SYMMETIRC KEY ENCRYPTION

- Symmetric encryption, also **referred to as conventional encryption or single-key encryption**
- Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key.It is also known as conventional encryption.
- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm.
- Using the same key and a decryption algorithm,the plaintext is recovered from the ciphertext

Key Terms

● Plain Text : An original message

● Ciphertext: coded message

● Enciphering or encryption : The process of converting from plaintext to ciphertext

● Deciphering or Decryption: restoring the plaintext from the ciphertext.

The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a cryptographic system or a cipher.

Types of Cryptography

1. Symmetric Cryptography (Private Key Cryptography)

2. Asymmetric Cryptography (Public Key Cryptography)

BLOCK CIPHER V/S STREAM CIPHER

Block cipher and stream cipher are members of the family of symmetric key ciphers, essentially techniques used for directly transforming the plaintext into ciphertext.

CONFUSION AND DIFFUSION IN CRYPTOGRAPHY

- **Confusion** and **Diffusion** are both properties for creating a secure cipher.
- Confusion and diffusion are both used to prevent the encryption key from its deduction or to prevent the original message from being transmitted.
- **Confusion is utilized to create clueless ciphertext**
- **Diffusion is utilized to increase the redundancy of the plaintext** over most of the ciphertext to make it obscure.
- The stream cipher relies solely on confusion. On the other hand, diffusion is also utilized both stream and block ciphers.

```
Confusion = Substitution
a --> b
Example : Caesar Cipher

Diffusion = Transposition or Permutation
abcd --> dacb
Example : DES
```

DIFFERENCE BETWEEN CONFUSION AND DIFFUSION

CONFUSION	DIFFUSION
Confusion is a cryptographic technique that is used to create faint cipher texts.	Diffusion is used to create cryptic plain texts.
Confusion is possible through substitution algorithms.	Diffusion is possible through transposition algorithms.
In confusion, if one bit within the secret is modified, most or all bits within the cipher text also will be modified.	In diffusion, if one image within the plain text is modified, many or all image within the cipher text also will be modified
Both stream cipher and block cipher use confusion.	Only block cipher use diffusion.
The relation between the cipher text and the key is masked by confusion	The relation between the cipher text and the plain text is masked by diffusion.
`Example : Caesar Cipher`	`Example : DES`

BLOCK CIPHER & STREAM CIPHER

Block cipher is a **symmetric cryptographic technique** which used to **encrypt a fixed size data block** using a shared , secret key. During encryption, we used plaintext and ciphertext is the resultant encrypted text. It uses the same key to encrypt both the plaintext, and the ciphertext

Key Features of Block Ciphers:

- Fixed Block Size : The Data is encrypted in blocks of a predetermined size.
- Complex Operations : The Block ciphers use a series of the substitution and permutation operations to the achieve encryption.
- Modes of Operation : The Block ciphers can operate in the various modes such as the ECB (Electronic Codebook) and CBC (Cipher Block Chaining) to the enhance security.
- Examples : AES (Advanced Encryption Standard), DES (Data Encryption Standard) and Blowfish.

A **stream cipher encrypts data one bit or byte at a time** rather than in fixed-size blocks. It generates a keystream that is combined with the plaintext to the produce ciphertext.

Key Features of Stream Ciphers:

- Continuous Encryption : The Data is encrypted in a continuous stream, one bit or byte at a time.
- Keystream Generation : The Stream ciphers use a pseudorandom keystream
- generator to the create encryption keys.
- Efficiency : The Stream ciphers are typically more efficient for the encrypting data of variable length and in the streaming applications.
- Examples : RC4, Salsa20, and ChaCha20

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. E.g, vigenere cipher.

A block cipher is one in which a block of plaintext is treated as a whole and used to produce a cipher text block of equal length. Typically a block size of 64 or 128 bits is used.

- Popular variations of the block cipher algorithm include the ***Data Encryption Standard (DES), TripleDES,*** and the ***Advanced Encryption Standard (AES).***
- The stream cipher uses a shared key and operates on its input one bit at a time, which is the block cipher's counterpart.
- Alternative to the block cipher algorithm includes **public-key cryptography** and **asymmetric cryptography.** This algorithm uses the public key to encrypt plaintext and a private key to decrypt the ciphertext.

BLOCK CIPHER V/S STREAM CIPHER

Block Cipher	Stream Cipher
Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	**Stream cipher** is the kind of encryption that converts plaintext by taking one byte of the plaintext at a time.
It uses both diffusion and confusion principles for the conversion (used later in encryption).	Only the confusion principle is used by Stream Cipher for the conversion.
In **Block cipher,** ***reverse encryption or decryption*** is more difficult than stream cipher since more bits are combined to be encrypted in this scenario.	In a **stream cipher, XOR** is used for encryption that can quickly converted back to plain text.
Feistel Cipher is the most popular block cipher implementation.	**Vernam Cipher** is the main implementation of Stream Cipher.
Since a block cipher converts blocks at once, it converts more significant bits than a stream cipher, which can convert 64 bits or more.	In stream cipher, only 8 bits can be transformed simultaneously.
Simple design	Complex comparatively
64 Bits or more bits are used	**8 Bits** are used
Block ciphers can operate as a stream cipher.	Stream Cipher cannot operate as a block cipher.

BLOCK CIPHER PRINCIPLES

Virtually, all symmetric block encryption algorithms in current use are based on a structure referred to as **Fiestel block cipher**.

FEISTEL CIPHERS

- Feistel Cipher model is a structure or a design used to develop many block ciphers such as DES.
- Feistel cipher may have invertible, non-invertible and self invertible components in its design.
- Same encryption as well as decryption algorithm is used.
- A separate key is used for each round. However same round keys are used for encryption as well as decryption.

Feistel cipher algorithm

- Create a list of all the Plain Text characters.
- Convert the Plain Text to ASCII and then 8-bit binary format.
- Divide the binary Plain Text string into two halves: left half (L1)and right half (R1)
- Generate a random binary keys (K1 and K2) of length equal to the half the length of the Plain Text for the two rounds.

First Round of Encryption

a. Generate function f1 using R1 and K1 as follows:

F1 = xor (R1 , K1)

b. Now the new left half(L2) and right half(R2) after round 1 are as follows:

R2 = xor (F1 , L1)

L2 = R1

Second Round of Encryption

a. Generate function f2 using R2 and K2 as follows:

F2 = xor (R2, K2)

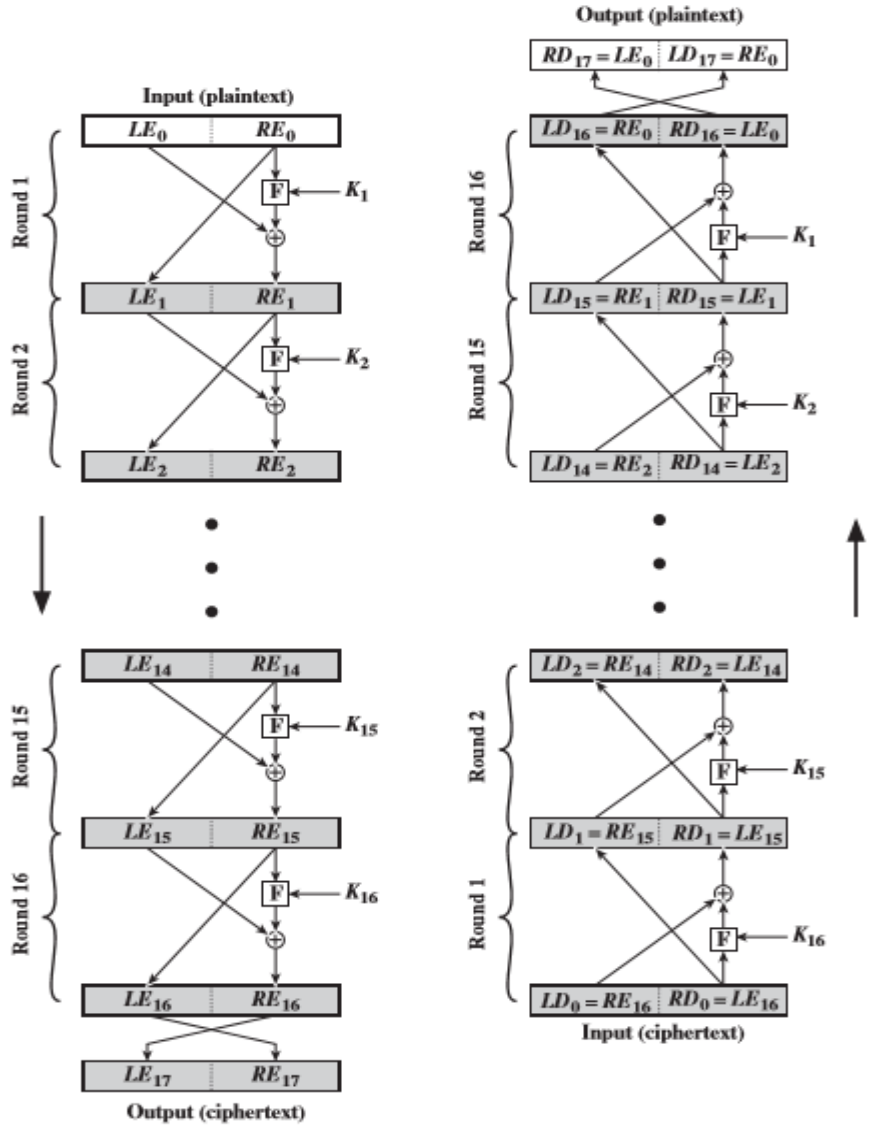
b. Now the new left half(L3) and right half(R3) after round 2 are as follows:

R3 = xor (F2,L2)

L3 =R2

Concatenation of R3 to L3 is the Cipher Text

Same algorithm is used for decryption to retrieve the Plain Text from the Cipher Text.



The exact realization of a Feistel network depends on the choice of the following parameters and design features:

Block size :

- Larger block sizes mean greater security (all other things being equal) but reduced encryption/decryption speed for a given algorithm.
- The greater security is achieved by greater diffusion.
- Traditionally, a block size of 64 bits has been considered a reasonable tradeoff and was nearly universal in block cipher design.
- However,the new AES uses a 128-bit block size.

Key size:

- Larger key size means greater security but may decrease encryption/ decryption speed.
- The greater security is achieved by greater resistance to brute-force attacks and greater confusion.Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.

Number of rounds:

- The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security.
- A typical size is 16 rounds.

Subkey generation algorithm:

- Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

Round function F:

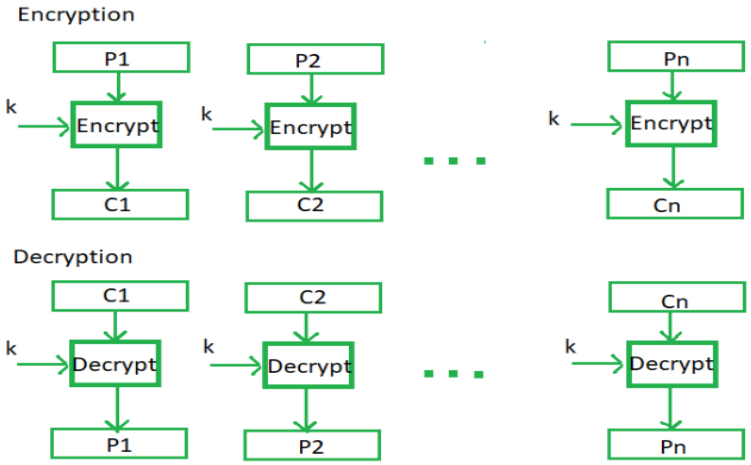
- Again, greater complexity generally means greater resistance to cryptanalysis.

MODES OF OPERATION OF BLOCK CIPHER

1. **Electronic Code Book (ECB) mode**
2. **Cipher Block Chaining (CBC) mode**
3. **Cipher Feedback (CFB) mode**
4. **Output Feedback (OCB) mode**
5. **Counter (CTR) mode.**

ELECTRONIC CODEBOOK MODE (ECB)

- Electronic code book is the **easiest block cipher mode of functioning.**
- It is easier because of **direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext.** Generally, if a message is larger than ***b*** bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.
- It is the most straightforward block cipher operating mode.
- It does not introduce any randomness to the key stream, and **it is the only mode we can use to encrypt a single-bit stream.**
- Using the cipher's key and substitution alphabet, each plaintext symbol, such as a character from the plaintext alphabet, is transformed into a ciphertext symbol.
- **Each block of plaintext is encrypted separately from every other block**.
- Only 8 bytes of the key are used when the plaintext block is only **8 bytes** long, and all **100 bytes** of the key are utilised when the plaintext block is **100 bytes** long.



Advantages of using ECB –

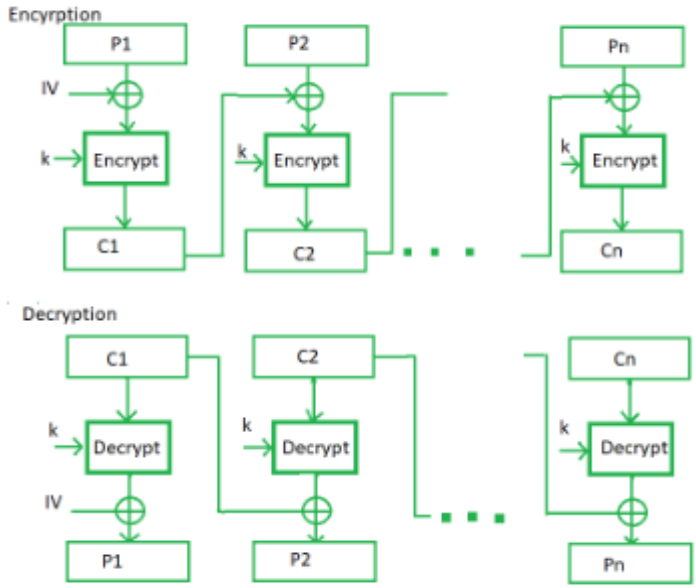
- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

CIPHER BLOCK CHAINING MODE (CBC)

- Cipher block chaining or CBC is an **advancement made on ECB** since ECB compromises some security requirements.
- In CBC, **the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block**. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.
- When using CBC mode to encrypt data, each block of plaintext is combined with the ciphertext that came before it.
- A ciphertext generated by the symmetric algorithm depends on all plaintext block processed in the data stream before it. This is done to ensure that every block of the ciphertext depends on every other block that came before it. Before using the cipher algorithm to encrypt the data, each block of plaintext is XORed (exclusive OR) with the block of ciphertext that came before it.
- Numerous security applications used CBC mode**.**
- For example, **Secure Sockets Layer/Transport Layer Security uses CBC mode** in order to encrypt data which is transferred over the internet.



Advantages of CBC –

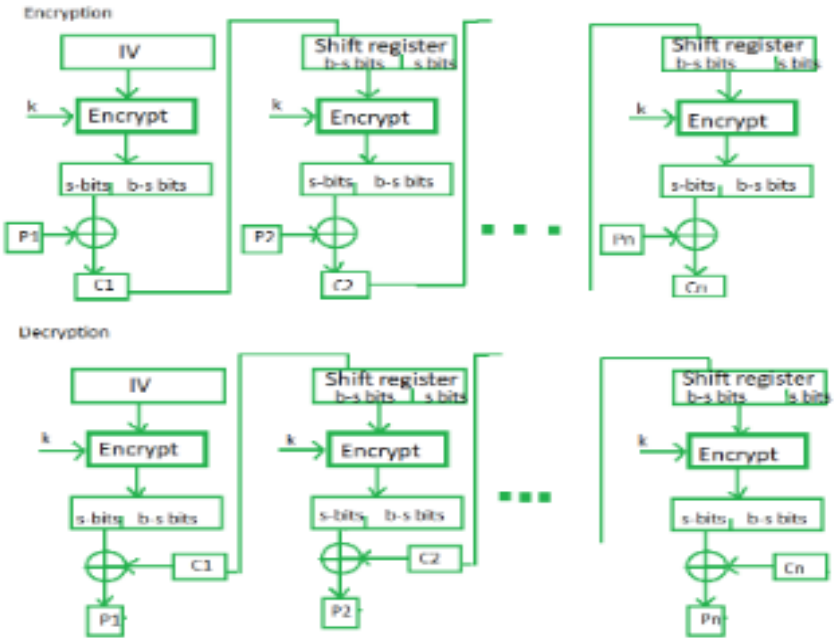
- CBC works well for input greater than **b** bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.

Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.

CIPHERTEXT FEEDBACK MODE (CFB)

- In this mode **the cipher is given as feedback to the next block of encryption with some new specifications**
- First, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits.
- The left-hand side *s* bits are selected along with plaintext bits to which an XOR operation is applied.
- The result is given as input to a shift register having b-s bits to LHS, s bits to RHS and the process continues.
- The encryption and decryption process for the same, both of them use encryption algorithms.



Advantages of CFB –

- There is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

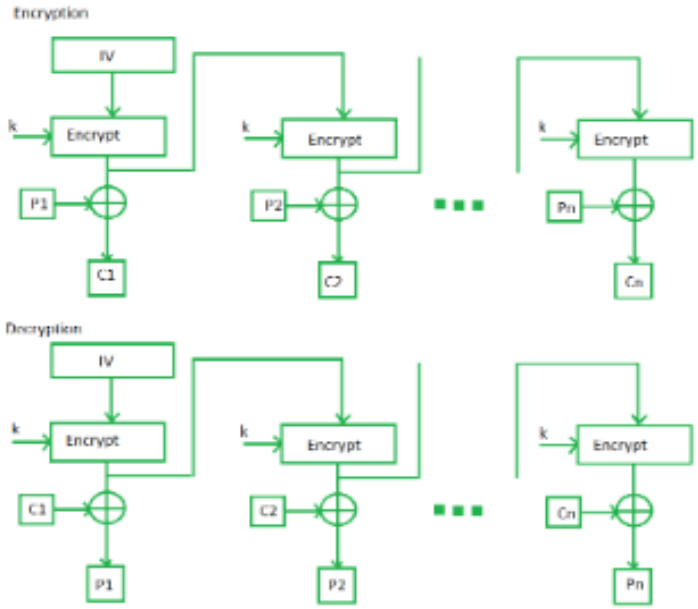
Disadvantages of using CFB –

- The drawbacks of CFB are the same as those of CBC mode.
- Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

OUTPUT FEEDBACK MODE (OFB)

- The output feedback mode follows nearly the same process as the Cipher Feedback mode except that **it sends the encrypted output as feedback instead of the actual cipher which is XOR output.**
- All bits of the block are sent instead of sending selected *s* bits.
- The Output Feedback mode of block cipher holds **great resistance towards bit transmission errors**.
- It also **decreases the dependency or relationship of the cipher on the plaintext.**

In certain ways, **CBC and OFB modes are comparable** and can be used with any block cipher. It uses a feedback mechanism; however, in OFB mode, the preceding block of ciphertext is XORed with the plaintext after encryption rather than prior to encryption.



Advantages of OFB –

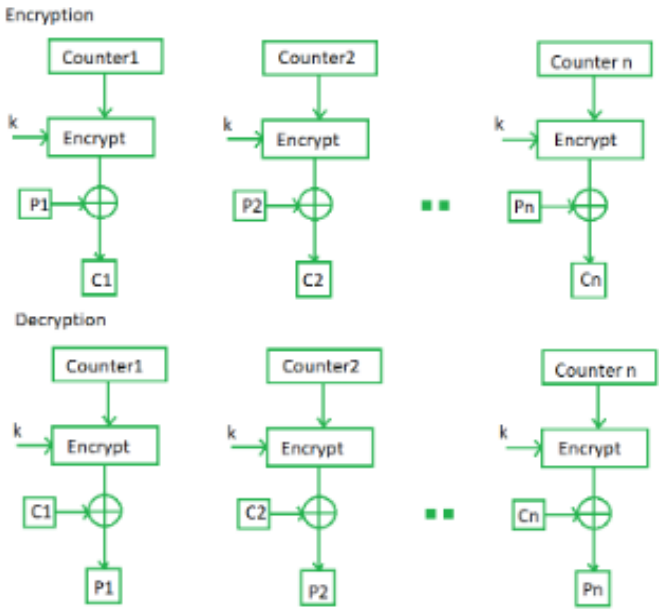
- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as **it is free from bit errors in the plaintext block**.

Disadvantages of OFB-

- Because to its operational modes, it is more susceptible to a message stream modification attack than CFB.

COUNTER MODE (CTR)

- The Counter Mode or CTR is a simple counter-based block cipher implementation.
- CTR mode **uses a block chaining mode of encryption** as a building block
- **Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block**.
- The CTR mode is **independent of feedback use** and thus can be implemented in parallel.



Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.
- Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

Disadvantages of Counter-

- CTR mode requires a synchronous counter at both the transmitter and the receiver is a severe drawback.
- The recovery of plaintext is erroneous when synchronisation is lost.

DATA ENCRYPTION STANDARD (DES)

Published by the National Institute of Standards and Technology (NIST).

- The Data Encryption Standard (DES) is **a symmetric-key block cipher**
- 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).

DES uses a 56-bit key.

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	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

Figure - discording of every 8th bit of original key

Actually, The initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded. Thus, the discarding of every 8th bit of the key produces a **56-bit key** from the original **64-bit key**.

DES is based on the two fundamental attributes of cryptography:

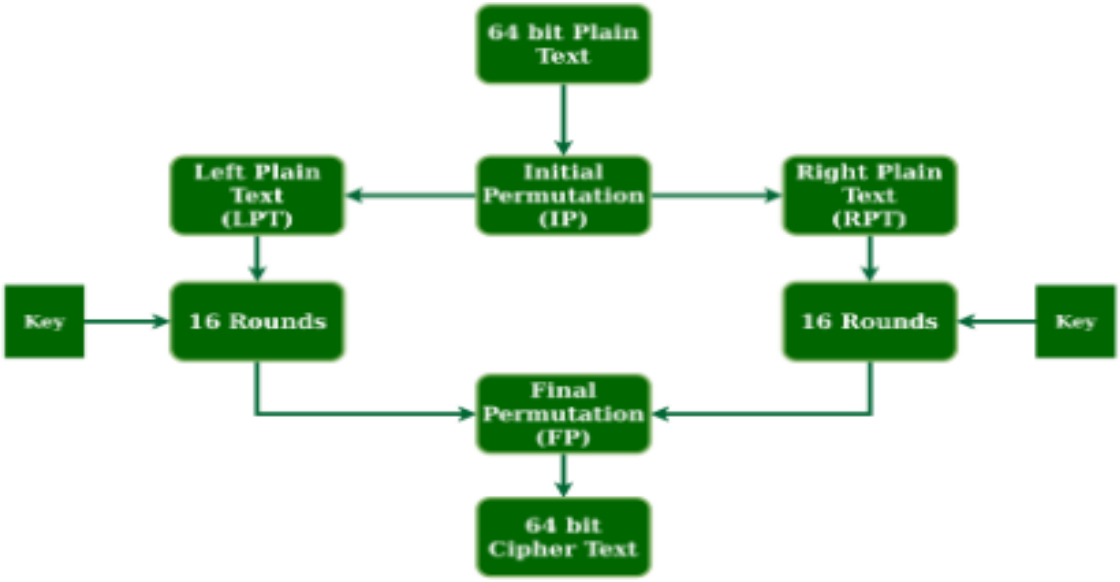
- **substitution** (also called confusion)
- **transposition** (also called diffusion).

DES consists of 16 steps

Each of which is called a round. Each round performs the steps of substitution and transposition.

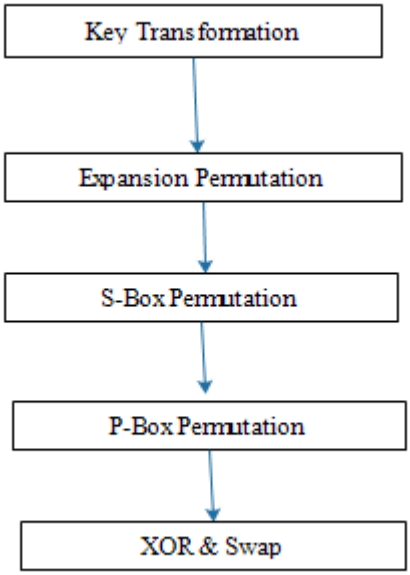
In the first step, the 64-bit plain text block is handed over to an Initial Permutation (IP) function.

- The initial permutation is performed on plain text.
- Next, the initial permutation (IP) produces two halves of the permuted block:
    - Left Plain Text (LPT)
    - Right Plain Text (RPT).
- Now each LPT and RPT go through 16 rounds of the encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block.
- The result of this process produces 64-bit ciphertext.



INITIAL PERMUTATION (IP)

- **The plain text is divided into smaller chunks of 64-bit size. The IP is performed before the first round. This phase describes the implementation of the transposition process.** For example, the 58th bit replaces the first bit, the 50th bit replaces the second bit, and so on**. The resultant 64-bit text is split into two equal halves of 32-bit each called Left Plain Text (LPT) and Right Plain Text (RPT).**



Step 1: Key Transformation

- The DES process uses a 56-bit key, which is obtained by eliminating all the bits present in every 8th position in a 64-bit key. In this step, a 48-bit key is generated. The 56-bit key is split into two equal halves and depending upon the number of rounds the bits are shifted to the left in a circular fashion.

- Due to this, all the bits in the key are rearranged again. We can observe that some of the bits get eliminated during the shifting process, producing a 48-bit key. This process is known as compression permutation.

- Because of this compression permutation technique, a different subset of key bits is used in each round. That makes DES not easy to crack.

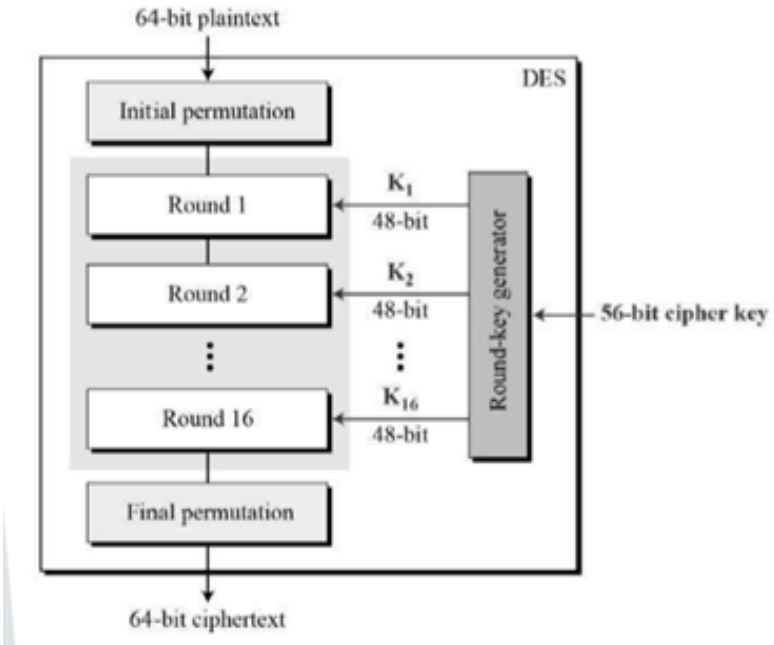
Step 2: Expansion Permutation

- In this step, it is expanded from 32-bit to 48-bit.

- The RPT of 32-bit size is broken down into 8 chunks of 4 bits each and extra two bits are added to every chunk, later on, the bits are permutated among themselves leading to 48-bit data.

- An XOR function is applied in between the 48-bit RPT and the resulting output is given to the next step, which is the **S-Box substitution**.

DES Algorithm Procedure



DES transforms 64-bit plain text into a 64-bit cipher text. The same key is also utilized to decode the text

- The 64-bit plain text block is first sent to an initial permutation (IP) function to start the process.
- The plain text is subsequently subjected to the initial permutation (IP).
- The Left Plain Text (LPT) and Right Plain Text (RPT) portions of the permuted block are then created by the initial permutation (IP).

- There are 16 rounds of encryption for each LPT and RPT.
- Finally, the LPT and RPT are reunited, and the newly combined block is subjected to a Final Permutation (FP).
- This procedure provides the necessary 64-bit ciphertext as a result.

The phase of the encryption processis further divided into the following five stages:

- Key transition
- Expansion permutation
- XOR and swap
- S-Box
- P-Box permutations

We employ the same procedure for decryption and arrange the 16 round keys in the other direction.

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.

Weaknesses in DES -Key selected are weak keys. These keys shall be avoided.

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.

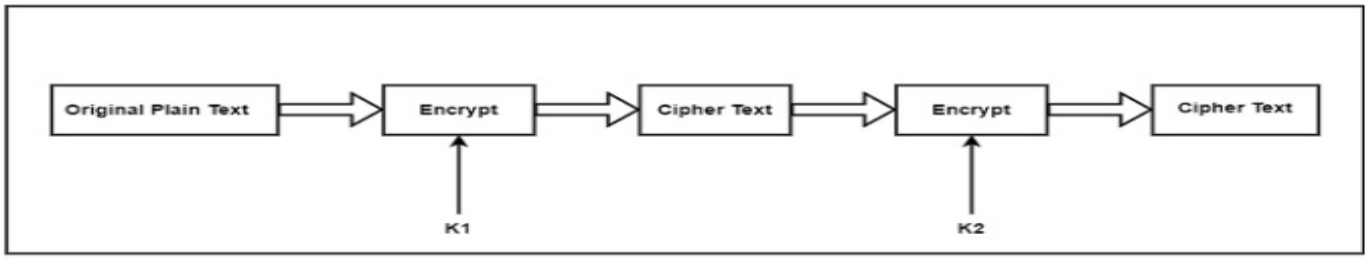
DOUBLE DES (2DES)

Double DES is an encryption approach which uses two instance of DES on same plain text. In both instances it uses different keys (K1 and K2) to encrypt the plain text. Both keys are required at the time of decryption.

It can implement DES on the original plain text using K1 to get the encrypted text. Then it can implement DES on that encrypted text, but this time with the different key K2.

At the time of decryption,the double encrypted cipher-text block is first decrypted using the key K2 to make the singly encrypted cipher text. This ciphertext block is then decrypted using the key K1 to acquire the original plaintext block.

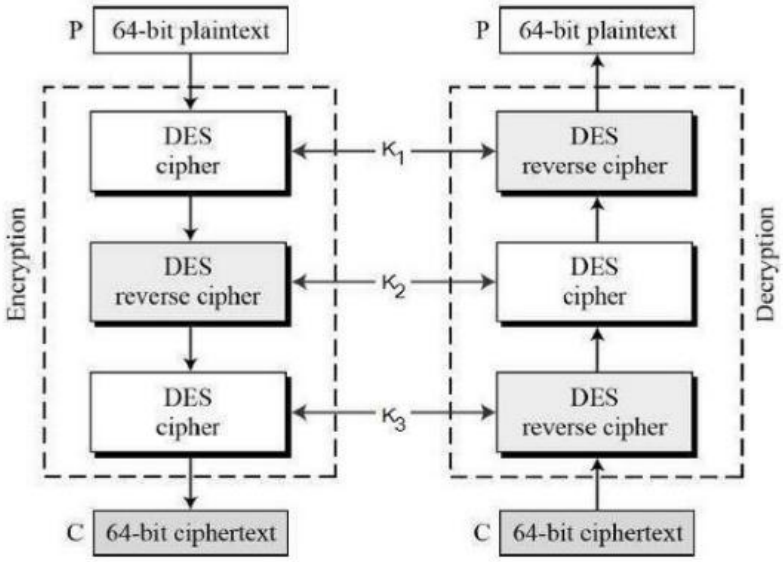
However **double DES uses 112 bit key but gives security level of 2^56 not 2^112 and this is because of meet-in-the middle attack which can be used to break through double DES.**



Double DES Encryption

TRIPLE DES (3DES)

- Triple Data Encryption Standard (Triple DES) is a symmetric block cipher-based cryptography standard that uses multiple rounds of the Data Encryption Standard (DES) to improve security.
- It is also known as Triple DES because it uses the Data Encryption Standard (DES) cypher which takes three times to encrypt its data.
- It is essentially a block cypher used to encrypt data in 64-bit blocks.
- As a symmetric cryptographic scheme, DES implementations rely on the same secret keys shared between the sender and the recipient.
- Security-wise, it outperforms the original Data Encryption Standard (DES).
- However, Triple DES is less efficient and slower than the Advanced Encryption Standard (AES).



FEATURES OF TRIPLE DES

- It utilizes a triple layer of encryption which means it utilizes three different keys to encrypt the plaintext three times.
- It supports variable key sizes which range from 128 bits to 192 bits.
- It basically involves the usage of a symmetric key encryption system, which states that the same key is used for both encryption and decryption.
- It is a block cypher encryption algorithm that works with 64-bit blocks of plaintext at a time.
- It is suitable for legacy systems that require secure encryption.

ENCRYPTION PROCESS

The Encryption process of Triple DES involves the following steps:-

Key Generation

This is the first step of the Encryption process of Triple DES. In this step, three unique keys are generated using a key derivation algorithm.

Initial Permutation

This step comes after the process of Key Generation. It involves the rearrangement of the bits of the plaintext according to a predefined permutation table.

Three Rounds of Encryption
This is regarded as the most important round of the encryption process of Triple DES. It consists of multiple rounds typically 48 rounds in total. In this step, the plaintext is processed three times and get encrypted, each time we take use of a different key, to create three layers of encryption.

Final Permutation
It completes the Triple DES encryption process. In this step, the resulting ciphertext block undergoes a final permutation (FP) operation, which is the inverse of the initial permutation. It returns the bits of the ciphertext block to their original order.

ALGORITHM -3DES

The encryption-decryption process is as follows –

● Encrypt the plaintext blocks using single DES with key K1.

● Now decrypt the output of step 1 using single DES with key K2.

● Finally, encrypt the output of step 2 using single DES with key K3.

● The output of step 3 is the ciphertext.

● Decryption of a ciphertext is a reverse process. User first decrypt using K3, then encrypt with K2, and finally decrypt with K1.

Due to this design of Triple DES as an **encrypt–decrypt–encrypt process**

Advantages of Triple DES

- It provides three layered encryption technique which provides enhanced security features.
- It offers backward compatibility with Data Encryption Standard which means it can use legacy system that DES uses.
- It supports variable key sizes, which led to enhanced security.
- It is widely used encryption algorithm and is used with many encryption standards and protocols.

Applications of Triple DES

- **Financial Transactions:** Triple DES is widely used in financial transactions because it secures the transaction that takes place like online banking, credit card payment, etc.

- **Data Protection:** Triple DES is often used to protect sensitive data which are stored on computers, servers, and other electronic devices. It is used in various fields such as healthcare, government sector, etc.

- **Virtual Private Networks:** Triple DES is used to secure the communication process between remote locations. It is done by securing the virtual private networks.

- **Authentication and Digital Signatures:** Triple DES can be used in combination with cryptographic hash functions for generating digital signatures and verifying the authenticity of digital documents and messages

ADVANCED ENCRYPTION STANDARD (AES ALGORITHM)

- 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.
- Developed by the National Institute of Standards and Technology (NIST)
- AES is a **Block Cipher Symmetric Encryption Algorithm.**
- AES relies on the **substitution-permutation network principle**
- **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

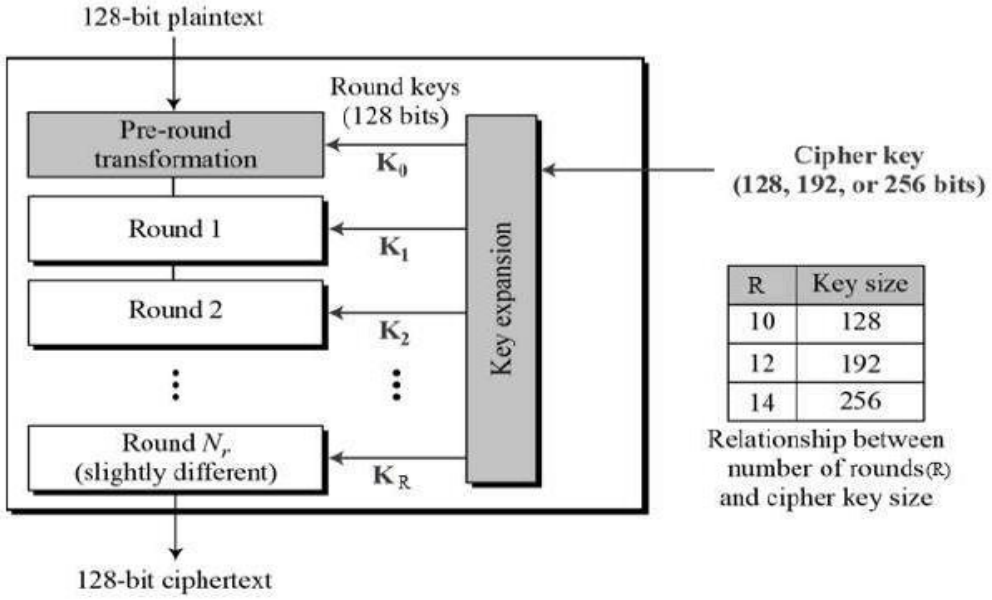
WORKING OF THE CIPHER

- 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 :
    - **128-bit key – 10 rounds**
    - **192-bit key – 12 rounds**
    - **256-bit key – 14 rounds**

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.

The schematic of AES structure is given in the following illustration –

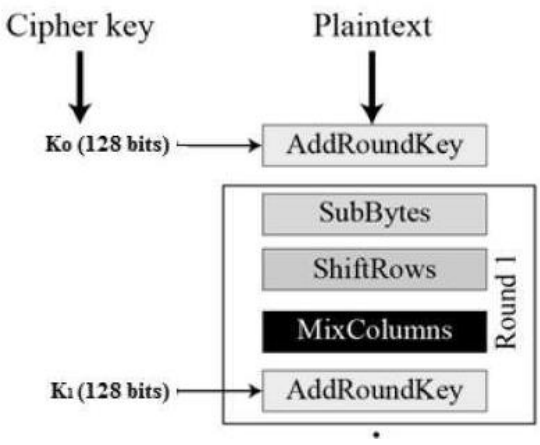


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

ENCRYPTION PROCESS

Each round comprise of four sub-processes. The first round process is depicted below



Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns perform the permutation in the algorithm.

Byte Substitution (SubBytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

Shift rows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

- 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.

Mix Columns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

Add round key

The 16 bytes of the matrix are now considered as 128 bits and are XOR ed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

DECRYPTION PROCESS

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

APPLICATIONS

AES is widely used in many applications which require secure data storage and transmission. Some common use cases include:

- **Wireless security:** AES is used in securing wireless networks, such as Wi-Fi networks, to ensure data confidentiality and prevent unauthorized access.
- **Database Encryption:** AES can be applied to encrypt sensitive data stored in databases. This helps protect personal information, financial records, and other confidential data from unauthorized access in case of a data breach.
- **Secure communications:** AES is widely used in protocols such as internet communications, email, instant messaging, and voice/video calls. It ensures that the data remains confidential.
- **Data storage:** AES is used to encrypt sensitive data stored on hard drives, USB drives, and other storage media, protecting it from unauthorized access in case of loss or theft.
- **Virtual Private Networks (VPNs):** AES is commonly used in VPN protocols to secure the communication between a user's device and a remote server. It ensures that data sent and received through the VPN remains private and cannot be deciphered by eavesdroppers.
- **Secure Storage of Passwords:** AES encryption is commonly employed to store passwords securely. Instead of storing plaintext passwords, the encrypted version is stored. This adds an extra layer of security and protects user credentials in case of unauthorized access to the storage.
- **File and Disk Encryption:** AES is used to encrypt files and folders on computers, external storage devices, and cloud storage. It protects sensitive data stored on devices or during data transfer to prevent unauthorized access.

STREAM CIPHERS

- In stream cipher, one byte is encrypted at a time while in block cipher ~128 bits are encrypted at a time
- Initially, a key(k) will be supplied as input to pseudorandom bit generator and then it produces a random 8-bit output which is treated as keystream.
- The resulted keystream will be of size 1 byte, i.e., 8 bits.
- Stream ciphers are fast because they encrypt data bit by bit or byte by byte, which makes them efficient for encrypting large amounts of data quickly.

- Stream ciphers work well for real-time communication, such as video streaming or online gaming, because they can encrypt and decrypt data as it's being transmitted.

Encryption

- Plain Text and Keystream produces Cipher Text (Same keystream will be used for decryption.).
- The Plaintext will undergo XOR operation with keystream bit-by-bit and produces the Cipher Text.

Decryption

- Cipher Text and Keystream gives the original Plain Text (Same keystream will be used for encryption.).
- The Ciphertext will undergo XOR operation with keystream bit-by-bit and produces the actual Plain Text.

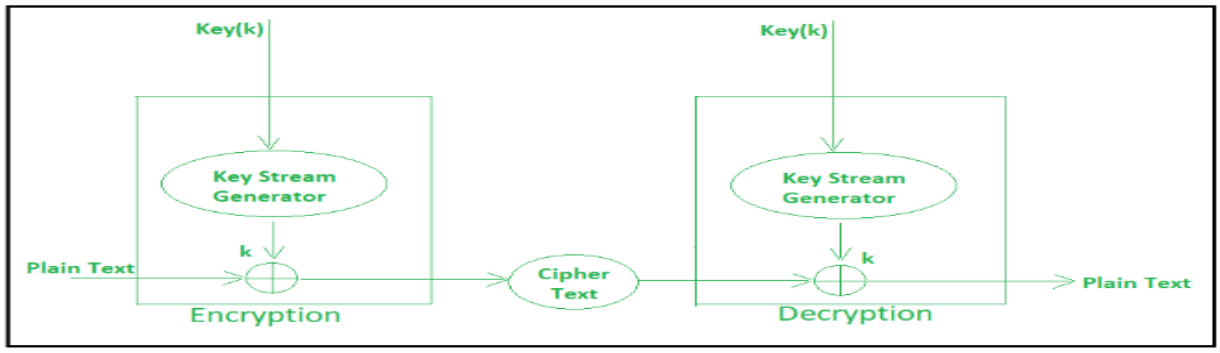


Diagram of Stream Cipher

Advantages of Stream Ciphers

- **Speed :** Generally, this type of encryption is quicker than others, such as block ciphers.
- **Low complexity :** Stream ciphers are simple to implement into contemporary software, and developers don't require sophisticated hardware to do so.
- **Sequential in nature :** Certain companies handle communications written in a continuous manner. Stream ciphers enable them to transmit data when it's ready instead of waiting for everything to be finished because of their bit-by-bit processing.
- **Accessibility :** Using symmetrical encryption methods like stream ciphers saves businesses from having to deal with public and private keys. Additionally, computers are able to select the appropriate decryption key to utilize thanks to mathematical concepts behind current stream ciphers.

Disadvantages of Stream Ciphers

- If an error occurs during transmission, it can affect subsequent bits, potentially corrupting the entire message because stream ciphers rely on previously stored cipher bits for decryption
- Maintaining and properly distributing keys to stream ciphers can be difficult, especially in large systems or networks.
- Some stream ciphers may be predictable or vulnerable to attack if their key stream is not properly designed, potentially compromising the security of the encrypted data.

Eg . of Stream Cipher is RC4

RC4 ENCRYPTION

- RC4 stands for Rivest Cipher 4, and it is a stream cipher.
- Because RC4 is a stream cipher, it encrypts data bytes by bits.
- Because of its speed and simplicity
- RC4 is the most extensively used stream cipher of all the stream ciphers.
- **RC4** is a stream cipher and variable-length key algorithm.

- This algorithm encrypts one byte at a time (or larger units at a time).
- A key input is a pseudorandom bit generator that produces a stream 8-bit number that is unpredictable without knowledge of input key.
- The output of the generator is called key-stream, is combined one byte at a time with the plaintext stream cipher using X-OR operation.

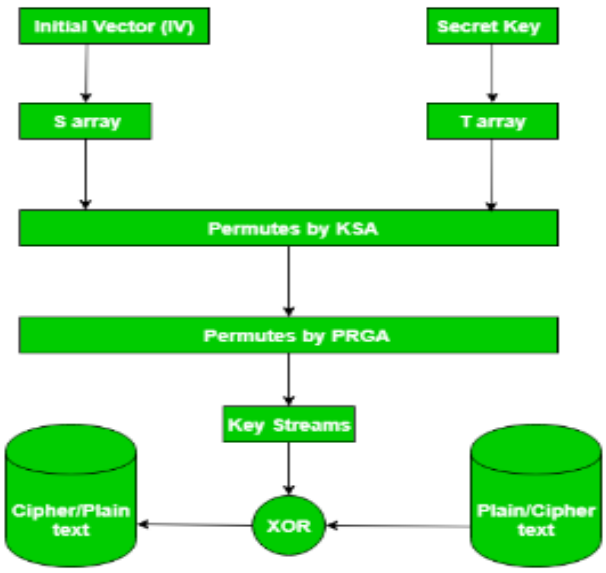
Working of RC4

For encryption −

- The user enters the Plaintext and a secret key.
- For the secret key entered, the encryption engine creates the key stream using the KSA and PRGA algorithms.
- Plaintext is XOR ed with the generated key stream. Because RC4 is a stream cipher, byte-by-byte XORing is used to generate the encrypted text.
- This encrypted text is now sent in encrypted form to the intended recipient.

For Decryption −

- The same byte-wise XOR technique is used on the ciphertext to decrypt it.



Features of the RC4 encryption algorithm:

1. **Symmetric key algorithm:** RC4 is a symmetric key encryption algorithm, which means that the same key is used for encryption and decryption.

2. **Stream cipher algorithm:** RC4 is a stream cipher algorithm, which means that it encrypts and decrypts data one byte at a time. It generates a key stream of pseudorandom bits that are XORed with the plaintext to produce the ciphertext.

3. **Variable key size:** RC4 supports variable key sizes, from 40 bits to 2048 bits, making it flexible for different security requirements.

4. **Fast and efficient:** RC4 is a fast and efficient encryption algorithm that is suitable for low-power devices and applications that require high-speed data transmission.

5. **Widely used:** RC4 has been widely used in various applications, including wireless networks, secure sockets layer (SSL), virtual private networks (VPN), and file encryption.

6. **Vulnerabilities:** RC4 has several vulnerabilities, including a bias in the first few bytes of the keystream, which can be exploited to recover the key. As a result, RC4 is no longer recommended for use in new applications.

Advantages:

1. **Fast and efficient:** RC4 is a very fast and efficient encryption algorithm, which makes it suitable for use in applications where speed and efficiency are critical.

2. **Simple to implement:** RC4 is a relatively simple algorithm to implement, which means that it can be easily implemented in software or hardware.

3. **Variable key size:** RC4 supports variable key sizes, which makes it flexible and adaptable for different security requirements.

4. **Widely used:** RC4 has been widely used in various applications, including wireless networks, secure sockets layer (SSL), virtual private networks (VPN), and file encryption.

Disadvantages:

1. **Vulnerabilities:** RC4 has several known vulnerabilities that make it unsuitable for new applications. For example, there is a bias in the first few bytes of the keystream, which can be exploited to recover the key.

2. **Security weaknesses:** RC4 has some inherent weaknesses in its design, which make it less secure than other encryption algorithms, such as AES.

3. **Limited key length:** The maximum key length for RC4 is 2048 bits, which may not be sufficient for some applications that require stronger encryption.

4. **Not recommended for new applications:** Due to its vulnerabilities and weaknesses, RC4 is no longer recommended for use in new applications.