**Security** = **Security** theater is a critical term for measures that change perceptions of **security** without necessarily affecting **security** itself. For **example**, visual signs of **security** protections, such as a home that advertises its alarm system, may deter an intruder, whether or not the system functions properly.

**Need of Cryptography-** In IT security, **cryptography** is important in achieving data confidentiality, data integrity, user authentication and non-repudiation.

**Principles of security:-**

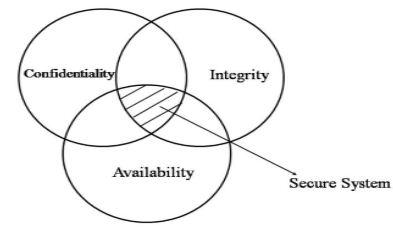
**1.Confidentiality:**

* Confidentiality is probably the most common aspect of information security. The principle of confidentiality specifies that only the sender and intended recipient should be able to access the contents of a message.
* Confidentiality gets compromised if an unauthorized person is able to access a message. Protection of confidential information is needed. An organization needs to guard against those malicious actions to endanger the confidentiality of its information.
* Example: Banking customers accounts need to be kept secret. Confidentiality not only applies to the storage of the information but also applies to the transmission of information. When we send a piece of the information to be stored in a remote computer or when we retrieve a piece of information from a remote computer we need to conceal it during transmission. Interception causes loss of message confidentiality.

**2. Integrity:**

* Information needs to be changed constantly. Integrity means that changes need to be done only by authorized entities and through authorized mechanisms. When the contents of a message are changed after the sender sends it, before it reaches the intended recipient it is said that integrity of the message is lost.
* Integrity violation is not necessarily the result of a malicious act; an interruption in the system such as a power surge may also create unwanted changes in some information.
* Modification causes loss of message integrity.

**3. Availability:**

* The principle of availability states that resources should be available to authorized parties at all times. The information created and stored by an organization needs to be available to authorized entities. Information is useless if it is not available.
* Information needs to be constantly changed which means it must be accessible to authorized entities. The unavailability of information is just as harmful for an organization as the lack of confidentiality or integrity.
* Example: The situation can be difficult for a bank if the customer could not access their accounts for transactions.
* Interruption puts the availability of resources in danger.
* The diagram above explains the balance concept. The right balance of the three goals is needed to build a secure system. If the goals are not balanced then a small hole is created for attackers to   
  nullify the other objectives of security. Having a highly confidential system but low availability then the system is not secure.
* Example: A system can protect confidentiality and integrity but if the resource us not available the other two goals also are of no use.

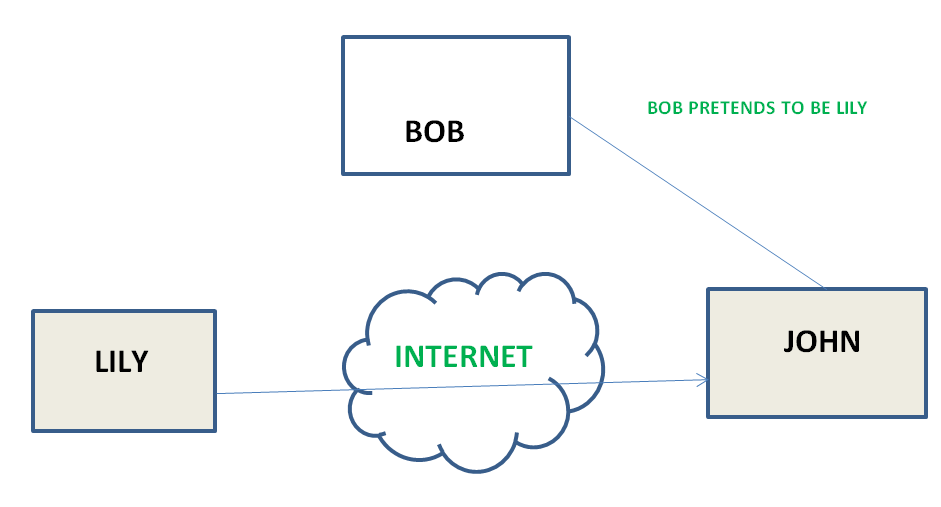
**Authentication: - User authentication** is a process that allows a device to verify the identity of someone who connects to a network resource. There are many technologies currently available to a network administrator to **authenticate** users. ... The Firebox also has its own **authentication** server.

**Non Repudiation**:- **Nonrepudiation** is the assurance that someone cannot deny something. ... You might send registered mail, for **example**, so the recipient cannot deny that a letter was delivered.

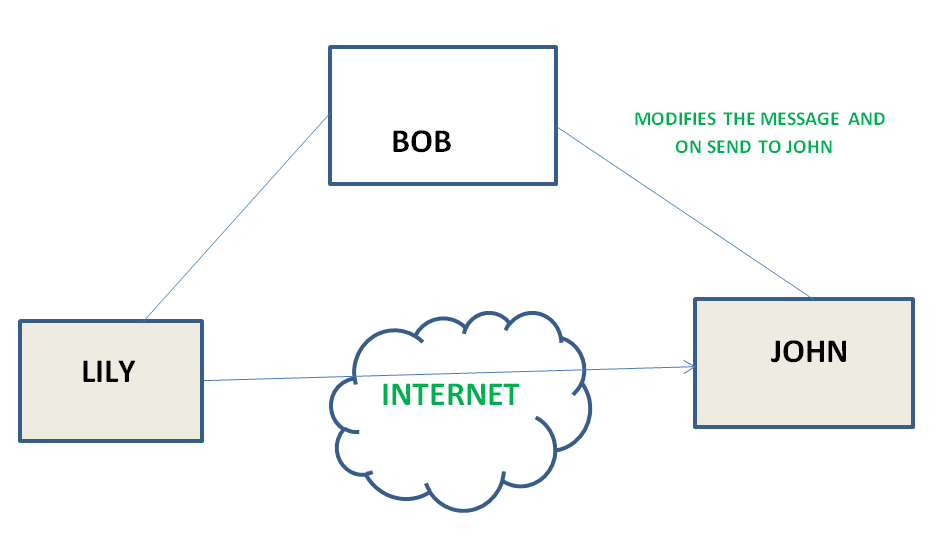
Active and Passive attacks in Information Security

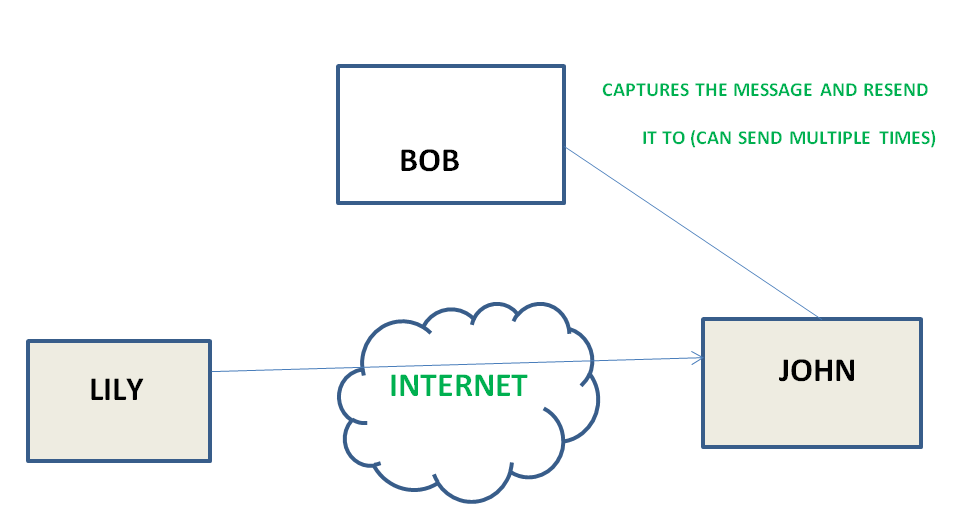
**Active attacks:** An Active attack attempts to alter system resources or effect their operations. Active attack involves some modification of the data stream or creation of false statement. Types of active attacks are as following:

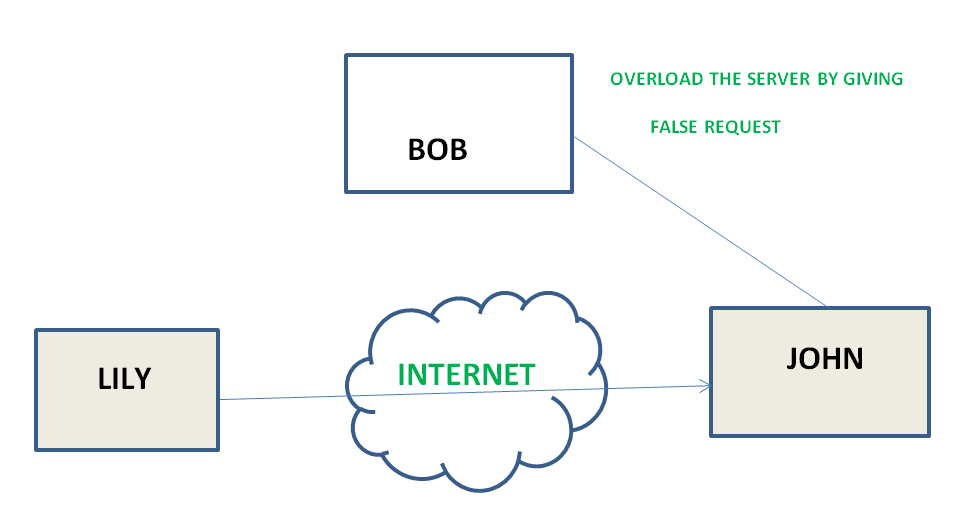
1. **Masquerade –**  
   Masquerade attack takes place when one entity pretends to be different entity. A Masquerade attack involves one of the other form of active attacks.



1. **Modification of messages –**  
   It means that some portion of a message is altered or that message is delayed or reordered to produce an unauthorized effect. For example, a message meaning “Allow JOHN to read confidential file X” is modified as “Allow Smith to read confidential file X”.

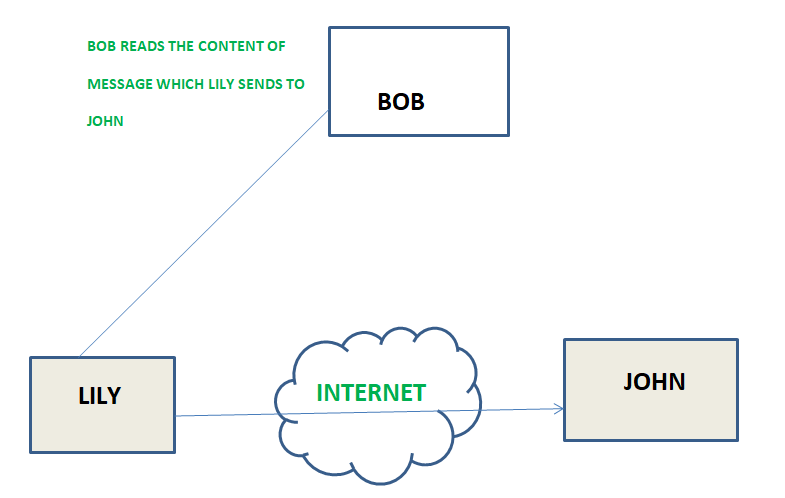


1. **Repudiation –**  
   This attack is done by either sender or receiver. The sender or receiver can deny later that he/she has sent or receive a message. For example, customer ask his Bank “To transfer an amount to someone” and later on the sender (customer) deny that he had made such a request. This is repudiation.
2. **Replay –**  
   It involves the passive capture of a message and its subsequent the transmission to produce an authorized effect.
3. **Denial of Service –**  
   It prevents normal use of communication facilities. This attack may have a specific target. For example, an entity may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network wither by disabling the network or by overloading it by messages so as to degrade performance.

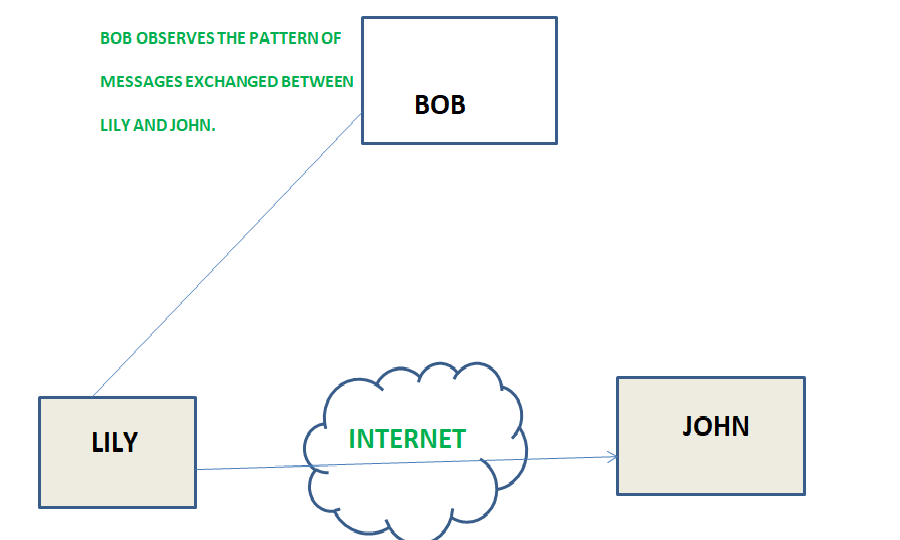


**Passive attacks:** A Passive attack attempts to learn or make use of information from the system but does not affect system resources. Passive Attacks are in the nature of eavesdropping on or monitoring of transmission. The goal of the opponent is to obtain information is being transmitted. Types of Passive attacks are as following:

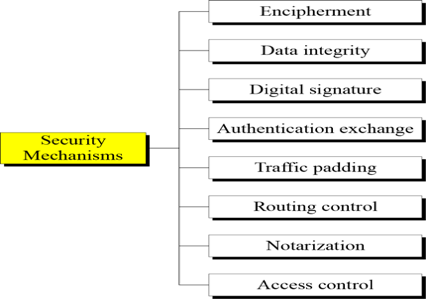
1. **The release of message content –**  
   Telephonic conversation, an electronic mail message or a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

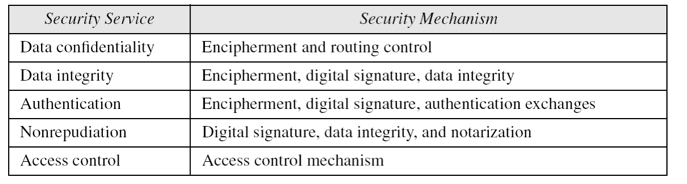
****

**Traffic analysis –**  
Suppose that we had a way of masking (encryption) of information, so that the attacker even if captured the message could not extract any information from the message.  
The opponent could determine the location and identity of communicating host and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

****

**Security Mechanism:-**

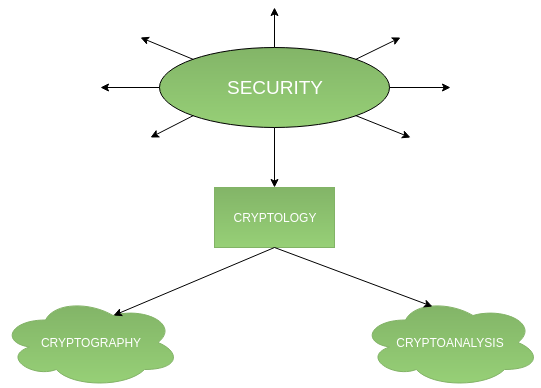
****

**Relation between security services and mechanisms**

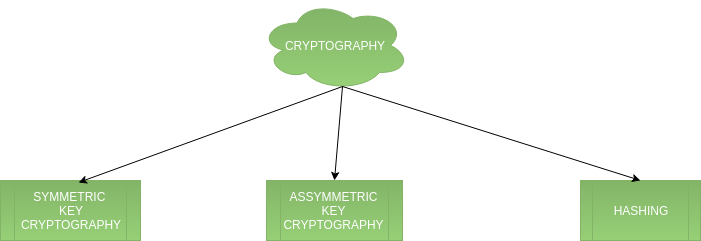
# Introduction to Crypto-terminologies

Cryptography is an important aspect when we deal with network security. ‘Crypto’ means secret or hidden. Cryptography is the science of secret writing with the intention of keeping the data secret. Cryptanalysis, on the other hand, is the science or sometimes the art of breaking cryptosystems. These both terms are a subset of what is called as Cryptology.

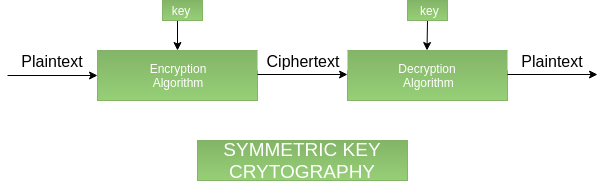
**Classification –**  
The flowchart depicts that cryptology is only one of the factors involved in securing networks. Cryptology refers to study of codes, which involves both writing (cryptography) and solving (cryptanalysis) them. Below is a classification of the crypto-terminologies and their various types.

****

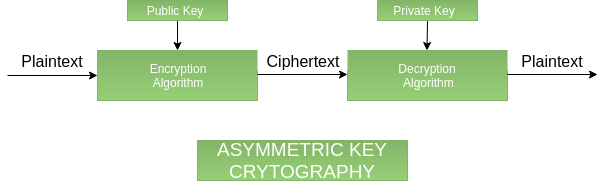
**Cryptography –**  
Cryptography is classified into symmetric cryptography, asymmetric cryptography and hashing. Below are the description of these types.

****

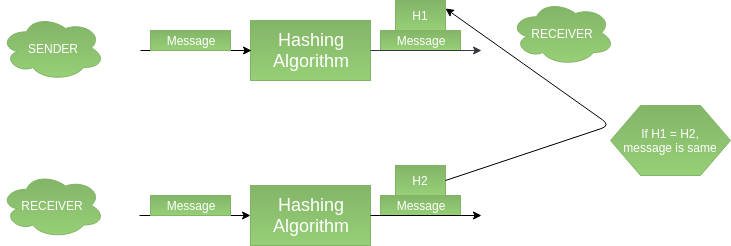
**1.Symmetric key cryptography –**  
It involves usage of one secret key along with encryption and decryption algorithms which help in securing the contents of the message. The strength of symmetric key cryptography depends upon the number of key bits. It is relatively faster than asymmetric key cryptography. There arises a key distribution problem as the key has to be transferred from the sender to receiver through a secure channel.

****

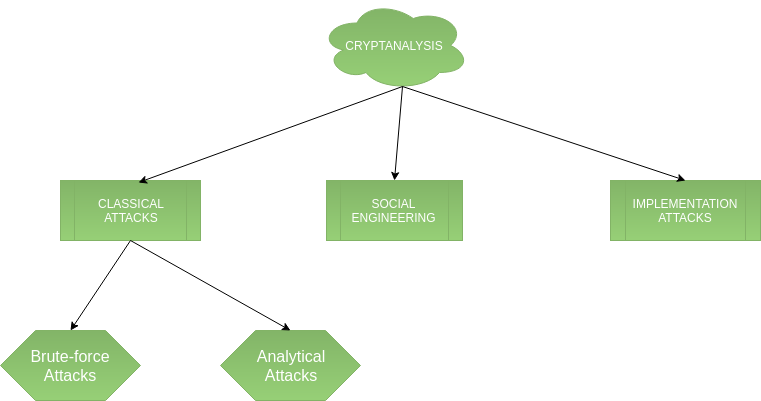
1. **Asymmetric key cryptography –**  
   It is also known as public key cryptography because it involves usage of a public key along with secret key. It solves the problem of key distribution as both parties uses different keys for encryption/decryption. It is not feasible to use for decrypting bulk messages as it is very slow compared to symmetric key cryptography.

****

**Hashing –**  
It involves taking the plain-text and converting it to a hash value of fixed size by a hash function. This process ensures integrity of the message as the hash value on both, sender\’s and receiver\’s side should match if the message is unaltered.

****

**Cryptanalysis:**

****

# Difference between Substitution Cipher Technique and Transposition Cipher Technique

Both **Substitution cipher technique** and **Transposition cipher technique** are the [types of Traditional cipher](https://www.geeksforgeeks.org/cryptography-traditional-symmetric-ciphers/) which are used to convert the plain text into cipher text.

**Substitution Cipher Technique:**  
In Substitution Cipher Technique plain text characters are replaced with with other characters, numbers and symbols as well as in substitution Cipher Technique, character’s identity is changed while its position remains unchanged.

**Transposition Cipher Technique:**  
Transposition Cipher Technique rearranges the position of the pain text’s characters. In transposition Cipher Technique, The position of the character is changed but character’s identity is not changed.

Difference between Substitution Cipher Technique and Transposition Cipher Technique:

|  |  |  |
| --- | --- | --- |
| **S.NO** | **SUBSTITUTION CIPHER TECHNIQUE** | **TRANSPOSITION CIPHER TECHNIQUE** |
| 1. | In substitution Cipher Technique, plain text characters are replaced with with other characters, numbers and symbols. | In transposition Cipher Technique, plain text characters are rearranged with respect to the position. |
| 2. | Substitution Cipher’s forms are: Mono alphabetic substitution cipher and poly alphabetic substitution cipher. | Transposition Cipher’s forms are: Key-less transposition cipher and keyed transposition cipher. |
| 3. | In substitution Cipher Technique, character’s identity is changed while its position remains unchanged. | While in transposition Cipher Technique, The position of the character is changed but character’s identity is not changed. |
| 4. | In substitution Cipher Technique, The letter with low frequency can detect plaint ext. | While in transposition Cipher Technique, The Keys which are nearer to correct key can disclose plain text. |
| 5. | The example of substitution Cipher is Caesar Cipher. | The example of transposition Cipher is Reil Fence Cipher. |

# Difference between Block Cipher and Stream Cipher

Prerequisite – [Block cipher modes of operation](https://www.geeksforgeeks.org/computer-network-block-cipher-modes-of-operation/)  
Both **Block Cipher** and **Stream Cipher** are belongs to the symmetric key cipher. These two block cipher and stream cipher are the methods used for converting the plain text into cipher text.

The main difference between **Block cipher** and **Stream cipher** is that block cipher converts Converts the plain text into cipher text by taking plain text’s block at a time. While stream cipher Converts the plaint text into cipher text by taking 1 byte of plain text at a time.

Substitution techniques:

1.Ceaser cipher

2. Play-fair cipher

**Ceaser Cipher:** The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.  
Thus to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down.  
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 1, B = 1,…, Z = 26. Encryption of a letter by a shift n can be described mathematically as.

En(x)=(x+n)mod26

Dn(x)=(x-n)mod26

Example: PT=HELLO,N=4

C(H)=(8+4)mod26=12

L=12

C(E)=(5+4)mod26=9

I=9

C(L)=(12+4)mod26=16

P=16

C(O)=(15+4)mod26=19

S=19

Ans=LIPPS

**Plaiy-Fair**:

The Playfair cipher encrypts pairs of letters (digraphs), instead of single letters as is the case with simpler substitution ciphers such as the Caesar Cipher..

The playfair cipher starts with creating a key table. The key table is a 5×5 grid of letters that will act as the key for encrypting your plaintext. Each of the 25 letters must be unique and one letter of the alphabet (usually Q) is omitted from the table (as there are 25 spots and 26 letters in the alphabet).

Let’s say we wanted to use the phrase “Hello World” as our key. The first characters (going left to right) in the table will be the phrase, with duplicate letters removed. The rest of the table will be filled with the remaining letters of the alphabet, in order. Our key table would look like this:

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

Now, we need a message to encrypt. In a playfair cipher the message is split into digraphs, pairs of two letters. If there is an odd number of letters, a Z is added to the last letter. Let’s say we want to encrypt the message “hide the gold”.

HI DE TH EG OL DZ

Now for the actual encryption process. The Playfair cipher uses a few simple rules relating to where the letters of each digraph are in relation to each other. The rules are:

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

Using these rules, the result of the encryption of “hide the gold” with the key of “hello world” would be “LF GD MW DN WO CV”.

**Transposition Techniques**:

1.Railfence Cipher

2.Raw Transposition Cipher

**Railfence Cipher**: In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

* In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
* When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus the alphabets of the message are written in a zig-zag manner.
* After each alphabet has been written, the individual rows are are combined to obtain the cipher-text.

Example:- Plain Text=”welcome to my session”

|  |
| --- |
| W l o e o y e s o |
| e c m t m s s i n |
| CT=wloeoyesoecmtmssin |

**Row Transposition ciphers**

* In general write message in a number of columns and then use some rule to read off from these columns
* key could be a series of number being the order to: read off the cipher; or write in the plain-text
* Example:- Plain Text=”welcome to my session”
* Key = 0 to9(32451)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 2 | 4 | 5 | 1 |
| w | E | l | c | o |
| m | E | t | o | m |
| y | S | e | s | s |
| i | O | n | x | y |

Cipher Text=omsyeesowmyiltencosx

**BLOCK CIPHER PRINCIPLES**

Virtually, all symmetric block encryption algorithms in current use are based on a structure referred to as Fiestel block cipher. For that reason, it is important to examine the design principles of the Fiestel cipher. We begin with a **comparison of stream cipher with block** **cipher.**

**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.

**1 Block cipher principles**

. most symmetric block ciphers are based on a **Feistel Cipher Structure**

. needed since must be able to **decrypt** ciphertext to recover messages efficiently

block ciphers look like an extremely large substitution

 would need table of 264 entries for a 64-bit block

.instead create from smaller building blocks

 using idea of a product cipher in 1949 Claude Shannon introduced idea of substitution-permutation (S-P) networks called modern substitution-transposition product cipher these form the basis of modern block ciphers

 S-P networks are based on the two primitive cryptographic operations we have seen before:

*substitution*(S-box)

*permutation*(P-box)

 provide *confusion* and *diffusion* of message

**diffusion**–dissipates statistical structure of plaintext over bulk of ciphertext

**confusion**–makes relationship between ciphertext and key as complex as possible

**Feistel cipher structure**

 The input to the encryption algorithm are a plaintext block of length 2w bits and a key K. the plaintext block is divided into two halves L0 and R0. The two halves of the data pass through „n‟ rounds of processing and then combine to produce the ciphertext block. Each round „i‟ has inputs Li-1 and Ri-1, derived from the previous round, as well as the subkey Ki, derived from the overall key K. in general, the subkeys Ki are different from K and from each other.

 All rounds have the same structure. A substitution is performed on the left half of the data (as similar to S-DES). This is done by applying a round function F to the right half of the data and then taking the XOR of the output of that function and the left half of the data. The round function has the same general structure for each round but is parameterized by the round subkey ki. Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data. This structure is a particular form of the substitution-permutation network.

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

**Block size**- Increasing size improves security, but slows cipher

**Key size**- Increasing size improves security, makes exhaustive key searching harder, butmay slow cipher

**Number of rounds**- Increasing number improves security, but slows cipher

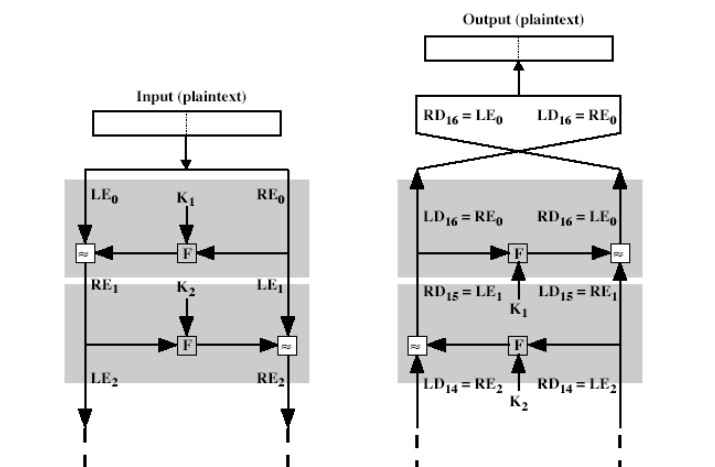
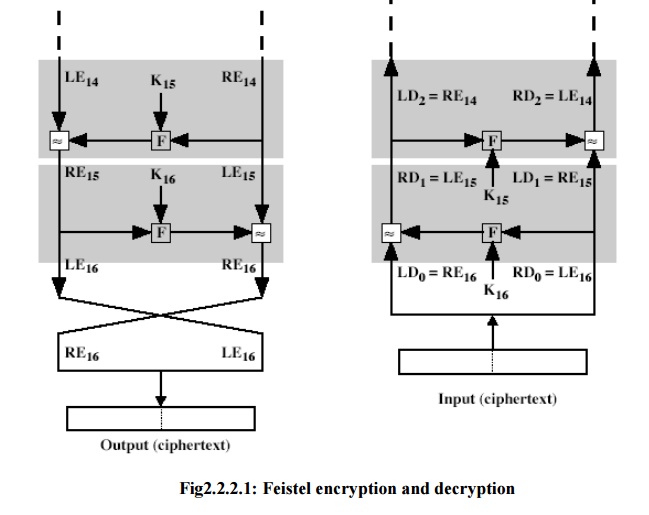
**Subkey generation**- Greater complexity can make analysis harder, but slows cipher

**Round function**- Greater complexity can make analysis harder, but slows cipher

**Fast software en/decryption & ease of analysis -**are more recent concerns for practical useand testing.

 The process of decryption is essentially the same as the encryption process. The rule is as follows: use the cipher text as input to the algorithm, but use the subkey ki in reverse order. i.e., kn in the first round, kn-1 in second round and so on. For clarity, we use the notation LEi and REi for data traveling through the decryption algorithm. The diagram below indicates that, at each round, the intermediate value of the decryption process is same (equal) to the corresponding value of the encryption process with two halves of the value swapped.

 i.e., REi || LEi (or) equivalently RD16-i || LD16-i

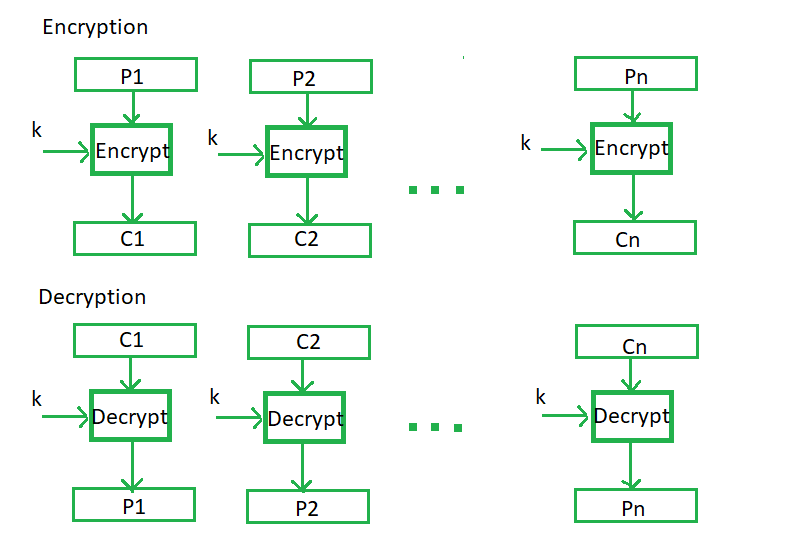
 After the last iteration of the encryption process, the two halves of the output are swapped, so that the cipher text is RE16 || LE16. The output of that round is the cipher text. Now take the cipher text and use it as input to the same algorithm. The input to the first round is RE16 || LE16, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process. 

# Block Cipher modes of Operation

Encryption algorithms are divided into two categories based on input type, as block cipher and stream cipher. **Block cipher** is an encryption algorithm which takes fixed size of input say b bits and produces a ciphertext of b bits again. If input is larger than b bits it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

**Electronic Code Book (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 bunch of blocks and the procedure is repeated.

Procedure of ECB is illustrated below:



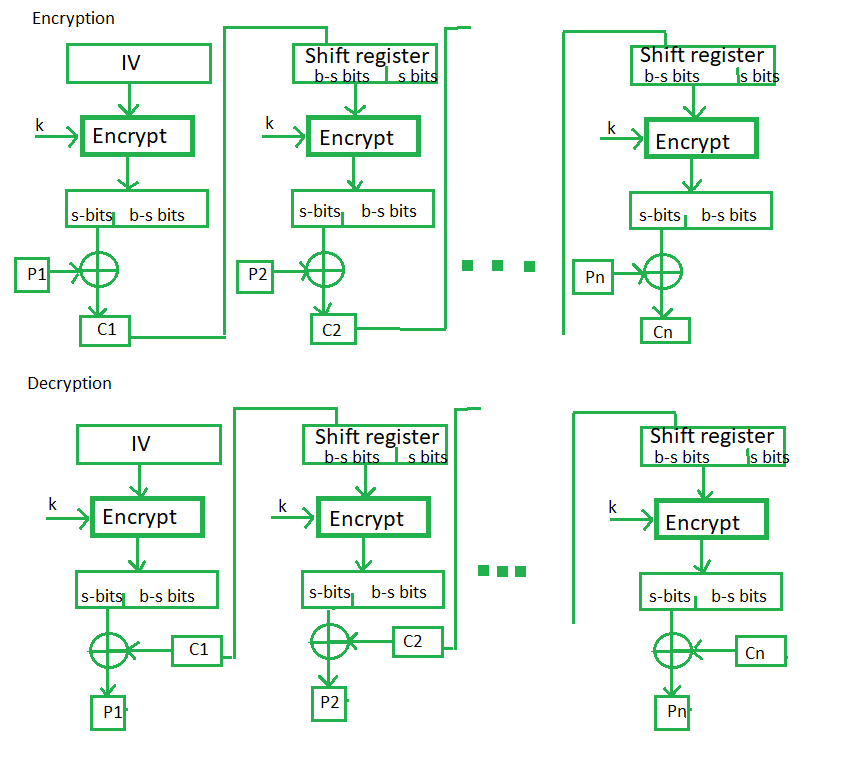
**Advantages of CBC –**

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

**Disadvantages of CBC –**

* Parallel encryption is not possible since every encryption requires previous cipher.

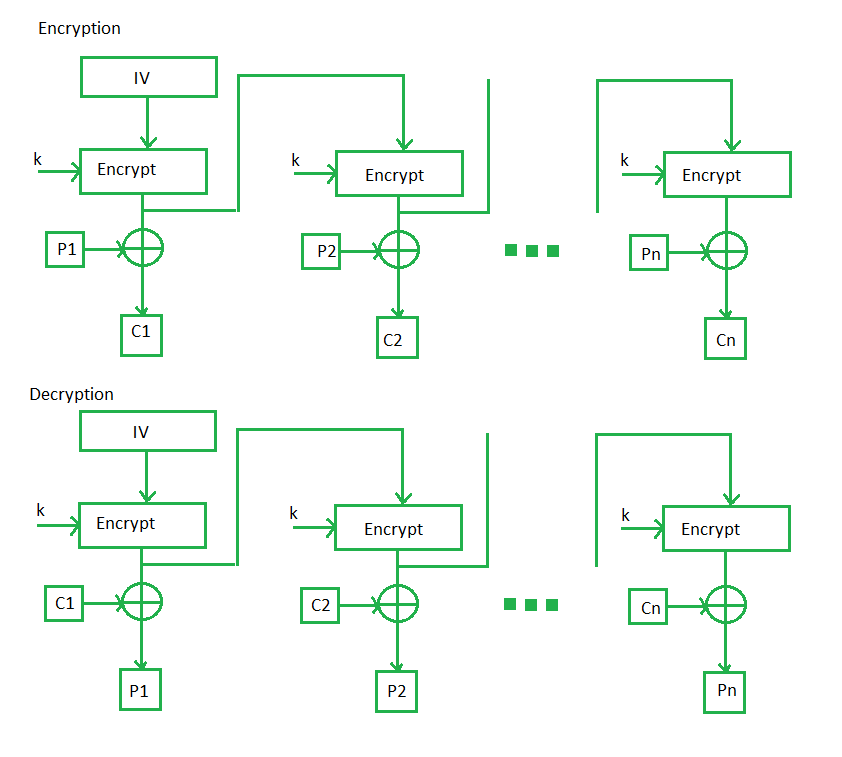
**Cipher Feedback Mode (CFB) –**  
In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first an initial vector IV is used for first encryption and output bits are divided as set of *s*and*b-s* bits the left hand side *s*bits are selected and are applied an XOR operation with plaintext bits. The result given as input to a shift register and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithm.



**Advantages of CFB –**

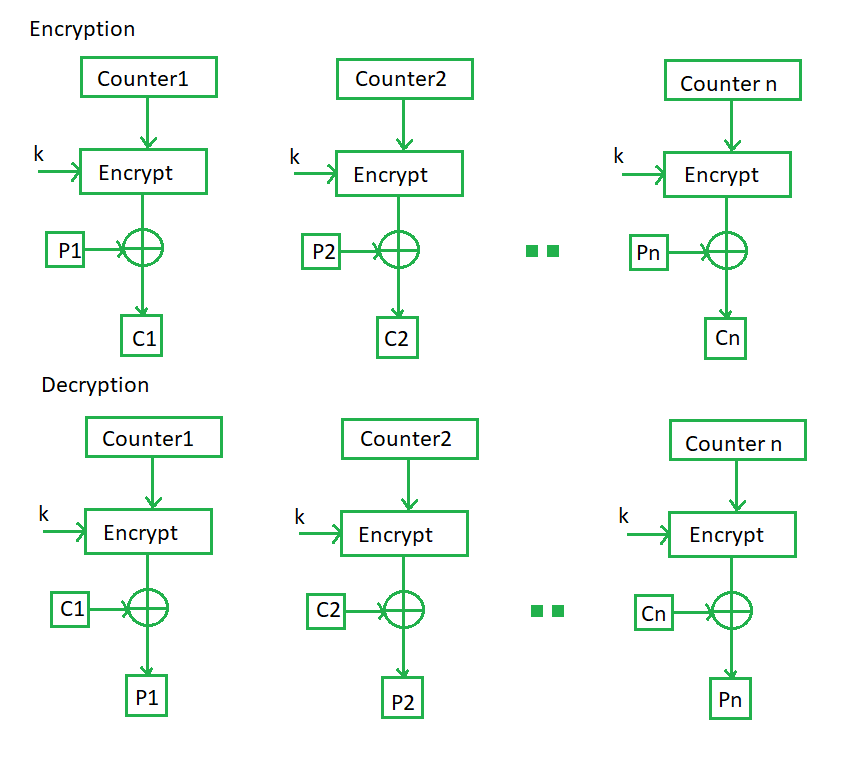
* Since, there is some data loss due to use of shift register, thus it is difficult for applying cryptanalysis.

**Output Feedback Mode –**  
The output feedback mode follows nearly 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. In this output feedback mode, all bits of the block are send instead of sending selected *s* bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases dependency or relationship of cipher on plaintext.



**Counter Mode –**  
The Counter Mode or CTR is a simple counter based block cipher implementation. 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.

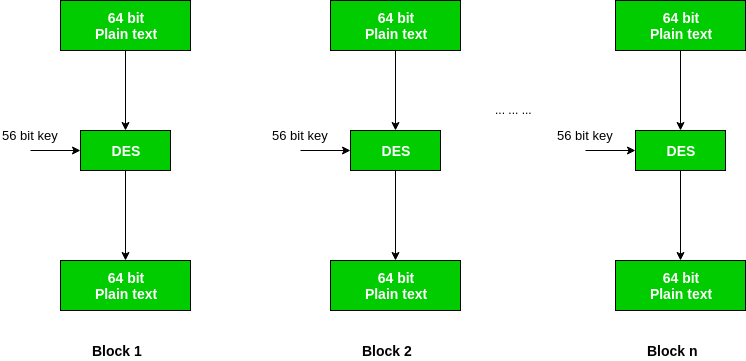
Its simple implementation is shown below:



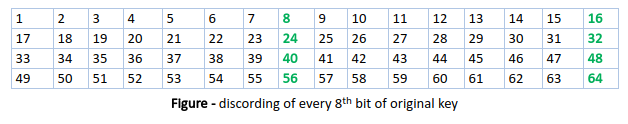
# Data encryption standard (DES) | Set 1

**Data encryption standard (DES)** has been found vulnerable against very powerful attacks and therefore, the popularity of DES has been found slightly on decline.

DES is a block cipher, and encrypts data in blocks of size of 64 bit each, means 64 bits of plain text goes as the input to DES, which produces 64 bits of cipher text. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits. The basic idea is show in figure.



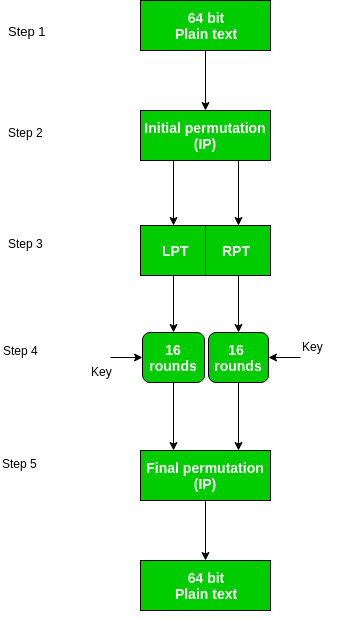
We have mention that DES uses a 56 bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56 bit key. That is bit position 8, 16, 24, 32, 40, 48, 56 and 64 are discarded.



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 as confusion) and transposition (also called as diffusion). DES consists of 16 steps, each of which is called as a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

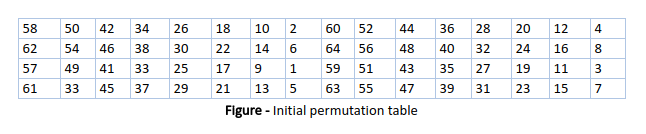
1. In the first step, the 64 bit plain text block is handed over to an initial Permutation (IP) function.
2. The initial permutation performed on plain text.
3. Next the initial permutation (IP) produces two halves of the permuted block; says Left Plain Text (LPT) and Right Plain Text (RPT).
4. Now each LPT and RPT to go through 16 rounds of encryption process.
5. In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
6. The result of this process produces 64 bit cipher text.



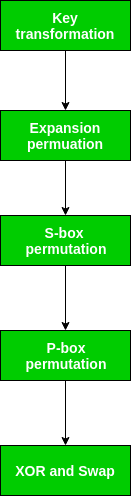
**Initial Permutation (IP) –**  
As we have noted, the Initial permutation (IP) happens only once and it happens before the first round. It suggests how the transposition in IP should proceed, as show in figure.

For example, it says that the IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block and so on.

This is nothing but jugglery of bit positions of the original plain text block. the same rule applies for all the other bit positions which shows in the figure.

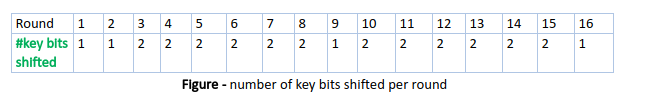


As we have noted after IP done, the resulting 64-bit permuted text block is divided into two half blocks. Each half block consists of 32 bits, and each of the 16 rounds, in turn, consists of the broad level steps outlined in figure.

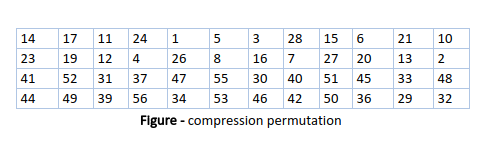


**Step-1: Key transformation –**  
We have noted initial 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key. Thus, for each a 56-bit key is available. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called as key transformation. For this the 56 bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round.

For example, if the round number 1, 2, 9 or 16 the shift is done by only position for other rounds, the circular shift is done by two positions. The number of key bits shifted per round is show in figure.

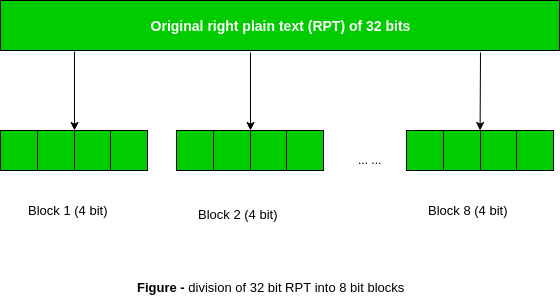


After an appropriate shift, 48 of the 56 bit are selected. for selecting 48 of the 56 bits the table show in figure given below. For instance, after the shift, bit number 14 moves on the first position, bit number 17 moves on the second position and so on. If we observe the table carefully, we will realize that it contains only 48 bit positions. Bit number 18 is discarded (we will not find it in the table), like 7 others, to reduce a 56-bit key to a 48-bit key. Since the key transformation process involves permutation as well as selection of a 48-bit sub set of the original 56-bit key it is called Compression Permutation.



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

**Step-2: Expansion Permutation –**  
Recall that after initial permutation, we had two 32-bit plain text areas called as Left Plain Text(LPT) and Right Plain Text(RPT). During the expansion permutation, the RPT is expanded from 32 bits to 48 bits. Bits are permuted as well hence called as expansion permutation. This happens as the 32 bit RPT is divided into 8 blocks, with each block consisting of 4 bits. Then, each 4 bit block of the previous step is then expanded to a corresponding 6 bit block, i.e., per 4 bit block, 2 more bits are added.



This process results into expansion as well as permutation of the input bit while creating output. Key transformation process compresses the 56-bit key to 48 bits. Then the expansion permutation process expands the 32-bit RPT to 48-bits. Now the 48-bit key is XOR with 48-bit RPT and resulting output is given to the next step, which is the S-Box substitution.

## Definition - What does *Triple DES* mean?

Triple Data Encryption Standard (DES) is a type of computerized cryptography where block cipher algorithms are applied three times to each data block. The key size is increased in Triple DES to ensure additional security through encryption capabilities. Each block contains 64 bits of data. Three keys are referred to as bundle keys with 56 bits per key. There are three keying options in data encryption standards:

1. All keys being independent
2. Key 1 and key 2 being independent keys
3. All three keys being identical

Key option #3 is known as triple DES. The triple DES key length contains 168 bits but the key security falls to 112 bits.

### ****Difference Between Linear and Differential Cryptanalysis****

### ****Linear Cryptanalysis****

* **Linear cryptanalysis** first defined by **Matsui** and **Yamagishi** in 1992. It was extended **Matsui** later in 1993 published a linear attack on **DES.**
* **Linear cryptanalysis** is a known plaintext attack in which cryptanalyst access larger plaintext and ciphertext messages along with an encrypted unknown key.
* In a **linear Cryptanalysis**, the role of cryptanalyst is to identify the linear relation between some bits of the plaintext, some bits of the ciphertext and some bits of the unknown key. This relation helps cryptanalyst to understand the logic used during encryption and decryption. the decryption of messages and to find how many bits of messages undergoes for encryption.
* There are two basic approaches. The first is to use an approximation which relates some way as mentioned earlier. bits of plain text with some bits cipher text messages and user-defined key in a linear.
* The second focus on statistical analysis against one round of decrypted cipher text. The  
  cryptanalyst each cipher text using all possible sub keys for one round of encryption and studies the resulting intermediate cipher text to analyze the random result.
* The subkey obtained during this pro and dec g this process called as candidate key used during encryption of a large amount of data.

### ****Differential Cryptanalysis****

* **Differential cryptanalysis** is a method for breaking certain classes of cryptosystems. It was invented in 1990 by Israeli researchers **Eli Biham** and **Adi Shamir.**
* **Differential cryptanalysis** is available to obtain clues about some bits of the key, thereby shortening an exhaustive search. By analyzing the changes in some chosen plaintexts, and the difference in the outputs resulting from encrypting each one, it is possible to recover some properties of the key.
* **Differential cryptanalysis** is a chosen plaintext attack which identifies a relationship between ciphertexts produced by same plaintexts.
* The differential analysis focuses on a statistical analysis of two inputs and two outputs of a cryptographic algorithm. ***For example***, assume that the ciphertext obtained from one exclusive-or operation of plain text and key.
* Without knowing the value of the key, the cryptanalyst can easily find the differences between plaintext and ciphertext. Plaintext difference is represented by**P1⊕P2.**
* Whereas the ciphertext difference represented by **C1⊕C2**. The following proves that **C1⊕C2 =P1⊕P2** First ciphertext C1 obtained = First plaintext P1 ⊕ Key K
* Second ciphertext C2 obtained = Second plaintext P2 **⊕**Key K, if C1 and C2 obtained from XORing P1 and P2 and using Key K, can be represented by,

C1⊕C2=P1⊕K⊕P2⊕K=P1⊕P2

* Differential cryptanalysis and linear cryptanalysis attacks are related to each other basically used in symmetric key cryptography. Whatever ciphertext produced from the same plain text the multiple rounds of encryption applied using for each round.
* Subkey Cryptanalyst studies changes to the intermediate cipher text obtained between multiple rounds of encryption. The attacks can be combined, which is called differential-linear cryptanalysis.

AES:

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six time faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows −

* Symmetric key symmetric block cipher
* 128-bit data, 128/192/256-bit keys
* Stronger and faster than Triple-DES
* Provide full specification and design details
* Software implementable in C and Java

## Operation of AES

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix −

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

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

### C:\Users\Rajnikant\Desktop\aes_structure.jpg

## Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −



### 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.

### Shiftrows

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 −

* First row is not shifted.
* Second row is shifted one (byte) position to the left.
* Third row is shifted two positions to the left.
* Fourth row is shifted three positions to the left.
* The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

### MixColumns

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.

### Addroundkey

The 16 bytes of the matrix are now considered as 128 bits and are XORed 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.

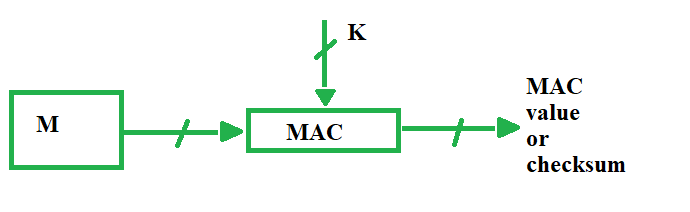
## AES Analysis

In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered. Additionally, AES has built-in flexibility of key length, which allows a degree of ‘future-proofing’ against progress in the ability to perform exhaustive key searches.

However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

# How message authentication code works?

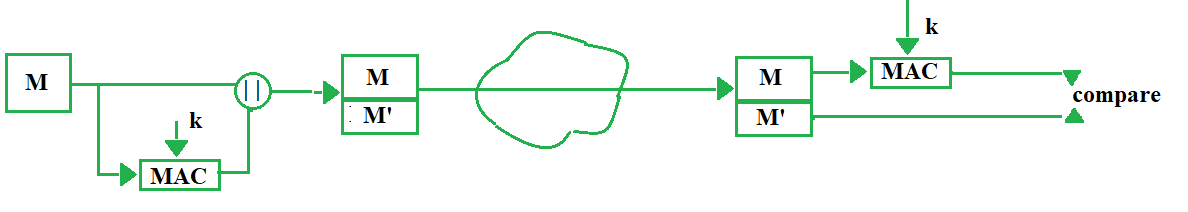
Prerequisite – [Message authentication codes](https://www.geeksforgeeks.org/message-authentication-codes/)  
Apart from intruders, the transfer of message between two people also faces other external problems like noise, which may alter the original message constructed by the sender. To ensure that the message is not altered there’s this cool method MAC.



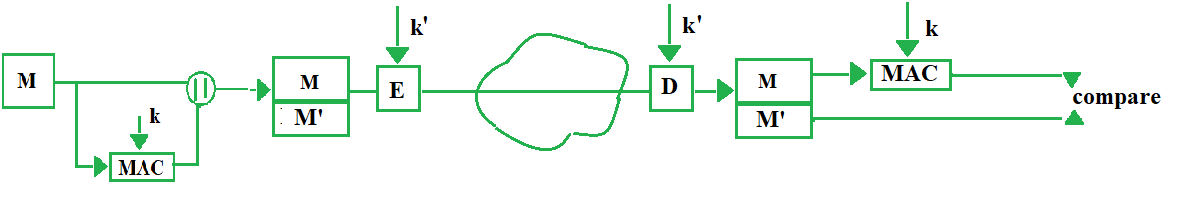
MAC stands for Message Authentication Code. Here in MAC, sender and receiver share same key where sender generates a fixed size output called Cryptographic checksum or Message Authentication code and appends it to the original message. On receiver’s side, receiver also generates the code and compares it with what he/she received thus ensuring the originality of the message. These are components:

* Message
* Key
* MAC algorithm
* MAC value

There are different types of models Of Message Authentication Code (MAC) as following below:

1. **MAC without encryption –**  
   This model can provide authentication but not confidentiality as anyone can see the message.  
   
2. **Internal Error Code –**  
   In this model of MAC, sender encrypts the content before sending it through network for confidentiality. Thus this model provides confidentiality as well as authentication.

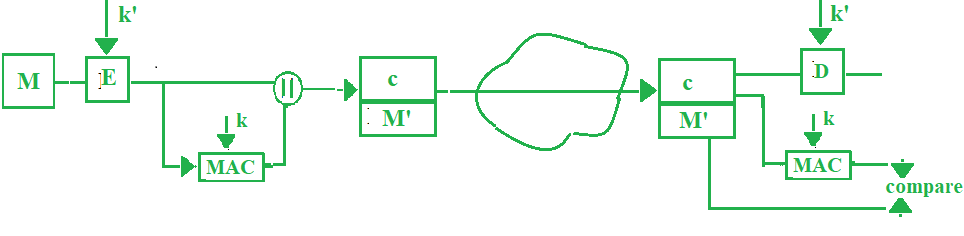
M' = MAC(M, k)



1. **External Error Code –**  
   For cases when there is an alteration in message, we decrypt it for waste, to overcome that problem, we opt for external error code. Here we first apply MAC on the encrypted message ‘c’ and compare it with received MAC value on the receiver’s side and then decrypt ‘c’ if they both are same, else we simply discard the content received. Thus it saves time.

c = E(M, k')

M' = MAC(c, k)



**Problems in MAC –**  
If we do reverse engineering we can reach plain text or even the key. Here we have mapped input to output, to overcome this we move on to hash functions which are “One way”.