Network and Cyber Security

Bachelor in Computer Engineering

Er. Anuj Sherchan

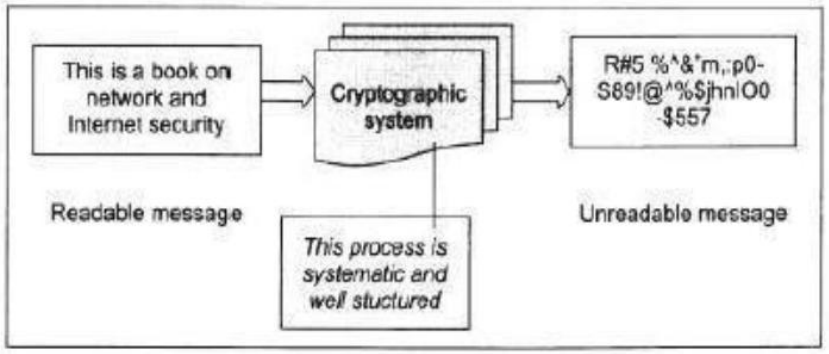
Assistant Professor

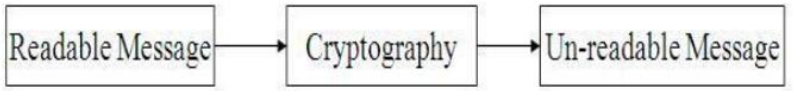
Unit 2: Symmetric Key Cryptography

- Outline:
- 2.1 Cryptology, Cryptography, Cryptanalysis
- 2.2 Conventional Cryptography, Single key Encryption, Computationally Secure Algorithm
- 2.3 Stream Cipher, Substitutional and Transposition Cipher
- 2.4 Block Cipher, Feistel Cipher Structure
- 2.4.1 Data Encryption Standard(DES), 2DES, 3DES
- 2.4.2 Advanced Encryption Standard(AES)

Introduction

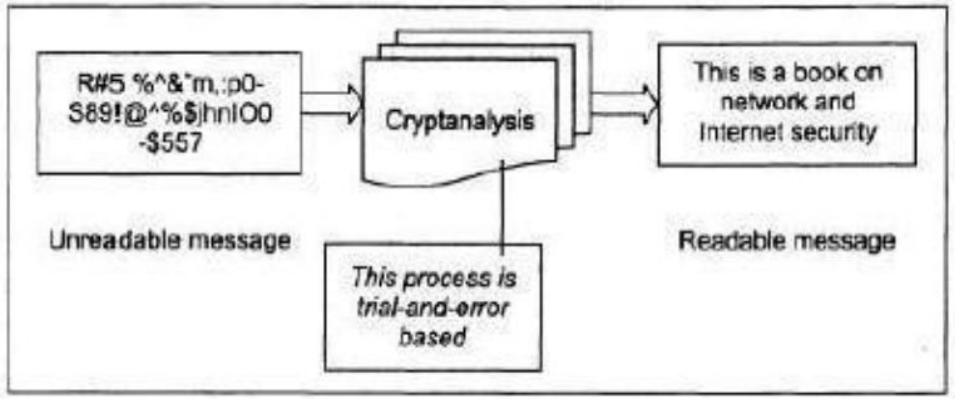
- **Cryptography**
- The word “cryptography“ is the combination of two Greek words, “Krypto” meaning hidden or secret and “graphene” meaning writing.
- It is the art of achieving security by encoding messages to make them nonreadable format.
- It is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it .

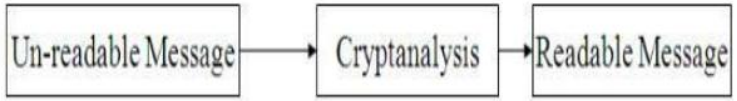




Introduction

- **Cryptanalysis**
- It is the technique of decoding messages from a non-readable format back to a readable format.
- It is done without knowing how they were initially converted from readable format to non-readable format.
- Also called code breaking.





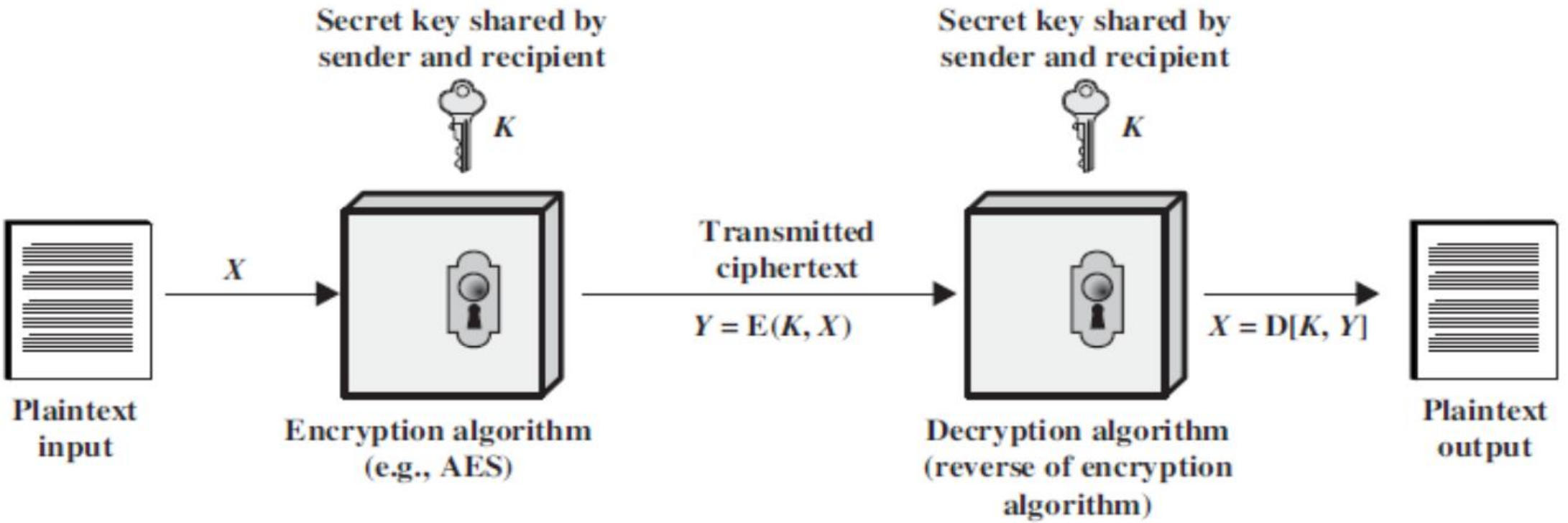
Introduction

- **Cryptology**
- Cryptology is a combination of Cryptography and Cryptanalysis.
- **Plain Text:** Clear text, or plain text, signifies a message that can be understood by the sender, the recipient, and also by anyone else who gets access to that message.
- **Cipher text:-**When a plain text message is codifies using any suitable scheme, the resulting message is called as cipher text.
- There are two types of techniques used to covert plain text to cipher text.
- **Substitution Techniques**
- **Transposition Techniques**

Conventional Cryptography

- **Conventional Cryptography/ encryption** is a cryptographic system that uses the same key used by the sender to encrypt the message and by the receiver to decrypt the message.
- It was the only type of encryption in use prior to the development of public-key encryption.
- It is still much preferred of the two types of encryption systems due to its simplicity.
- It is a relatively fast process since it uses a single key for both encryption and decryption
- In this encryption model, the sender encrypts plaintext using the receiver’s secret key, which can be later used by the receiver to decrypt the ciphertext.

Conventional Cryptography



Conventional Cryptography

- Here the original message, referred to as plaintext, is converted into apparently random nonsense, referred to as cipher text.
- The encryption process consists of an algorithm and a key.
- The key is a value independent of the plaintext.
- Changing the key changes the output of the algorithm.
- Once the cipher text is produced, it may be transmitted.
- Upon reception, the cipher text can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption.
- The security depends on several factors.
- First, the encryption algorithm must be powerful enough that it is impractical to decrypt a message on the basis of cipher text alone.
- Beyond that, the security depends on the secrecy of the key, not the secrecy of the algorithm.

Conventional Cryptography

- Two requirements for secure use of symmetric encryption:
- A strong encryption algorithm
- A secret key known only to sender / receiver
- Y = EK(X)
- X = DK(Y)
- assume encryption algorithm is known
- implies a secure channel to distribute key

Conventional Cryptography

- A source produces a message in plaintext, X = [X1, X2… XM] where M are the number of letters in the message.
- A key of the form K = [K1, K2… KJ] is generated.
- If the key is generated at the source, then it must be provided to the destination by means of some secure channel.
- With the message X and the encryption key K as input, the encryption algorithm forms the cipher text Y = [Y1, Y2, YN].
- This can be expressed as Y = EK(X)
- The intended receiver, in possession of the key , is able to invert the transformation: X = DK(Y)
- An opponent, observing Y but not having access to K or X, may attempt to recover X or K or both.
- It is assumed that the opponent knows the encryption and decryption algorithms.
- If the opponent is interested in only this particular message, then the focus of effort is to recover X by generating a plaintext estimate.
- Often if the opponent is interested in being able to read future messages as well, in which case an attempt is made to recover K by generating an estimate.

Stream Cipher

- A stream cipher encrypts text by applying a key and algorithm to each bit of a data stream one by one.
- Stream ciphers are mainly used to encrypt one byte (8 bits) at a time.
- Since stream ciphers are linear, messages are encrypted and decrypted with the help of the same key.
- And, while cracking them is difficult, hackers will have to manage to do it.
- In this, a keystream, a random series of bits, is generated from a key.
- To encrypt the data stream, each bit is XORed with an equivalent bit from the keystream.

Stream Cipher

- **How does it work?**
- Stream ciphers make use of a common key (symmetric key) to code their data.
- Encryption and decryption processes of the data are handled by this symmetric key.
- Unlike public-key ciphers, stream ciphers utilize one key for encryption as well as decryption, eliminating the need for different keys for each task (for instance, using one key to encrypt and another to decrypt).
- Cryptographic methods generally conceal data from unauthorized access by scrambling it.
- However, stream ciphers differ by processing data bit-by-bit, unlike block ciphers that operate on collections of data known as blocks.

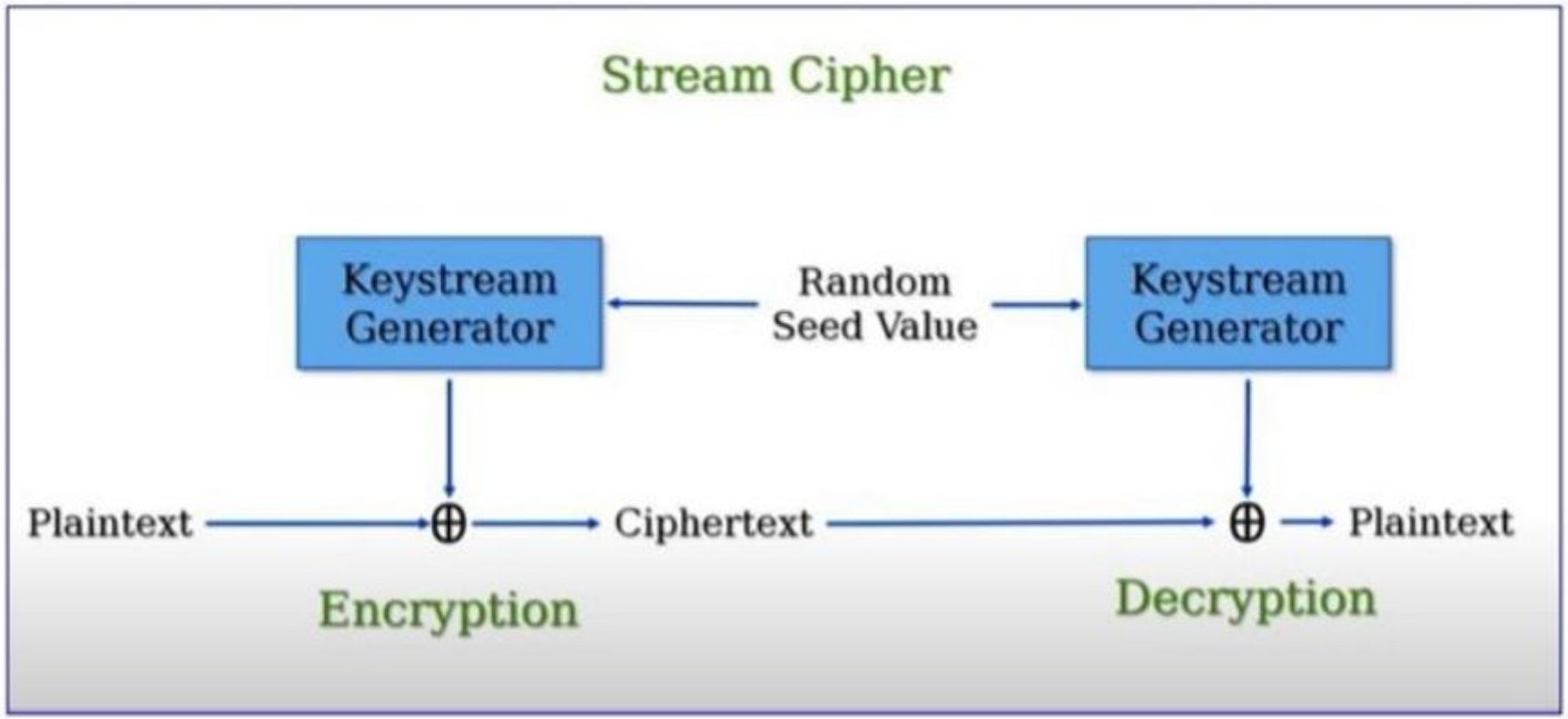
Stream Cipher

- Stream ciphers involve
- **Plaintext** − The original message to be encrypted.
- **Keystreams** − Random sequences of characters (e.g., numbers, letters, symbols) that replace the plaintext characters.
- **Ciphertext** − The encrypted message.
- Key generation is a complex mathematical operation, but modern computers can perform it quickly.

Stream Cipher

- In a stream cipher, individual bits of plaintext are inputted and subjected to a mathematical operation.
- The result is jumbled text which needs the right key to decode.
- Using the proper key, the receiver can reverse the process and convert the scrambled text back to its plaintext.
- In stream cipher encryption, the key known as a one-time pad is exceptionally secure due to its unique property.
- It is designed to be equivalent in length to the message being encrypted, ensuring that an attacker cannot mathematically decipher the message without having the original key.

Stream Cipher



Stream Cipher

- **Keystream Generation**
- Let us see at a basic example of keystream creation with the help of an XOR-based stream cipher.
- Let's say we have the below data −
- Key: 101011
- Initialization Vector (IV): 110100
- To create a stream of encrypted data (keystream):
- Set up the encryption algorithm with a secret key and initialization vector (IV).
- If needed, adjust the key and IV to be the same length as the message being encrypted.
- Combine the key and IV using an exclusive OR (XOR) operation to generate the keystream.

Stream Cipher

- Here is the step by step process −
- Key: 101011
- IV: 110100
- Keystream: 011111
- Now, let us say we have a plaintext message as: 1100101.
- To encrypt this plaintext using the keystream −
- Plaintext: 1100101
- Keystream: 011111
- Ciphertext: 1011010
- To decrypt the ciphertext, we would use the same keystream −
- Ciphertext: 1011010
- Keystream: 011111
- Plaintext: 1100101

Stream Cipher

- Popular Stream Ciphers
- **RC4** − Because it was quick and easy to use, RC4 was earlier widely used in the SSL/TLS and WEP/WPA protocols, but it was out of date due to security vulnerabilities.
- **Salsa20** − Salsa20, developed by Daniel J. Bernstein, is known for its efficiency and safety. Applications like secure communications and disk encryption frequently use it.
- **ChaCha** − ChaCha is a newer version of Salsa20, designed with better diffusion and protection against certain attacks. It's often used in protocols like TLS and VPNs.
- **HC-128** − Hongjun Wu's stream cipher offers excellent efficiency along with robust security. It works effectively in devices with limited capabilities, like cell phones.
- **Grain** − Martin Hell and Thomas Johansson developed this lightweight stream cipher. It is particularly efficient when implemented in hardware, making it ideal for use in applications such as RFID tags and sensor networks.

Substitution Cipher technique

- In the substitution-cipher technique, the each characters of a plain-text message are replaced by other characters, numbers or symbols.
- They are:
- Caesar Cipher
- Modified version of Caesar Cipher
- Monoalphabetic Cipher
- Polyalphabetic Cipher
- Homophonic Substitution Cipher
- Polygram Substitution Cipher
- Playfair Cipher
- Hill Cipher

Substitution Cipher technique

- **Caesar Cipher**
- Proposed by Julius Caesar.
- Mechanism to make a plaintext message into ciphertext message.
- It replacing each letter of the alphabet with the letter standing 3 places further down the alphabet.
- Example: Replace each A with D, B with E, etc.
- ABCDEFGHIJKLMNOPQRSTUVYZ
- DEFGHIJKLMNOPQRSTUVWXYZC
- Plaint Text: KIIT
- Cipher Text: NLLW

Substitution Cipher technique

- **Modified version of Caesar Cipher**
- The Caesar cipher is very simple and very easy to break.
- To make it complicated the modified version of Caesar cipher comes into play.
- Let us assume that the cipher-text alphabets corresponding to the original plain-text alphabets may not necessarily be three places down the order, but instead, can be any places down the order.
- As we know, the English language contains 26 alphabets.
- Thus, an alphabet A can be replaced by any other alphabet in the English alphabet set, (i.e. B through Z).
- Of course, it does not make sense to replace an alphabet by itself (i.e. replacing A with A).
- Thus, for each alphabet, we have 25 possibilities of replacement.
- Hence, to break a message in the modified version of Caesar cipher, our earlier algorithm would not work.

Substitution Cipher technique

- **Mono-alphabetic Cipher**
- A monoalphabetic cipher is a substitution cipher where a symbol in the plaintext has a one- to-one relationship with a symbol in the ciphertext.
- It means that a symbol in the plaintext is always replaced with the same symbol in the ciphertext, irrespective of its position in the plaintext.
- It uses random substitution.
- This means that in a given plain-text message, each A can be replaced by any other alphabet (B through Z), each B can also be replaced by any other random alphabet (A or C through Z), and so on.
- The crucial difference being, there is no relation between the replacement of B and replacement of A.
- That is, if we have decided to replace each A with D, we need not necessarily replace each B with E—we can replace each B with any other character!

Substitution Cipher technique

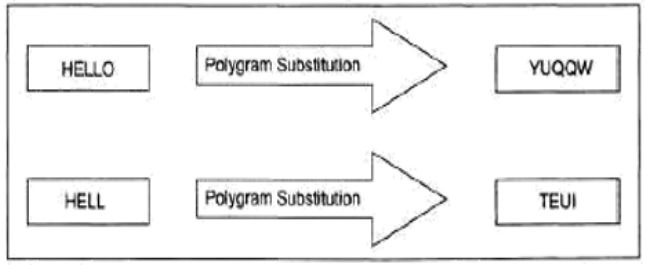
- **Polyalphabetic Substitution Cipher**
- Leon Battista invented the polyalphabetic substitution cipher in 1568.
- This cipher uses multiple one-character keys.
- Each of the keys encrypts one plain text character.
- The first key encrypts the first plain-text character; the second key encrypts the second plain-text character, and so on.
- After all the keys are used, they are recycled.
- Thus, if we have 30 one-letter keys, every 30th character in the plain text would be replaced with the same key.

Substitution Cipher technique

- **Homophonic Substitution Cipher**
- This substitution cipher is very similar to mono-alphabetic cipher.
- However, the difference between the two techniques is in homophonic substitution cipher, one plain-text alphabet can map to more than one cipher-text alphabet.
- For instance, A can be replaced by<D, H, P, R> ; B can be replaced by <E, I, Q, S>etc.
- Test using https://www.dcode.fr/homophonic-cipher

Substitution Cipher technique

- **Polygram Substitution Cipher**
- Polygram substitution cipher technique replaces one block of plain text with another block of cipher text—it does not work on a character-by-character basis.
- For instance, HELLO could be replaced by YUQQW, but HELL could be replaced by a totally different cipher text block TEUI ,as shown in Fig.
- This is true in spite of the first four characters of the two blocks of text (HELL) being the same.
- This shows that in the PolyGram substitution cipher, the replacement of plain text happens block by block, rather than character by character.



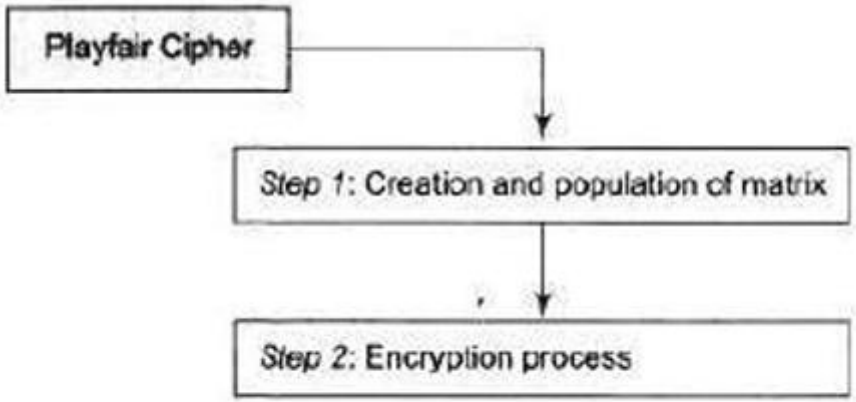
Polygram substitution

Substitution Cipher technique

- **Playfair Cipher**
- The Playfair cipher scheme was invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher.
- In Playfair cipher unlike traditional cipher we encrypt a pair of alphabets(digraphs) instead of a single alphabet.
- It was used for tactical purposes by British forces in the Second Boer War and in World War I and for the same purpose by the Australians during World War II.
- This was because Playfair is reasonably fast to use and requires no special equipment.
- The Playfair encryption scheme uses two main processes.

Substitution Cipher technique

- The Playfair encryption scheme uses two main processes.
- **Creation and population of matrix**
- **Encryption process**



Substitution Cipher technique

- **Step 1: Creation and Population of Matrix**
- The Playfair cipher makes use of a 5 x 5 matrix (table), which is used to store a keyword or phrase that becomes the key for encryption and decryption.
- The way this is entered into the 5 x 5 matrix is based on some simple rules:
- 1. Enter the keyword in the matrix row-wise: left-to-right, and then top-to-bottom.
- 2. Drop duplicate letters.
- 3. Fill the remaining spaces in the matrix with the rest of the English alphabets (A-Z) that were not a part of our keyword.
- While doing so, combine I and J in the same cell of the table.
- In other words, if I or J is a part of the keyword, disregard both I and J while filling the remaining slots.

Substitution Cipher technique

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

Fig: Creation and Population of matrix

Substitution Cipher technique

- **EXAMPLE OF ENCRYPTION AND DECRYPTION IN PLAYFAIR**
- For example, suppose that our keyword=TUTORIALS
- Then, the 5 x 5 matrix containing our keyword will look as shown
- Let us say, our Plaintext= HIDE MONEY

Substitution Cipher technique

Process of Playfair Cipher

- First, a plaintext message is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter. Let us say we want to encrypt the message hide money. It will be written as −

  HI DE MO NE YZ

- The rules of encryption are −

  - If both the letters are in the same column, take the letter below each one (going back to the top if at the bottom)

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

H and I are in same column, hence take letter below them to replace. HI → QC

Substitution Cipher technique

- If both letters are in the same row, take the letter to the right of each one (going back to the left if at the farthest right)

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

D and E are in same row, hence take letter to the right of them to replace. DE → EF

- If neither of the preceding two rules are true, form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

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

'M' and 'O' nor on same column or same row, hence form rectangle as shown, and replace letter by picking up opposite corner letter on same row
MO -> NU

Substitution Cipher technique

Using these rules, the result of the encryption of hide money with the key of tutorials would be –

QC EF NU MF ZV

Decrypting the Playfair cipher is as simple as doing the same process in reverse. Receiver has the same key and can create the same key table, and then decrypt any messages made using that key.

Substitution Cipher technique

- **Hill Cipher**
- The Hill cipher works on multiple letters at the same time.
- Lester Hill invented this in 1929.
- The Hill cipher uses the matrix theory of mathematics.
- Working:
- Treat each letter with a number like A=0, B=1, C=2…… Z=25. • Let us say, our original message is “TAJ”
- As per the rule, T=19 A=0 J=9
- Convert it into matrix form as:

$$\begin{bmatrix} 19 \\ 0 \\ 9 \end{bmatrix}$$

Substitution Cipher technique

- Now multiply the plain text matrix with any number as keys.
- The multiplying matrix should be of n x n where n is the number of rows of original matrix

$$\begin{bmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{bmatrix} \times \begin{bmatrix} 19 \\ 0 \\ 9 \end{bmatrix} = \begin{bmatrix} 123 \\ 337 \\ 515 \end{bmatrix}$$

- Now compute mod 26 on resultant matrix i.e. take the remainder after dividing by 26.

$$\begin{bmatrix} 123 \\ 337 \\ 515 \end{bmatrix} \bmod 26 = \begin{bmatrix} 19 \\ 25 \\ 21 \end{bmatrix}$$

Substitution Cipher technique

- Now translating numbers into alphabets, we get: 19=T 25= Z 21=V
- Therefore our cipher text is TZV
- To decrypt hill cipher, follow the steps:
- 1.) Take cipher text matrix and multiply it by inverse of original key matrix
- 2.) Again perform mod by 26.
- Thus we get our original text.

Transposition Cipher technique

- **Transposition techniques**
- Transposition technique is an encryption method which is achieved by performing permutation over the plain text.
- **Rail-Fence Technique**
- This technique is a type of Transposition technique which involves writing the plain text as a sequence of diagonals and then reading row-by-row to produce cipher text.
- It uses a simple algorithm.
- 1. Writing down the plaintext message into a sequence of diagonals.
- 2. Read the plain text in step-1 as a sequence of rows.

Transposition Cipher technique

- Example:
- Plain Text: meet me Tomorrow
- Now, we will write this plain text sequence wise in a diagonal form as you can see below:

- Cipher Text: m e m t m r o e t e o o r w

Transposition Cipher technique

- **Simple Columnar Transposition Technique**
- **A. Basic Technique**
- It is a slight variation to the Rail-fence technique, let’s see its algorithm:
- 1. In a rectangle of pre-defined size, write the plain-text message row by row.
- 2. Read the plain message in random order in a column-wise fashion.
- It can be any order such as 2, 1, 3 etc.
- 3. Thus Cipher-text is obtained.

Transposition Cipher technique

- Example: Original message: "INCLUDEHELP IS AWESOME".
- Now we apply the above algorithm and create the rectangle of 4 column (we decide to make a rectangle with four column it can be any number.)

Column 1	Column 2	Column 3	Column 4
I	N	C	L
U	D	E	H
E	L	P	I
S	A	W	E
S	O	M	E

- Now let’s decide on an order for the column as 4, 1, 3 and 2 and now we will read the text in column-wise.
- Cipher-text of round 1: LHIEEIUESSCEPWMNDLAO

Transposition Cipher technique

- Round 2:

Column 1	Column 2	Column 3	Column 4
L	H	I	E
E	I	U	E
S	S	C	E
P	W	M	N
D	L	A	O

- Now, we decide to go with a previous order that is 4,1,3,2.
- Cipher-text: EEENOLESPDIUCMAHISWL
- These multi-round columnar techniques are harder to crack as compared to methods seen earlier.

Transposition Cipher technique

- **Vigenere Cipher**
- is a method of encrypting alphabetic text.
- It uses a simple form of polyalphabetic substitution.
- The encryption of the original text is done using the Vigenère square or Vigenère table.
- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

Transposition Cipher technique

- **Example:**

```
Input : Plaintext :   GEEKSFORGEEKS
              Keyword :  AYUSH
Output : Ciphertext :  GCYCZFMLYLEIM
For generating key, the given keyword is repeated
in a circular manner until it matches the length of
the plain text.
The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"
The plain text is then encrypted using the process
explained below.
```

Transposition Cipher technique

- **Encryption:**
- The first letter of the plaintext, G is paired with A, the first letter of the key.
- So use row G and column A of the Vigenère square, namely G.
- Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E, and column Y is C.
- The rest of the plaintext is enciphered in a similar fashion.

Transposition Cipher technique

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

Transposition Cipher technique

- **Decryption:**
- Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column's label as the plaintext.
- For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter.
- Next, we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.

Transposition Cipher technique

- A more easy implementation could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

Encryption

The plaintext(P) and key(K) are added modulo 26.

$E_i = (P_i + K_i) \bmod 26$

Decryption

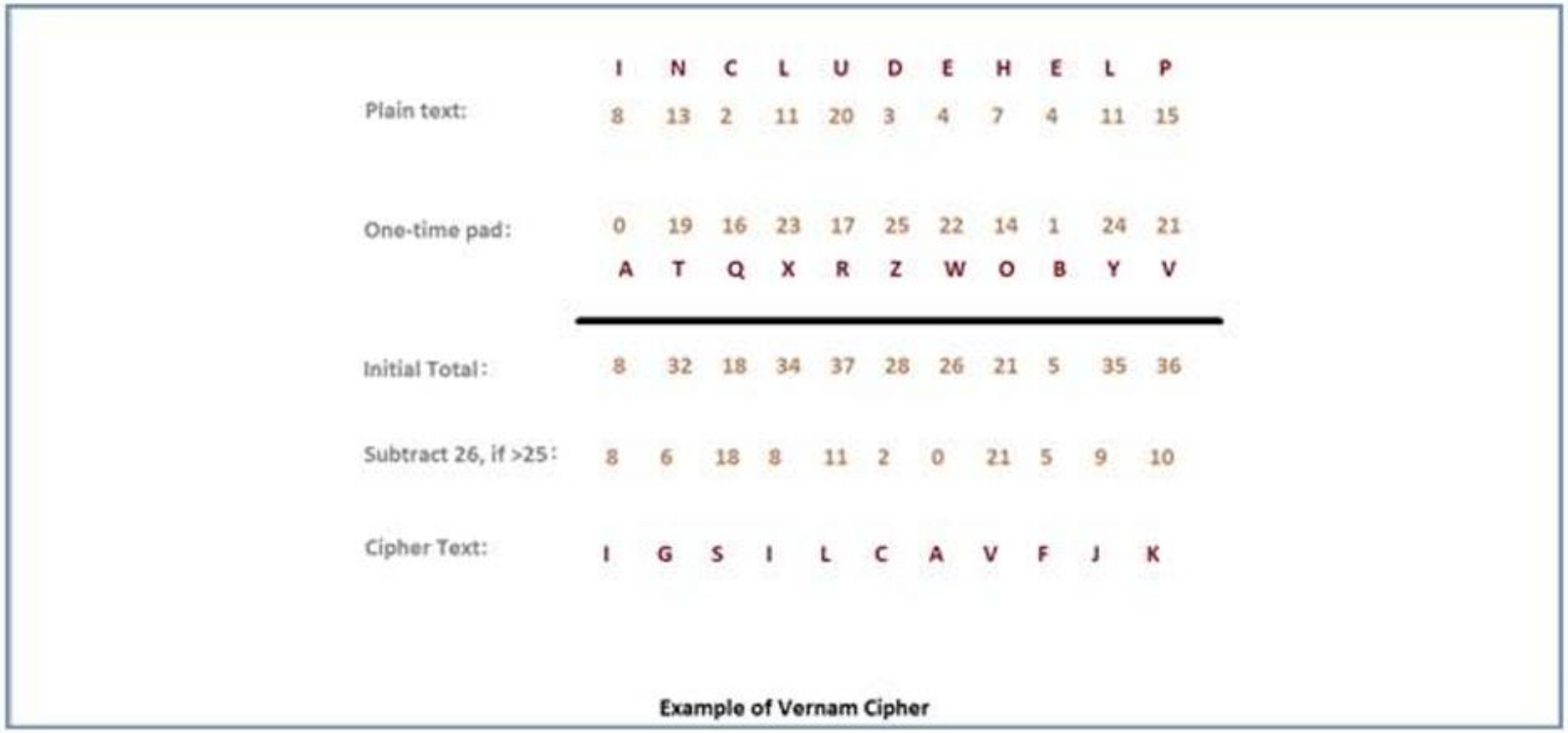
$D_i = (E_i - K_i) \bmod 26$

Note: D_i denotes the offset of the i-th character of the plaintext. Like offset of **A** is 0 and of **B** is 1 and so on.

Transposition Cipher technique

- **Vernam Cipher (one time pad)**
- The Vernam Cipher has a specific subset one-time pad, which uses input ciphertext as a random set of non-repeating character.
- The thing to notice here is that, once an input cipher text gets used it will never be used again hence one-time pad and length of cipher-text is the size that of message text.
- Algorithm:
- 1. Plain text character will be represented by the numbers as A=0, B=1, C=2,... Z=25.
- 2. Add each corresponding number of a plain text message to the input cipher text alphabet numbers.
- 3. If the sum is greater than or equal to 26, subtract 26 from it.
- 4. Translate each number back to corresponding letters and we got our cipher text.

Transposition Cipher technique



	I	N	C	L	U	D	E	H	E	L	P
Plain text:	8	13	2	11	20	3	4	7	4	11	15
One-time pad:	0	19	16	23	17	25	22	14	1	24	21
	A	T	Q	X	R	Z	W	O	B	Y	V
Initial Total:	8	32	18	34	37	28	26	21	5	35	36
Subtract 26, if >25:	8	6	18	8	11	2	0	21	5	9	10
Cipher Text:	I	G	S	I	L	C	A	V	F	J	K

Example of Vernam Cipher

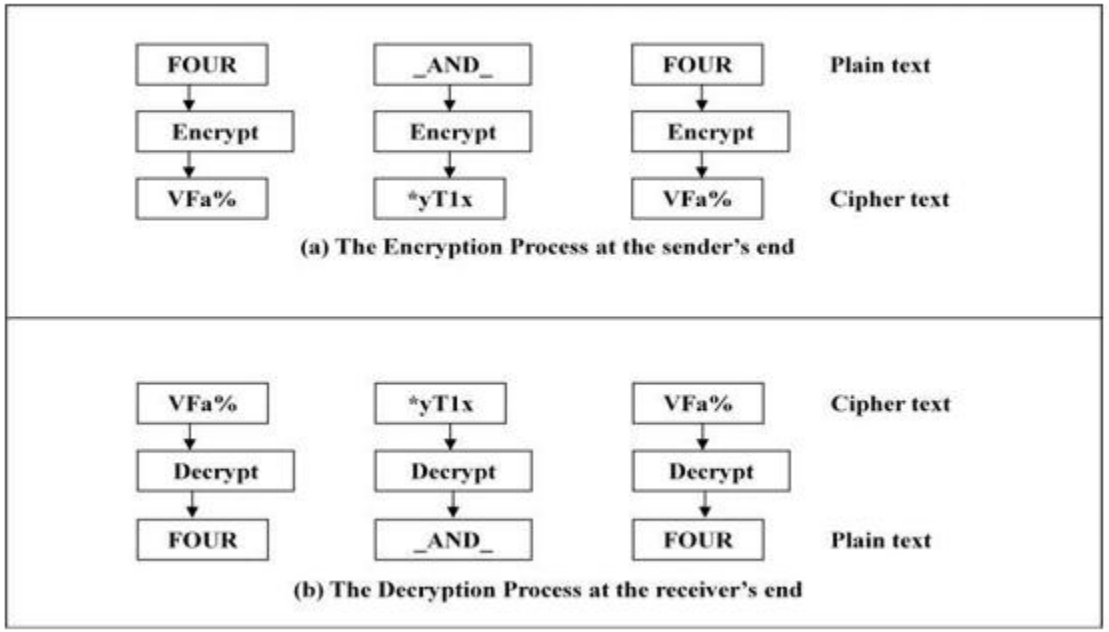
- One time pad should be discarded after every single use and this technique is proved highly secure and suitable for small messages but illogical if used for long messages.

Block Cipher technique

- **Block Cipher**
- Block-by-block encryption / decryption.
- In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of cipher text bits.
- The number of bits in a block is fixed.
- For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

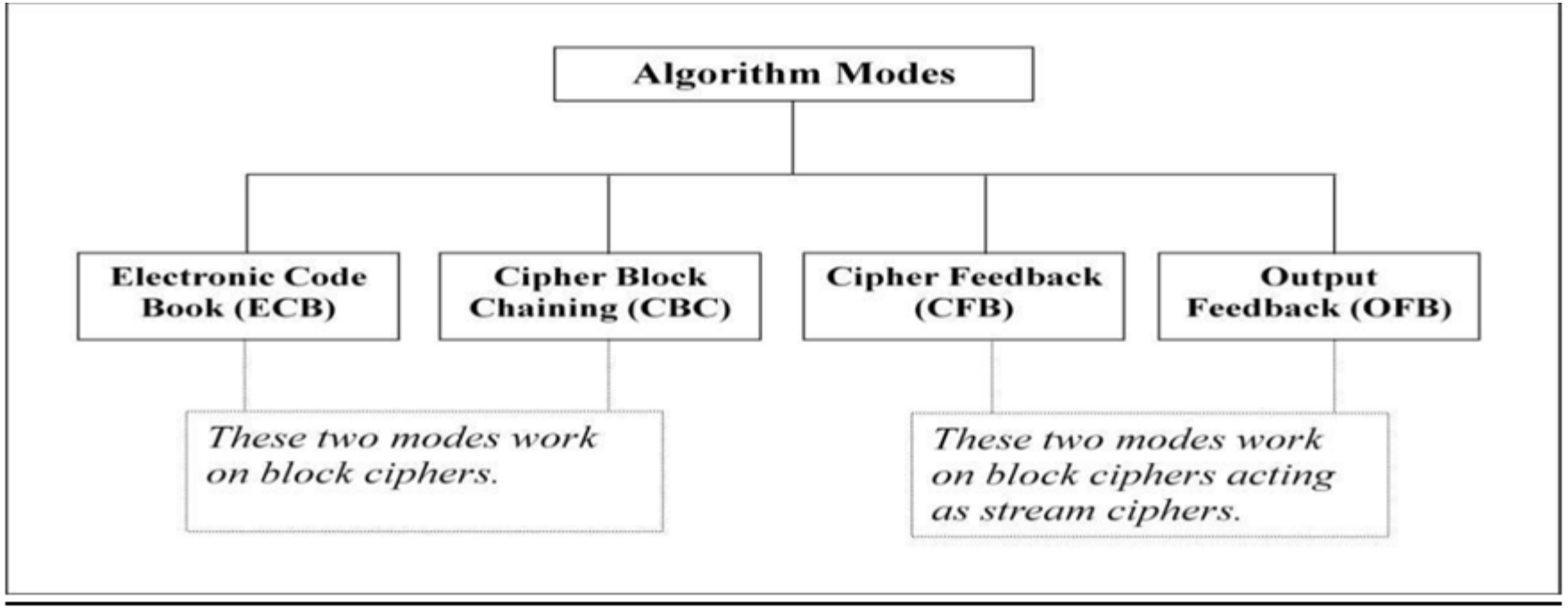
Block Cipher technique

- The basic scheme of a block cipher is given as follows:
- Block Cipher Example: Suppose we have a plain text “FOUR_AND _FOUR” that needs to be encrypted.
- By using this technique FOUR could be encrypted first followed by _AND_ and FOUR.



Block Cipher technique

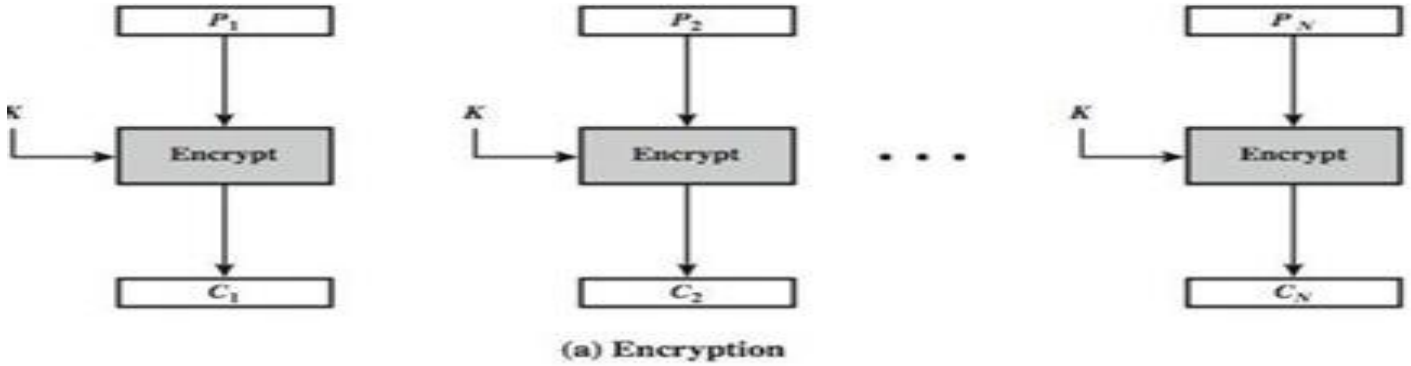
- **Algorithm Modes:**
- It is a combination of series of basic algorithm steps on block cipher and some sort of feedback from the previous steps.
- It is divided into four modes:



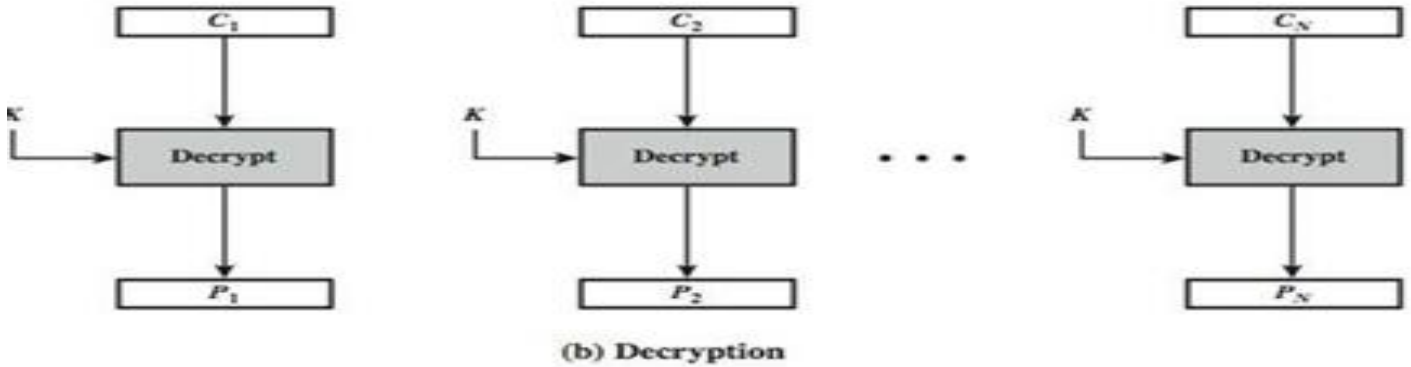
Block Cipher technique

- **Electronic Code book (ECB) Mode**
- ECB is a simplest and straightforward method of converting a block of plaintext into cipher text.
- Here, plain-text message is divided into blocks of 64 bits each.
- Each such block is then encrypted independently of the other blocks.
- For all blocks in a message, the same key is used for encryption.
- This encryption process is shown figure.
- At the receiver’s end, the incoming data is divided into 64-bit blocks.
- By using the same key as was used for encryption, each block is decrypted to produce the corresponding plain-text block.
- This decryption process is shown figure.

Block Cipher technique



(a) Encryption



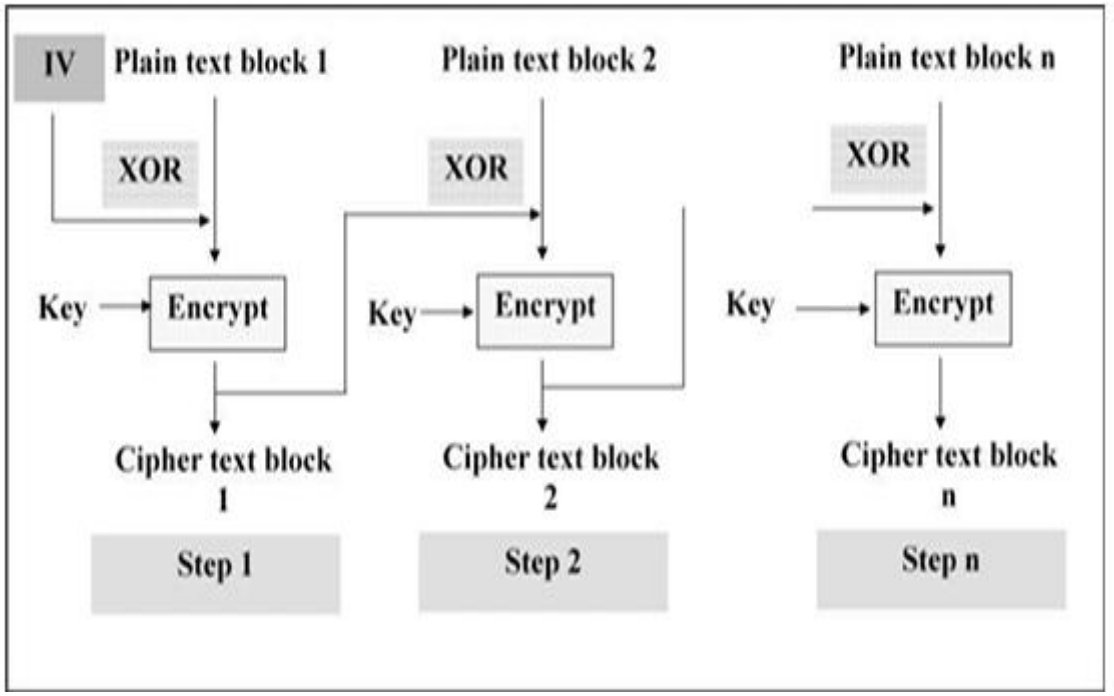
(b) Decryption

Block Cipher technique

- **Cipher Block Chaining (CBC) Mode**
- In CBC mode, a feedback mechanism is used.
- Chaining adds a feedback mechanism to a block cipher.
- In Cipher Block Chaining (CBC), the results of the encryption of the previous block are fed back into the encryption of the current block.
- That is, each block is used to modify the encryption of the next block.
- Thus, each block of cipher text is dependent on the corresponding current input plain-text block, as well as all the previous plain-text blocks.

Block Cipher technique

- **Operation:**
- Load the n-bit Initialization Vector (IV).
- IV is a random generated block of text in a register.
- XOR the n-bit plain text block with data va in IV register.
- Encrypt the result of XOR operation with the key K.
- Result is it produce the cipher text block.
- Feed cipher text block into the IV register and continue the operation till all plaintext blocks are processed.



Block Cipher technique

- **Cipher Feedback (CFB) Mode**
- Not all applications can work with blocks of data.
- Security is also required in applications that are character-oriented.
- For instance, an operator can be typing keystrokes at a terminal, which needs to be immediately transmitted across the communications link in a secure manner, i.e., by using encryption.
- In such situations, stream cipher must be used.
- The Cipher Feedback (CFB) mode is useful in such cases.
- In this mode, data is encrypted in units that are smaller (e.g., they could be of size 8 bits, i.e. the size of a character typed by an operator) than a defined block size (which is usually 64 bits).

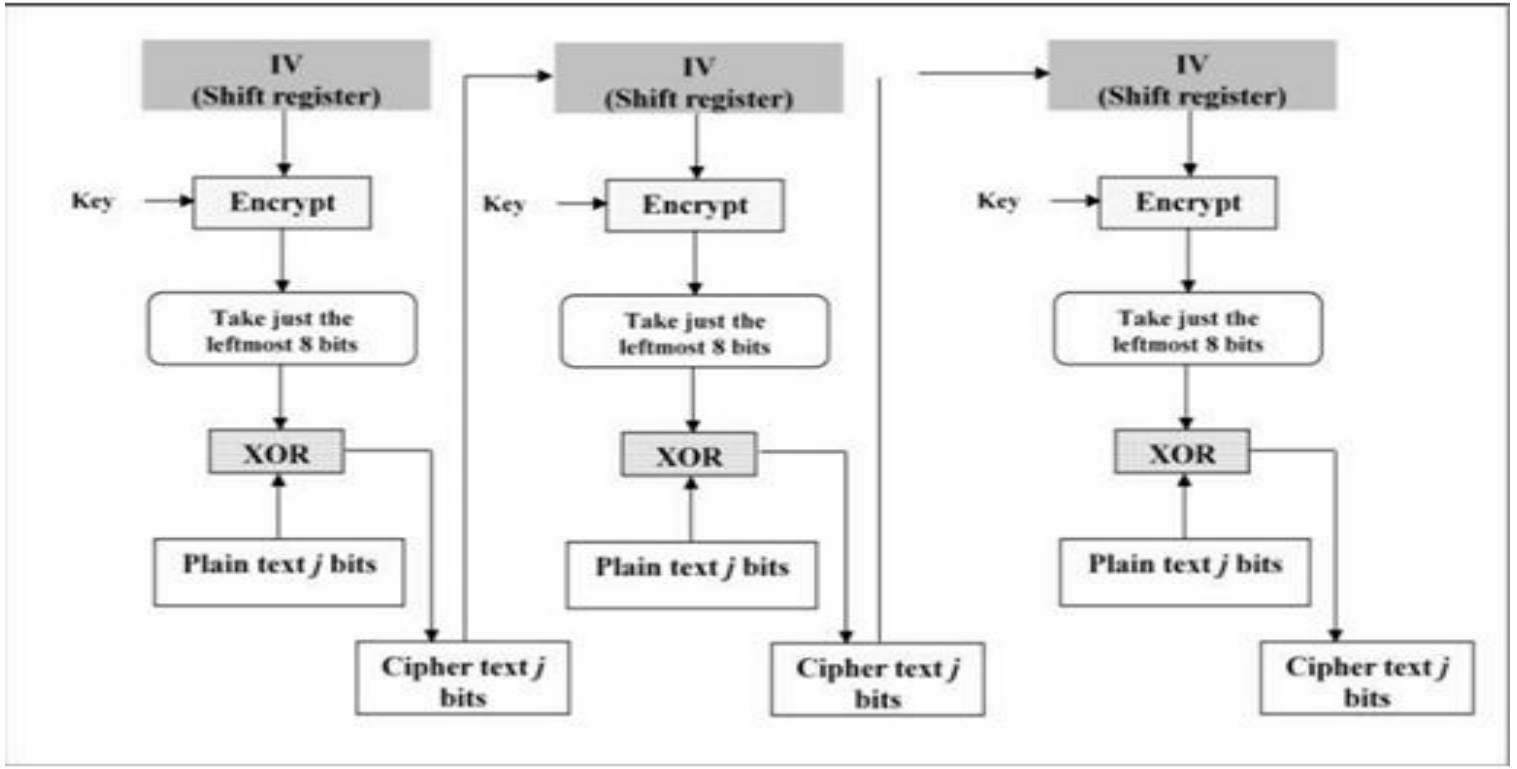
Block Cipher technique

- Steps of operation are:
- Assuming that we are dealing with j bits at a time (as we have seen usually, but not always, j = 8).
- Step 1: Like CBC, a 64-bit Initialization Vector (IV) is used in the case of CFB mode.
- The IV is kept in a shift register.
- It is encrypted in the first step to produce a corresponding 64 bit cipher text.
- Step 2 : Now, the leftmost (i.e. the most significant) j bits of the encrypted IV are XORed with the first j bits of the plain text.
- Step 3 :Now, the bits of IV (i.e. the contents of the shift register containing IV) are shifted left by j positions.
- Thus, the rightmost j positions of the shift register now contain unpredictable data. These rightmost j positions are now filled with C.
- Step 4 :Now, steps 1 through 3 continue until all the plain text units are encrypted.

Block Cipher technique

- That is, the following steps are repeated:
- IV is encrypted.
- The leftmost j bits resulting from this encryption process are XORed with the next j bits of the plain text.
- The resulting cipher-text portion (i.e., the next j bits of cipher text) is sent to the receiver.
- The shift register containing the IV is left shifted by j bits.
- The j bits of the cipher text are inserted from right into the shift register containing the IV.

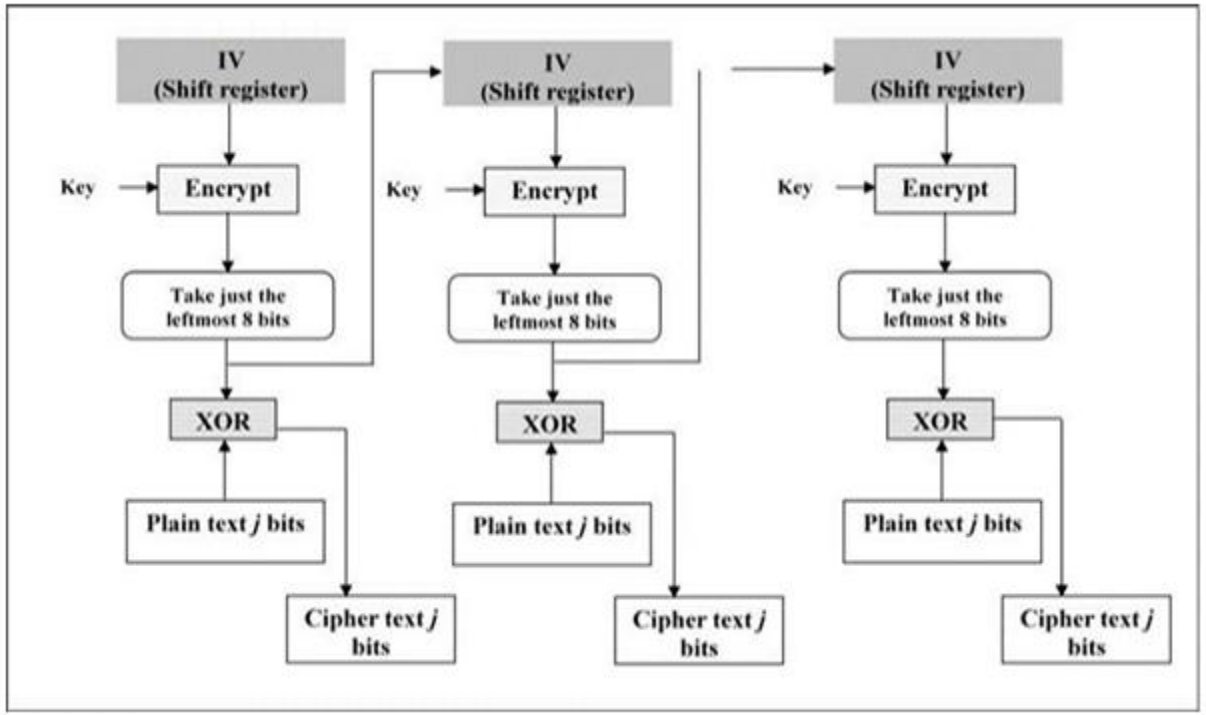
Block Cipher technique



Block Cipher technique

- **Output Feedback (OFB) Mode**
- The OFB mode is similar to CFB, but the only difference is that in CFB, the cipher text is fed into the next stage of encryption process.
- But in case of OFB the output of IV encryption process is fed into the next stage of encryption process.
- In this mode, if there are errors in individual bits, they remain errors in individual bits and do not corrupt the whole message.
- That is, bit errors do not get propagated.

Block Cipher technique



Feistel Cipher Structure

- The Feistel Cipher is not a complete cipher itself, but a design model used to build many block ciphers, such as DES.
- It provides a simple way to build secure encryption and decryption algorithms.
- The only difference: during decryption, the round keys are applied in reverse order.
- **Step-by-Step Algorithm**
- Here’s how a simple 2-round Feistel Cipher works:
- **1. Convert Plaintext to Binary**
- Take the plaintext characters.
- Convert each character to its ASCII value and then into an 8-bit binary string.

Feistel Cipher Structure

- **2. Divide into Two Halves**
- Split the binary string into two equal halves:
- Left half = L1
- Right half = R1
- **3. Generate Round Keys**
- Create random binary keys K1 and K2.
- Each key has a length equal to half the block size.

Feistel Cipher Structure

- **Round 1 (Encryption)**
- Compute function f1:
  f1 = R1 XOR K1
- Update halves:
  R2 = L1 XOR f1
  L2 = R1
- **Round 2 (Encryption)**
- Compute function f2:
  f2 = R2 XOR K2
- Update halves:
  R3 = L2 XOR f2
  L3 = R2
- **Final Ciphertext**
- Concatenate L3 and R3 to get the ciphertext.

Feistel Cipher Structure

- **Decryption Process**
- The Feistel Cipher uses the same algorithm for both encryption and decryption, with the round keys simply applied in reverse order.
- **Example:**
- Plaintext: Hello
- After encryption → Ciphertext: E1!w(
- After decryption → Retrieved Plaintext: Hello
- Plaintext: Geeks
- After encryption → Ciphertext: O;Q
- After decryption → Retrieved Plaintext: Geeks

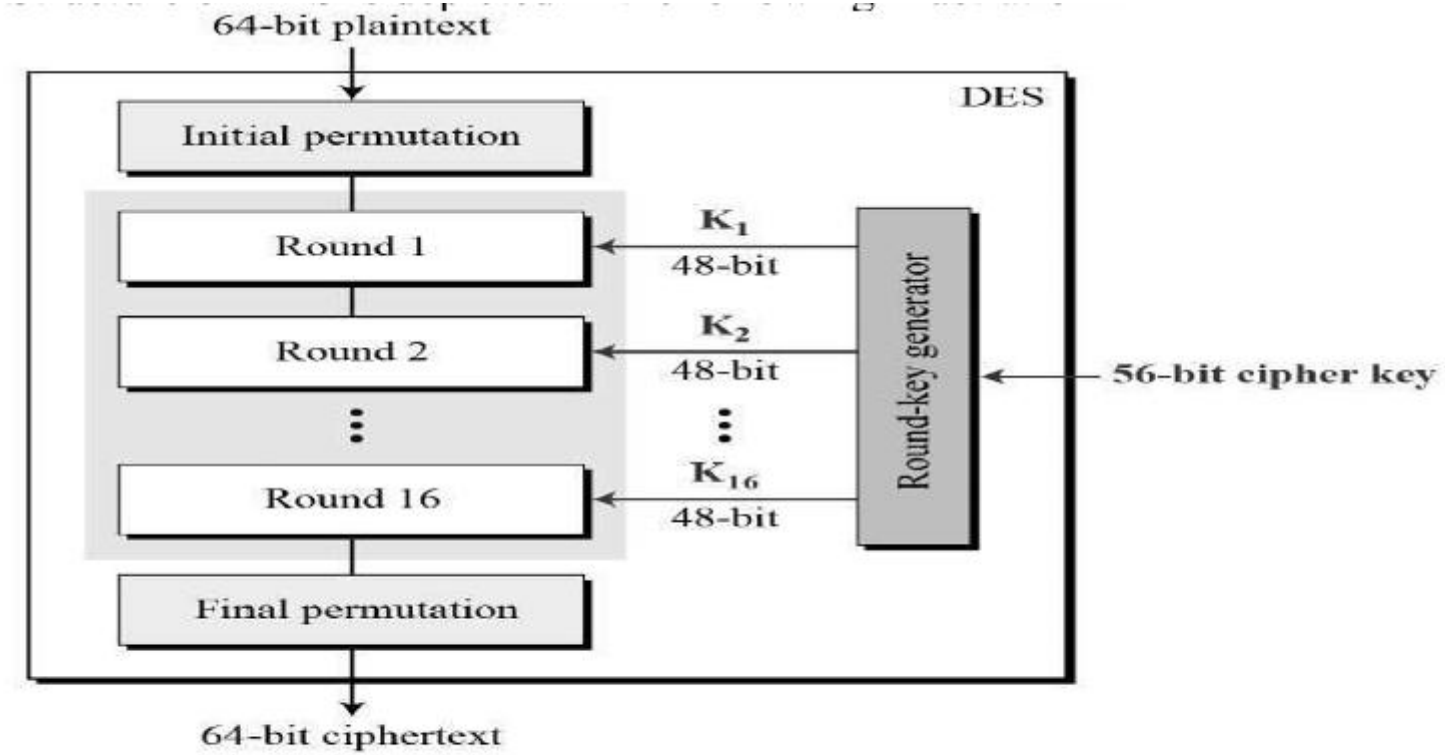
Data Encryption Standard(DES)

- **Data Encryption Standard**
- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
- DES is an implementation of a Feistel Cipher.
- It uses 16 round Feistel structure.
- The block size is 64-bit.
- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

Data Encryption Standard(DES)

- **How DES Works?**
- DES is an implementation of a Feistel Cipher.
- It uses 16 round Feistel structure.
- The block size is 64-bit.
- Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).
- General Structure of DES is depicted in the following illustration −

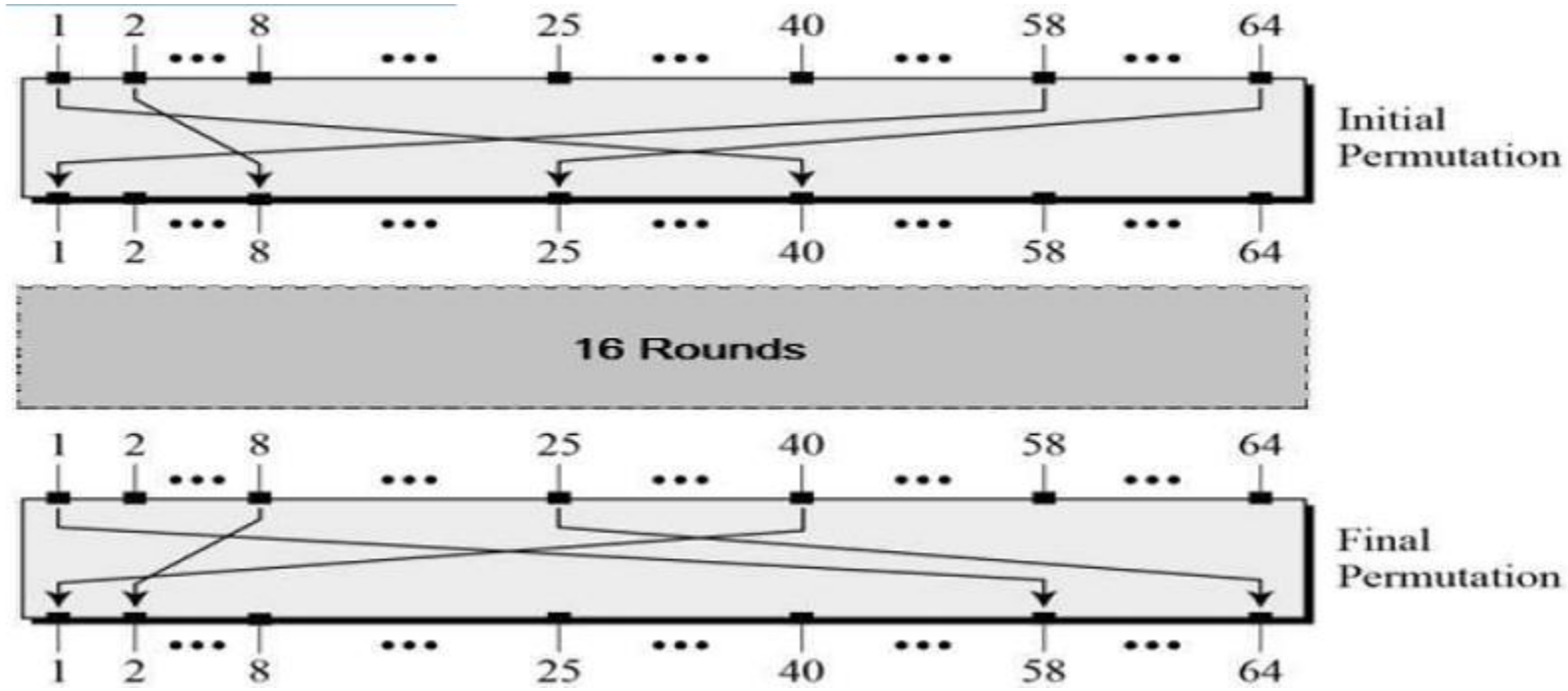
Data Encryption Standard(DES)



Data Encryption Standard(DES)

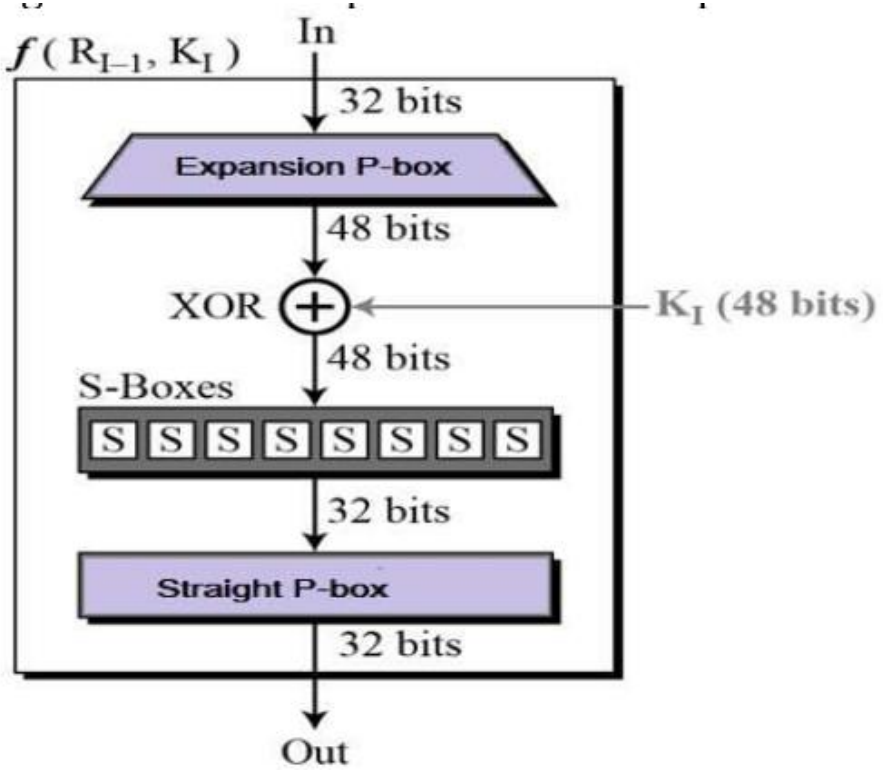
- Since DES is based on the Feistel Cipher, all that is required to specify DES is
- **Round function**
- **Key schedule**
- **Any additional processing − Initial and final permutation Initial and Final Permutation**
- The initial and final permutations are straight Permutation boxes (P-boxes) that are inverses of each other.
- They have no cryptography significance in DES.
- The initial and final permutations are shown as follows −

Data Encryption Standard(DES)



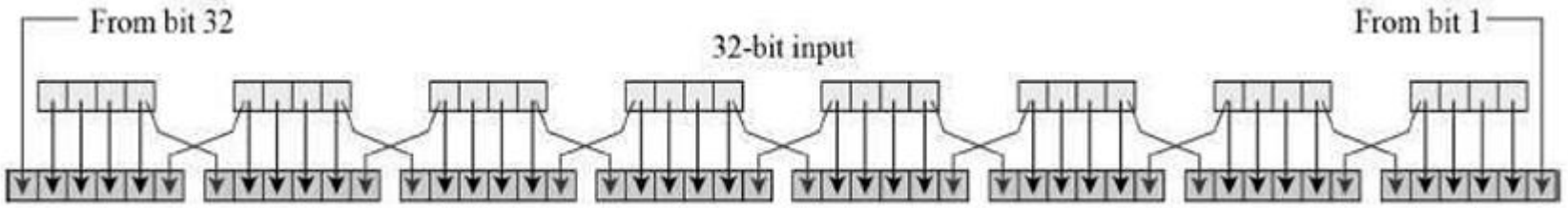
Data Encryption Standard(DES)

- **Round Function**
- The heart of this cipher is the DES function, f.
- The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Data Encryption Standard(DES)

- **Expansion Permutation Box**
- Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits.
- Permutation logic is graphically depicted in the following illustration −



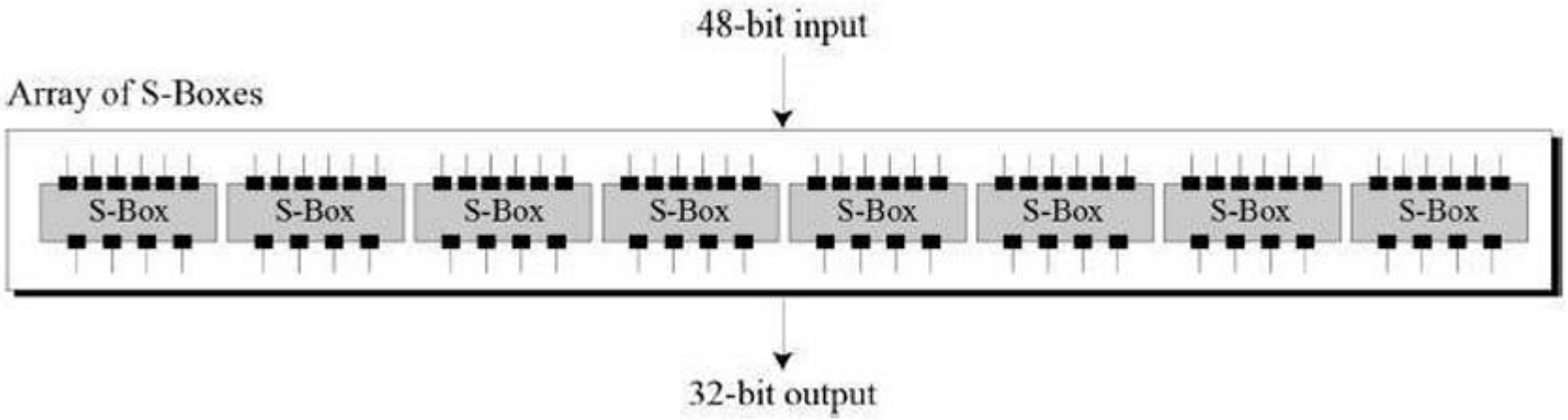
Data Encryption Standard(DES)

- The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Data Encryption Standard(DES)

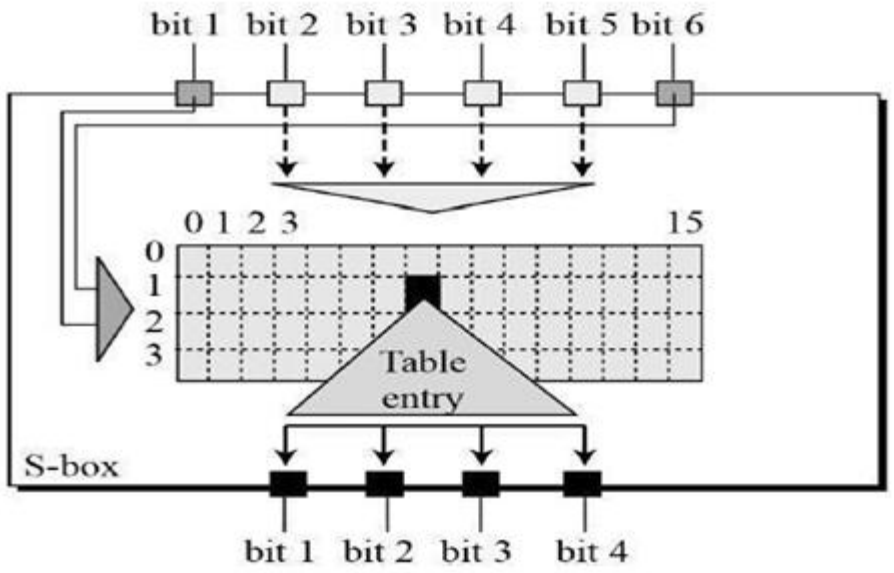
- **XOR (Whitener)**
- After the expansion permutation, DES does XOR operation on the expanded right section and the round key.
- The round key is used only in this operation.
- **Substitution Boxes**
- The S-boxes carry out the real mixing (confusion).
- DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



Data Encryption Standard(DES)

- There are a total of eight S-box tables.
- The output of all eight s-boxes is then combined in to 32 bit section.

The S-box rule is illustrated below –



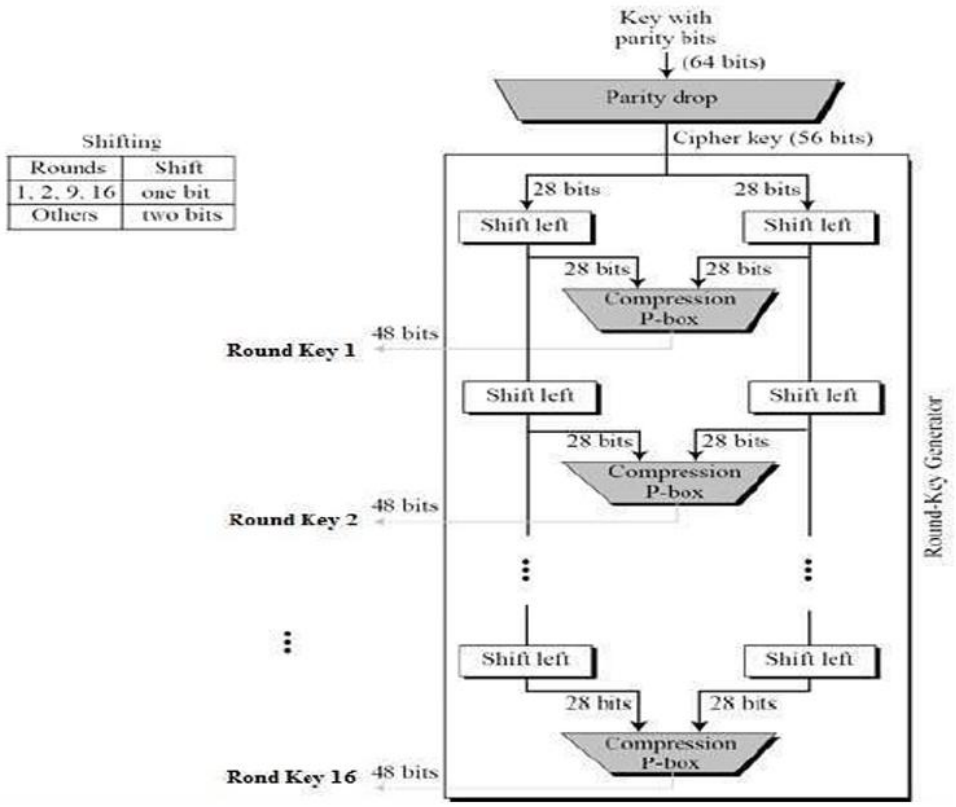
Data Encryption Standard(DES)

- **Straight Permutation**
- The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- **Key Generation**
- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.
- The process of key generation is depicted in the following illustration

Data Encryption Standard(DES)



The logic for Parity drops, shifting, and Compression P-box is given in the DES description.

Data Encryption Standard(DES)

- **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 cipher text depends on many bits of plaintext.
- These keys shall be avoided.
- DES has proved to be a very well designed block cipher.
- There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

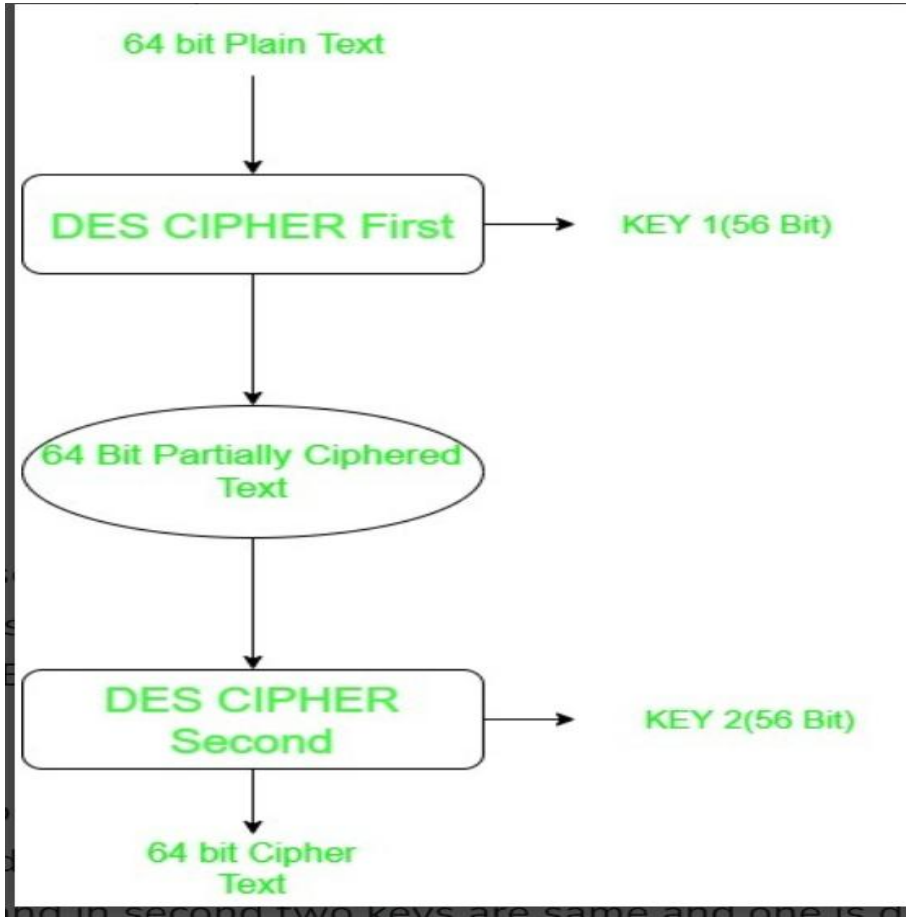
Double DES and Triple DES

- As we know the Data encryption standard (DES)uses 56 bit key to encrypt any plain text which can be easily be cracked by using modern technologies.
- To prevent this from happening double DES and triple DES were introduced which are much more secured than the original DES because it uses 112 and 168 bit keys respectively.
- They offer much more security than DES.

Double DES and Triple DES

- **Double DES**
- Double DES is a encryption technique which uses two instance of DES on same plain text.
- In both instances it uses different keys to encrypt the plain text. Both keys are required at the time of decryption.
- The 64 bit plain text goes into first DES instance which then converted into a 64 bit middle text using the first key and then it goes to second DES instance which gives 64 bit cipher text by using second key.
- However double DES uses 112 bit key but gives security level of 2^56 not 2^112 and this is because of meet-in-the middle attack which can be used to break through double DES.

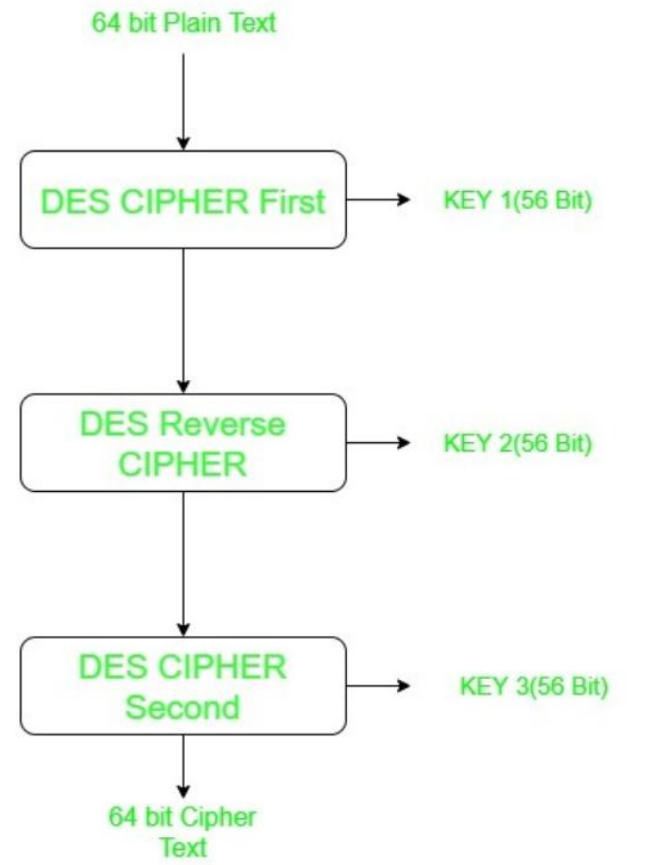
Double DES and Triple DES



Double DES and Triple DES

- **Triple DES**
- Triple DES is a encryption technique which uses three instance of DES on same plain text.
- It uses there different types of key choosing technique in first all used keys are different and in second two keys are same and one is different and in third all keys are same.
- Triple DES is also vulnerable to meet-in-the middle attack because of which it give total security level of 2^112 instead of using 168 bit of key.
- The block collision attack can also be done because of short block size and using same key to encrypt large size of text. It is also vulnerable to sweet32 attack.
-

Double DES and Triple DES



Advanced Encryption Standard (AES)

- **Advanced Encryption Standard (AES)**
- is a highly trusted encryption algorithm used to secure data by converting it into an unreadable format without the proper key.
- It is developed by the National Institute of Standards and Technology (NIST) in 2001.
- It is is widely used today as it is much stronger than DES and triple DES despite being harder to implement.
- AES encryption uses various key lengths (128, 192, or 256 bits) to provide strong protection against unauthorized access.
- This data security measure is efficient and widely implemented in securing internet communication, protecting sensitive data, and encrypting files. AES, a cornerstone of modern cryptography, is recognized globally for its ability to keep information safe from cyber threats.

Advanced Encryption Standard (AES)

- AES is a Block Cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.
- That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text.
- AES relies on the substitution-permutation network principle, which is performed using a series of linked operations that involve replacing and shuffling the input data.

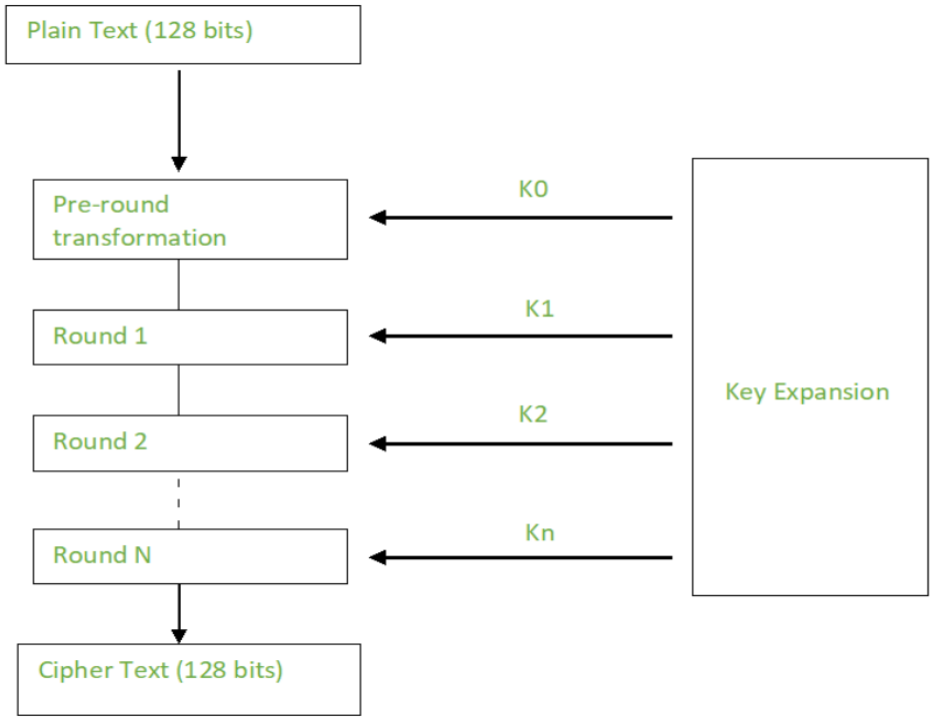
Advanced Encryption Standard (AES)

- **Working of The Cipher**
- AES performs operations on bytes of data rather than in bits.
- Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.
- The number of rounds depends on the key length as follows :

N (Number of Rounds)	Key Size (in bits)
10	128
12	192
14	256

Advanced Encryption Standard (AES)

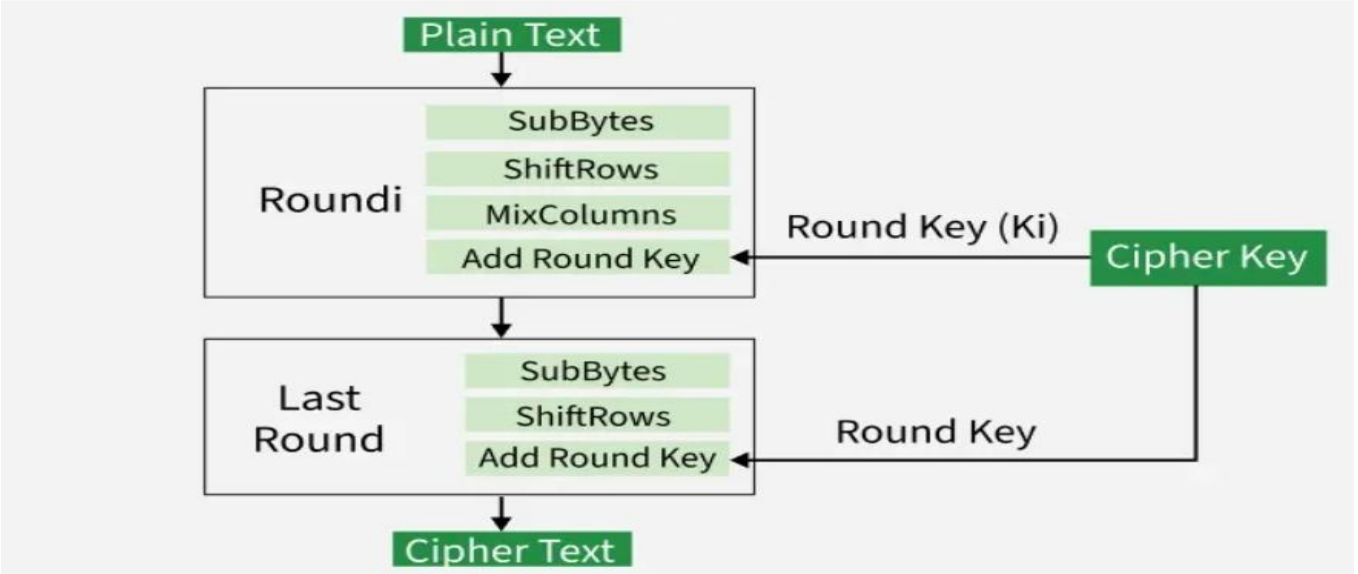
- **Creation of Round Keys**
- A Key Schedule algorithm calculates all the round keys from the key.
- So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



Advanced Encryption Standard (AES)

- **Encryption**
- AES considers each block as a 16-byte (4 byte x 4 byte = 128) grid in a column-major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```



Advanced Encryption Standard (AES)

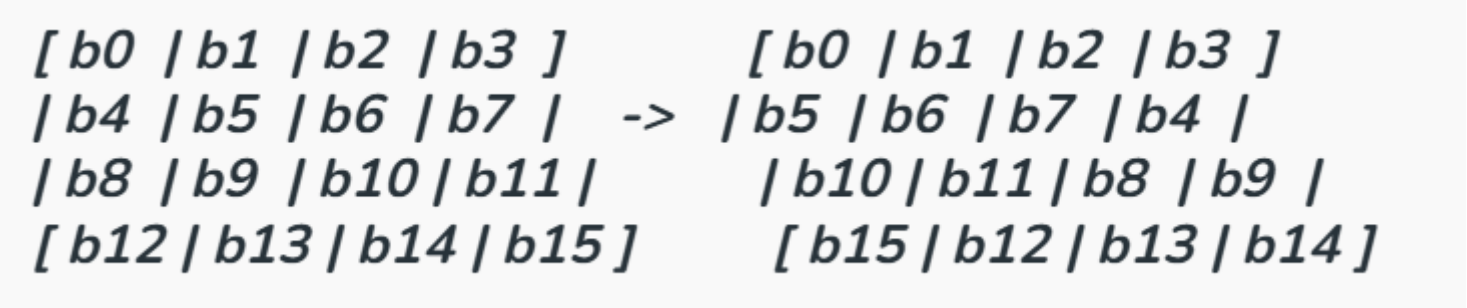
- **Each round comprises of 4 steps :**
- Sub Bytes
- Shift Rows
- Mix Columns
- Add Round Key

Advanced Encryption Standard (AES)

- **Step1: Sub Bytes**
- This step implements the substitution.
- In this step, each byte is substituted by another byte. It is performed using a lookup table also called the S-box.
- This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte.
- The result of this step is a 16-byte (4 x 4) matrix like before.
- The next two steps implement the permutation.

Advanced Encryption Standard (AES)

- **Step2: Shift Rows**
- This step is just as it sounds. Each row is shifted a particular number of times.
- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.
- (A left circular shift is performed.)



Advanced Encryption Standard (AES)

- **Step 3: Mix Columns**
- This step is a matrix multiplication.
- Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.
- This step is skipped in the last round.

$$\begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix}$$

Advanced Encryption Standard (AES)

- **Step 4: Add Round Keys**
- Now the resultant output of the previous stage is XOR-ed with the corresponding round key.
- Here, the 16 bytes are not considered as a grid but just as 128 bits of data.
- After all these rounds 128 bits of encrypted data are given back as output.
- This process is repeated until all the data to be encrypted undergoes this process.

Advanced Encryption Standard (AES)

- **Decryption**
- The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.
- Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.
- The stages of each round of decryption are as follows :
- **Add round key**
- **Inverse Mix Columns**
- **Shift Rows**
- **Inverse Sub Byte**
- The decryption process is the encryption process done in reverse so I will explain the steps with notable differences.

Advanced Encryption Standard (AES)

- **Inverse Mix Columns**
- This step is similar to the Mix Columns step in encryption but differs in the matrix used to carry out the operation.
- Mix Columns Operation each column is mixed independent of the other.
- Matrix multiplication is used.
- The output of this step is the matrix multiplication of the old values and a constant matrix.

$$
\begin{bmatrix} b0 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \end{bmatrix} \begin{bmatrix} c0 \end{bmatrix}
$$
$$
\begin{bmatrix} b1 \end{bmatrix} = \begin{bmatrix} 9 & 14 & 11 & 13 \end{bmatrix} \begin{bmatrix} c1 \end{bmatrix}
$$
$$
\begin{bmatrix} b2 \end{bmatrix} = \begin{bmatrix} 13 & 9 & 14 & 11 \end{bmatrix} \begin{bmatrix} c2 \end{bmatrix}
$$
$$
\begin{bmatrix} b3 \end{bmatrix} = \begin{bmatrix} 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c3 \end{bmatrix}
$$

Advanced Encryption Standard (AES)

- **Inverse Sub Bytes**
- Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.
- Function Substitute performs a byte substitution on each byte of the input word.
- For this purpose, it uses an S-box.

Advanced Encryption Standard (AES)

- **Applications of AES**
- Wireless security
- Database Encryption
- Secure communications
- Data storage
- Virtual Private Networks (VPNs)
- Secure Storage of Passwords
- File and Disk Encryption

THANK YOU