secure communication  $\Rightarrow$  different modes of operation using block cipher

# Block cipher modes of operation

## Cipher Modes

- Electronic Code Book Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Counter Mode (CTR)



# Electronic Code Book Mode (ECB)

# Electronic Code Book Mode (ECB)

The simplest mode  $\Rightarrow$  sequence of message  $\Rightarrow$  multiple b-bit blocks

Plaintext is handled one block at a time and each block of plaintext is encrypted using the same key

The term codebook is used because, for a given key, there is a unique ciphertext for every b-bit block of plaintext

We can imagine a gigantic codebook in which there is an entry for every possible b-bit plaintext pattern showing its corresponding ciphertext

# Electronic Code Book Mode (ECB)

For a message longer than  $b$  bits, the message is broken into  $b$ -bit blocks, padding the last block if necessary

Decryption is performed one block at a time, using the same key

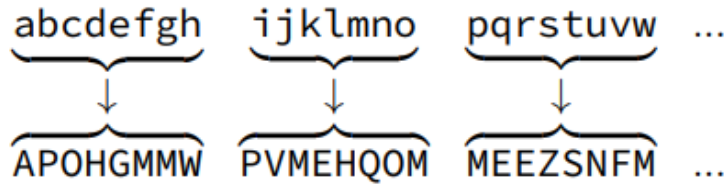
The plaintext (padded as necessary) consists of a sequence of  $b$ -bit blocks,  $P_1, P_2, \dots, P_N$ ; the corresponding sequence of ciphertext blocks is  $C_1, C_2, \dots, C_N$

ECB mode is used as:

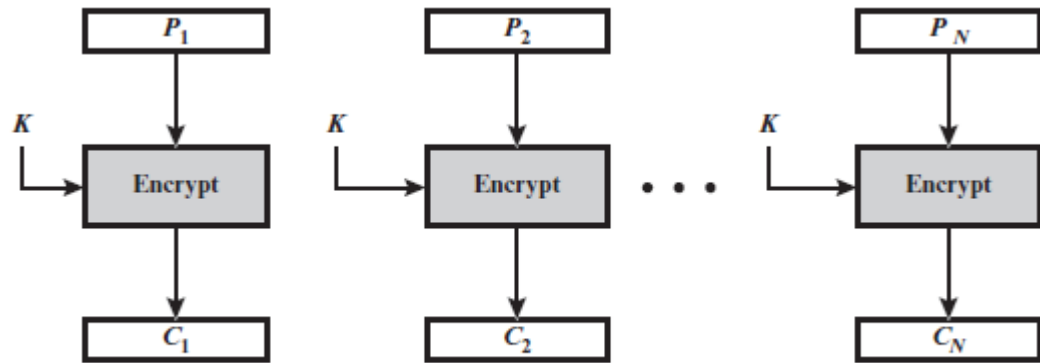
ECB	$C_j = E(K, P_j) \quad j = 1, \dots, N$	$P_j = D(K, C_j) \quad j = 1, \dots, N$
-----	---	---

# Electronic Code Book Mode (ECB)

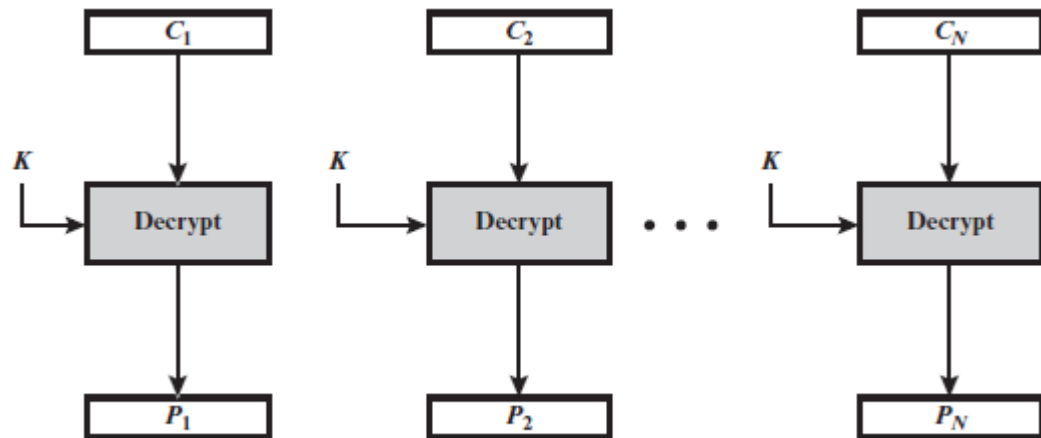
Let us simply divide an incoming stream into different blocks, and encrypt each block



Referred to as the Electronic Code Book method because the encryption process can be represented by a fixed mapping between the input blocks of plaintext and the output blocks of cipher text



Encryption



Decryption

ECB Block Diagram



# Electronic Code Book Mode (ECB) Issues

Simply divide an incoming stream into different blocks, and encrypt each block as:

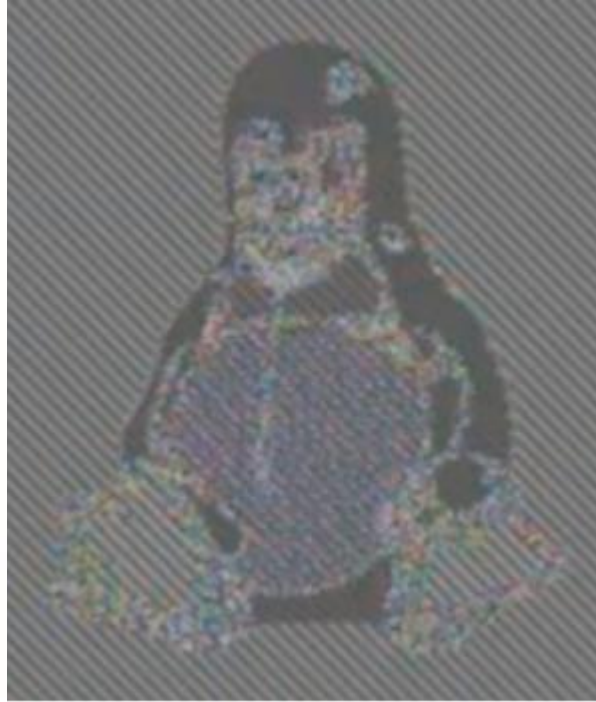
abcdefgh	ijklmno	pqrstuvw	...
↓	↓	↓	
APOHGMMW	PVMEHQOM	MEEZSNFM	...

In this mode, identical input blocks will always map to identical output blocks

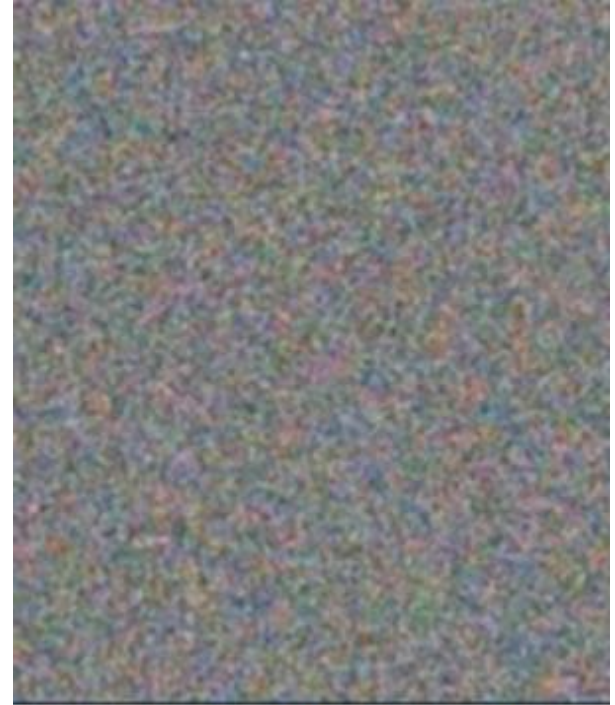
abcdefgh	abcdefgh	abcdefgh	...
↓	↓	↓	
APOHGMMW	APOHGMMW	APOHGMMW	...



**Original Image**



**Encrypted using ECB**



**Encrypted using other modes**

**Electronic Code Book Mode (ECB)**

# Advantages & Limitations

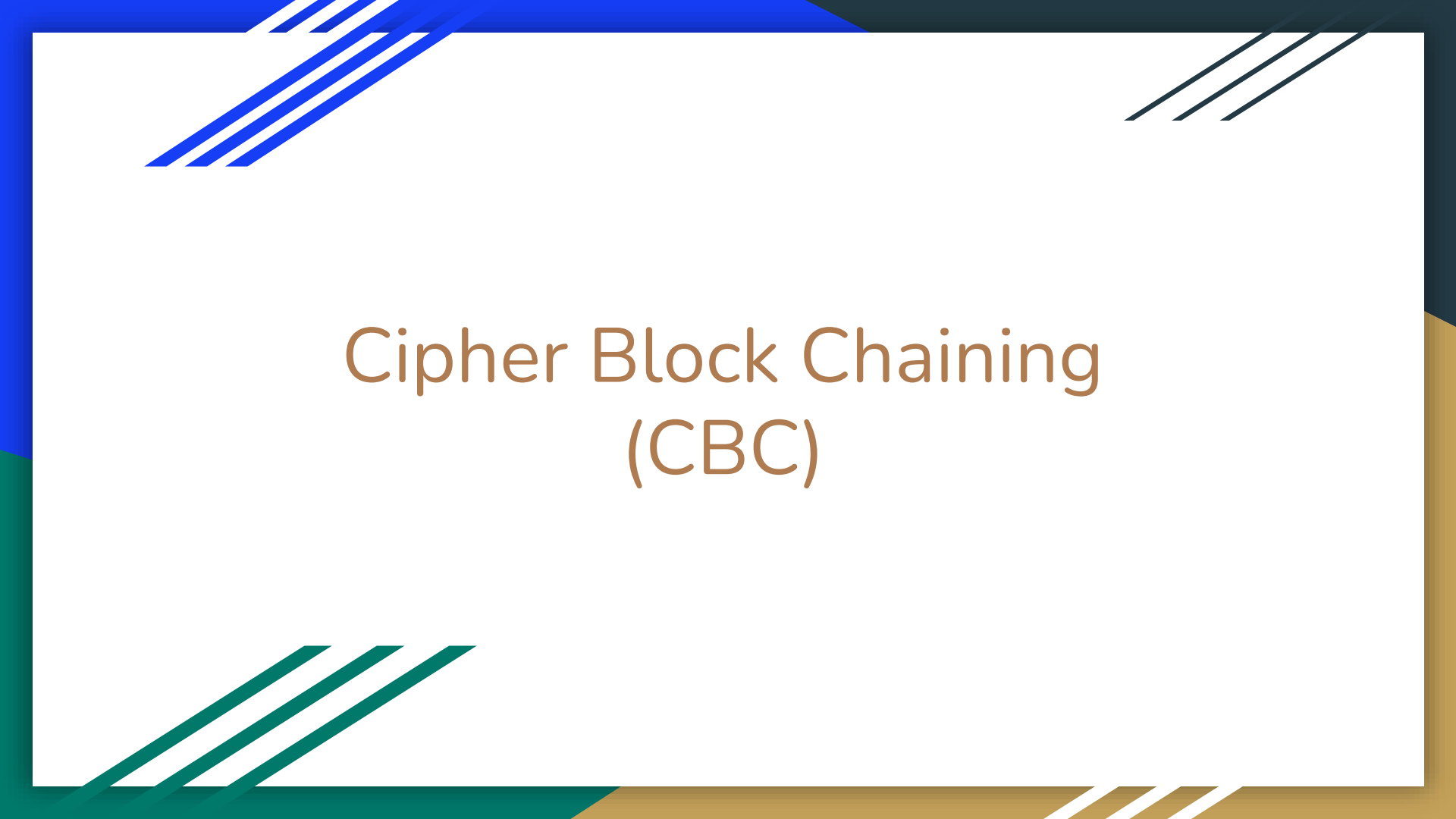
Very simple and easy to implement

Can be used for sending a few blocks of data

Message repetitions may show in ciphertext

Some information can be exposed

- With the messages that change very little
- Particularly with data such as graphics



# Cipher Block Chaining (CBC)

# Cipher Block Chaining (CBC)

To overcome the security deficiencies of ECB  $\Rightarrow$  need of a technique in which the same plaintext block, if repeated, produces different ciphertext blocks  $\Rightarrow$  a way to satisfy this requirement is the cipher block chaining (CBC) mode

In CBC, the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block  $\Rightarrow$  in effect, we have chained together the processing of the sequence of plaintext blocks

The input to the encryption function for each plaintext will be changed, i.e., same plaintext will be different  $\Rightarrow$  repeating patterns of bits are not exposed

# Cipher Block Chaining (CBC)

Message is broken into blocks

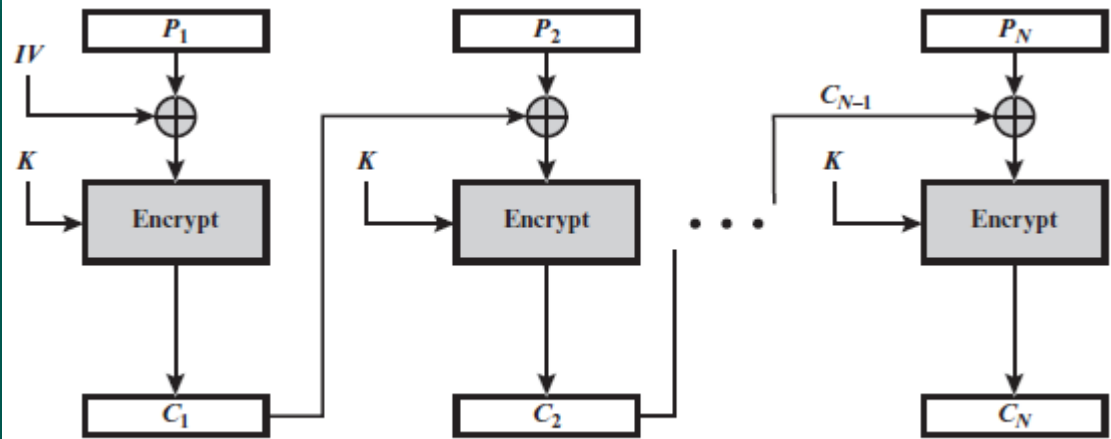
Lined together in encryption operation

Each previous cipher block is chained with current plaintext block, hence name

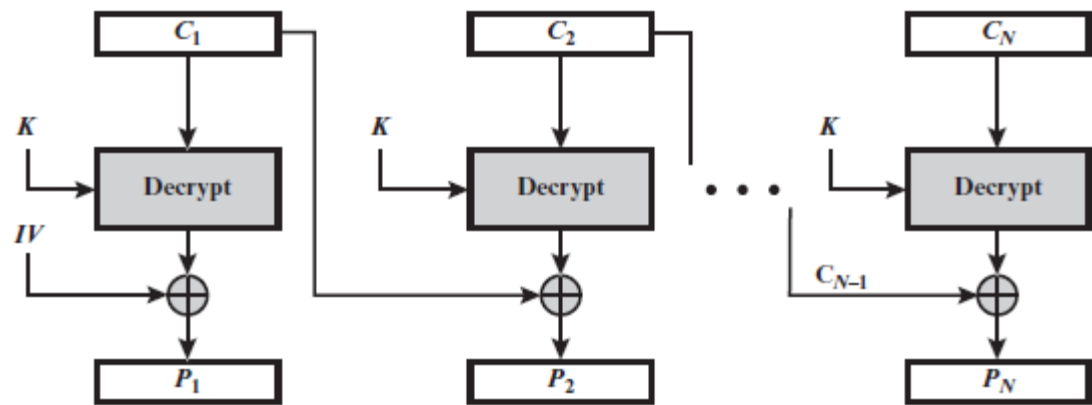
Use initial vector (IV) to start process

- $C_i = \text{DES}_{K1} (P_i \text{ XOR } C_{i-1})$
- $C_{-1} = \text{IV}$  (Initialization vector)

Uses: Bulk data encryption, authentication



**CBC Encryption**

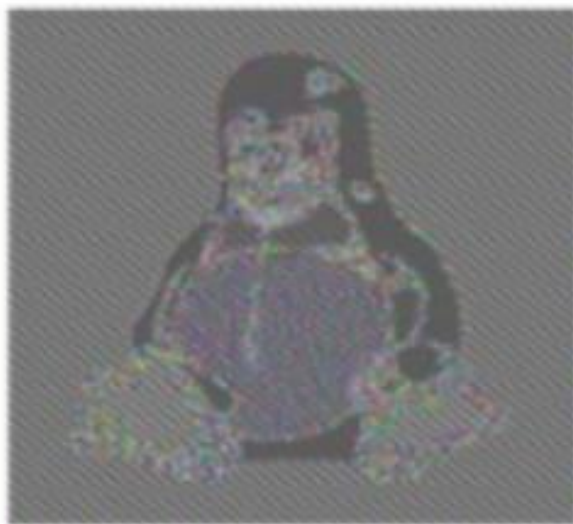


**CBC Decryption**

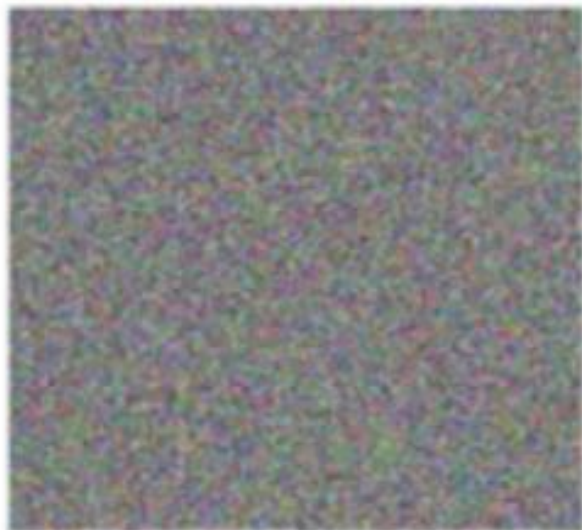
## CBC Block Diagram



**Original image**



**Image with AES  
using ECB**



**Image with AES  
using CBC**



# CBC Analysis

ECB: Independent blocks, Same PT  $\Rightarrow$  Same CT

CBC: Dependent blocks, Same PT  $\Rightarrow$  Different CT

CBC uses concept of chaining  $\Rightarrow$  No parallelism is possible  $\Rightarrow$  Slow

Repeating patterns of bits are not exposed in CBC

Key is same for all blocks as in ECB

Need of IV (Initialization Vector), that should be known to sender & receiver

# Cipher Feedback Mode (CFB)

# Cipher FeedBack (CFB)

For AES, DES, or any block cipher, encryption is performed on a block of  $b$  bits, so as in DES,  $b = 64$  and in of AES,  $b = 128$

However, it is possible to convert a block cipher into a stream cipher, using one of the three modes **CFB** mode, **OFB** mode, and **CTR** mode

A stream cipher eliminates the need to pad a message to be an integral number of blocks, and it also can operate in real time

In stream cipher, each character can be encrypted and transmitted immediately using a character-oriented stream cipher

# Cipher FeedBack (CFB)

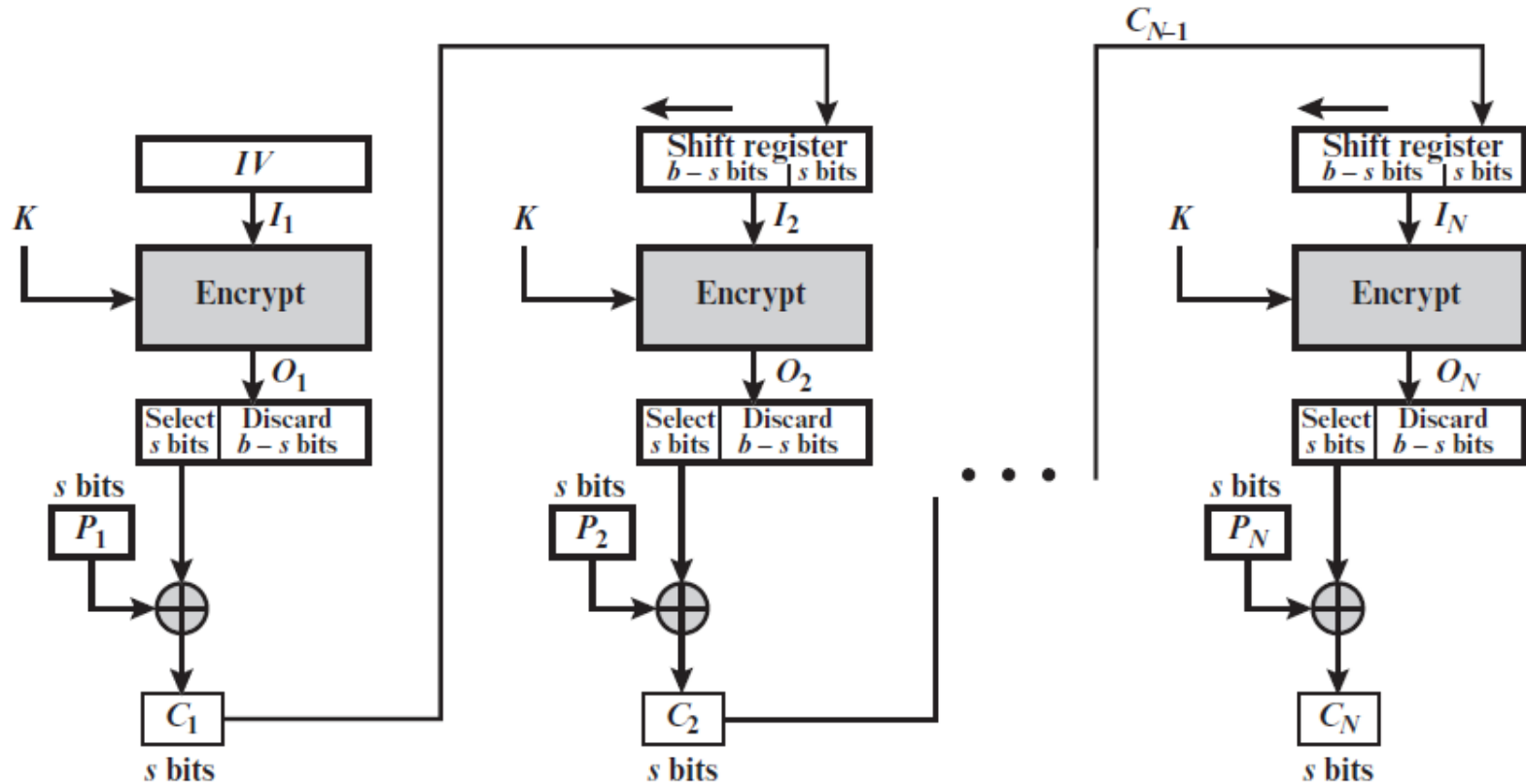
Message is treated as a stream of bits

Added to the output of the block cipher

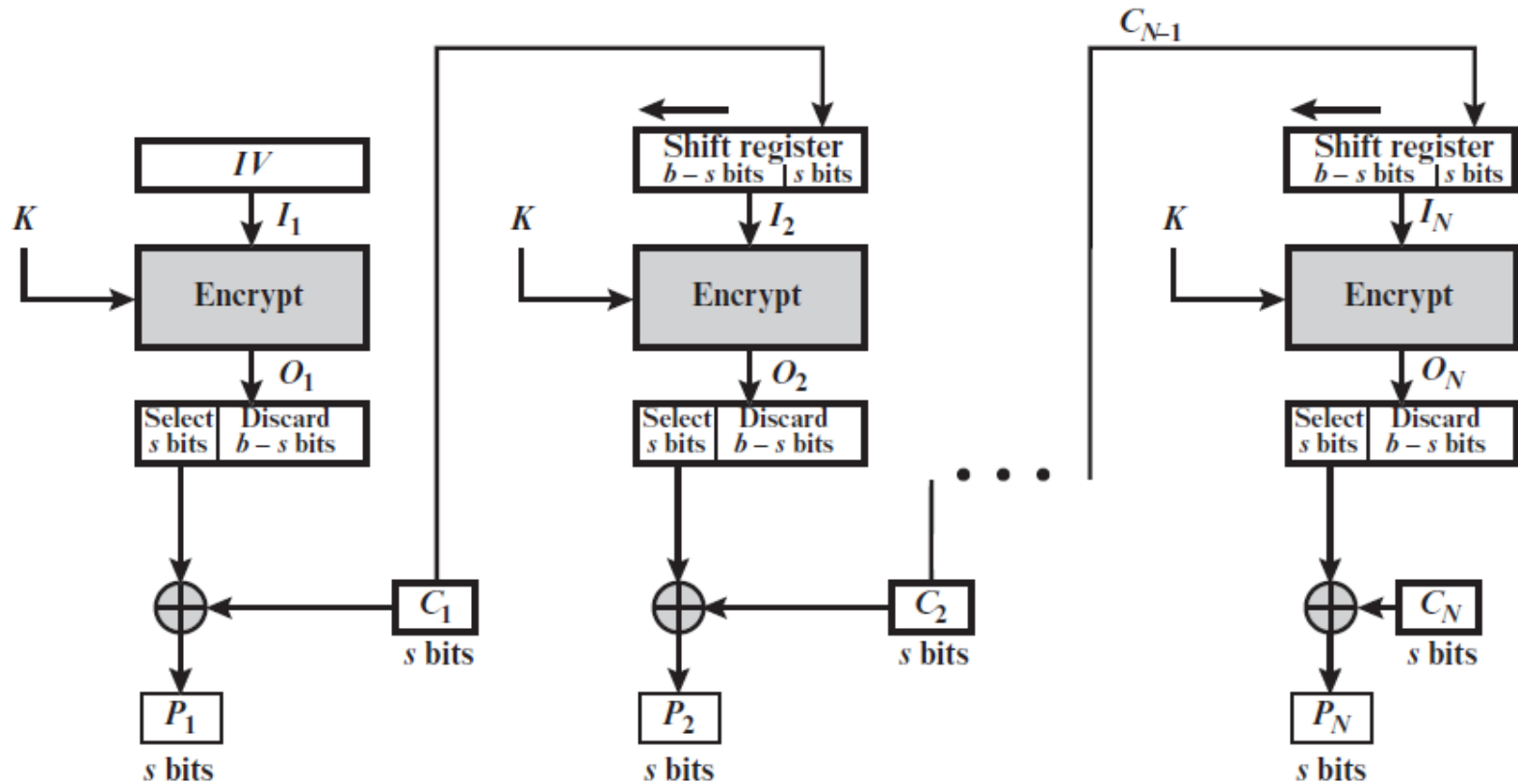
Standard allows any number of bits (1, 8, 64 or 128) to be feedback

- Denoted as CFB-1, CFB-8, CFB-64, CFB-128 etc

Most efficient to use all bits in block (64 or 128)



## CFB Encryption



## CFB Decryption

# CFB

The unit of transmission is  $s$  bits; a common value is  $s = 8$

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

Rather than blocks of  $b$  bits, the plaintext is divided into segments of  $s$  bits

# Output Feedback Mode (OFB)



# The Output Feedback Mode (OFB)

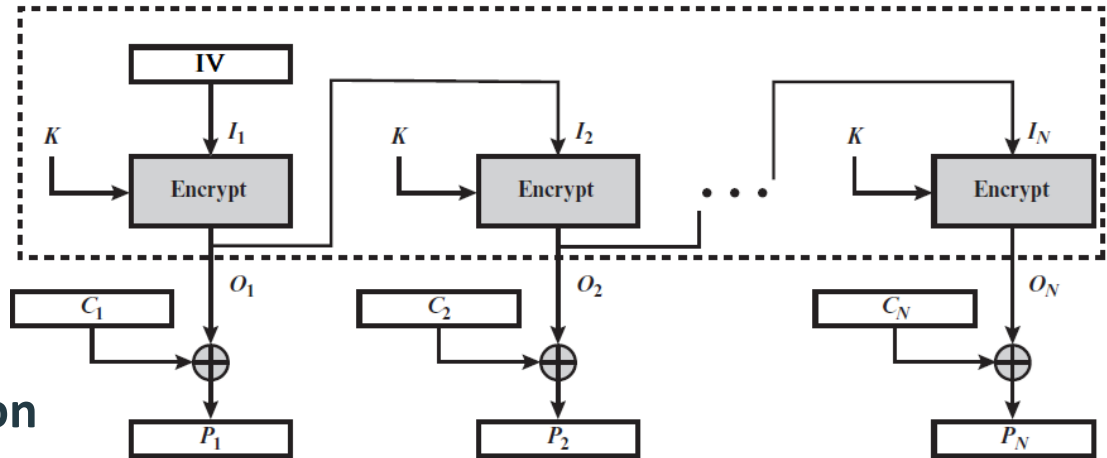
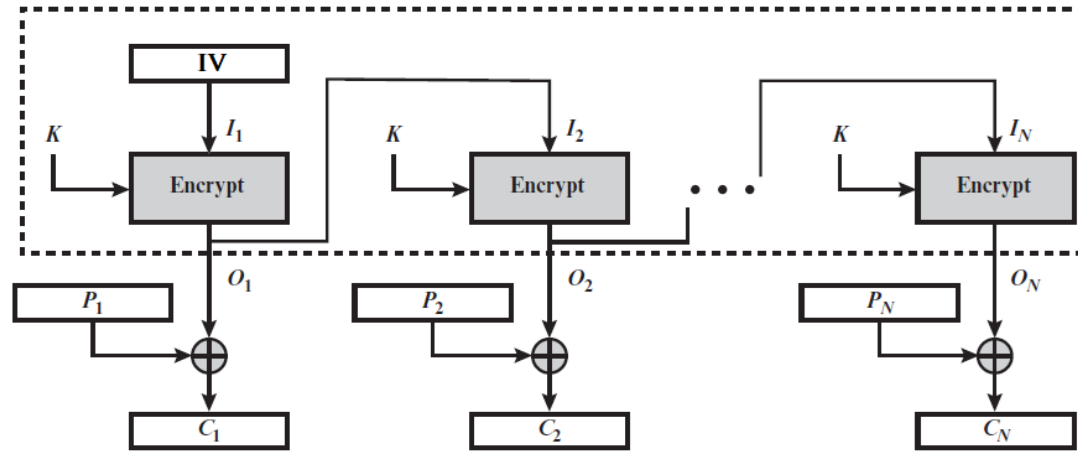
Very similar to the CFB mode

Can also be used as a stream cipher

Some bits of the most significant position are fed back from the output of the block cipher encryption algorithm, rather than actual ciphertext byte

If some bits of the ciphertext get garbled, only those bits of plaintext get garbled; so it is more resistant to transmission bit errors

The message can be of any arbitrary size



## OFB Encryption and Decryption

# OFB Pros & Cons

## Pros:

Bit errors do not propagate, e.g., if a bit error occurs in C1, only the recovered value of P1 is affected; subsequent plaintext units are not corrupted

Same PT with same key  $\Rightarrow$  different CT

PT length can be of random choice

## Cons:

More vulnerable to modification attack, consider that complementing a bit in the ciphertext complements the corresponding bit in the recovered plaintext

# Counter Mode (CTR)

# Counter Mode (CTR)

CTR mode makes block cipher way of working similar to a stream cipher

Although interest in the counter (CTR) mode has increased later on with application to ATM (asynchronous transfer mode) network security and IPsec (IP security), this mode was proposed in 1979

In this mode, subsequent values of an increasing counter are added to a nonce value (the nonce means a number that is unique: number used once) and the results are encrypted as usual

The nonce plays the same role as initialization vectors in the previous modes

# CTR

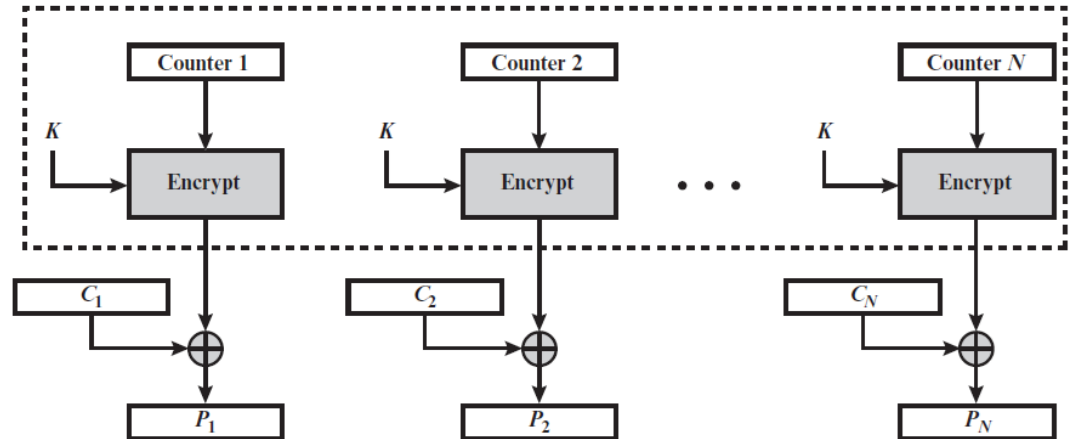
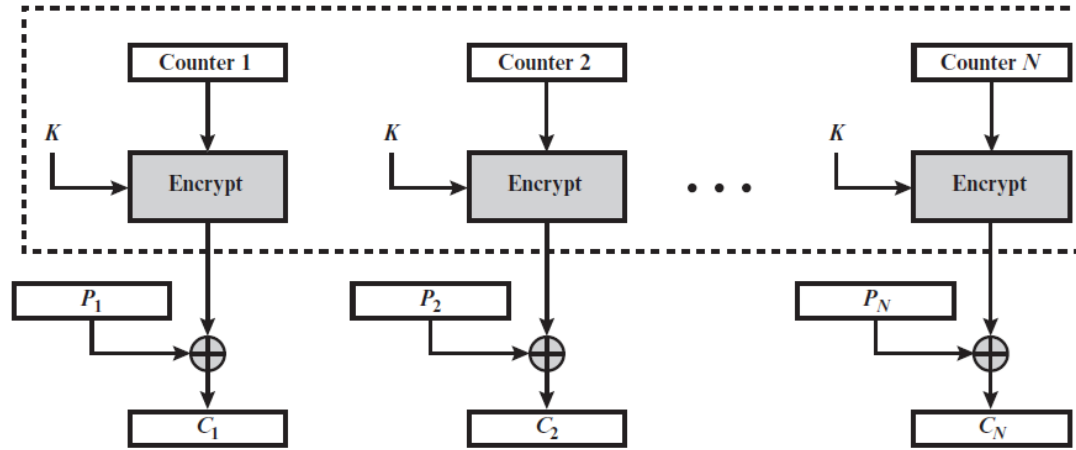
A counter equal to the plaintext block size is used

The counter value must be different for each plaintext block that is encrypted

Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block (modulo  $2^b$ , where  $b$  is the block size)

For encryption, the counter is encrypted and then XORed with the plaintext block to produce the ciphertext block; there is no chaining

For decryption, the same sequence of counter values is used, with each encrypted counter XORed with a ciphertext block to recover the corresponding plaintext block



## CTR Encryption and Decryption

# Counter Mode (CTR)

Fast encryption and decryption

Can be implemented on parallel machines because there is not block-to-block feedback

Any block can be decrypted with random access

The security of CTR is similar with the other modes for using block ciphers