**LAB #1**

**ENCRYPTION TECHNIQUE:** There are two basic building block of encryption technique.

1. Substitution Cipher ii) Transposition Cipher

**SUBSTITUTION CIPHER:**

Any character of the plain text is replaced by the other character,number or symbols.

Ex#

ABC ---> XYZ

**TRANSPOSITION CIPHER:**

Performing some sort of permutation on the plain text letters i.e. rearrangement of of the letter of the plain text.

Ex# ABC (3!)

ABC, BAC,CAB,BCA,ACB,CBA

**TYPES of SUBSTITUTION CIPHER:**

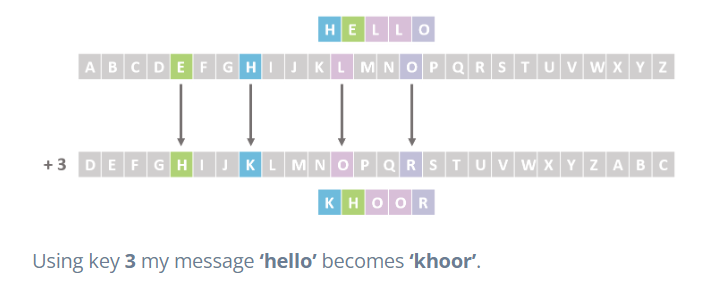
1. **CAESER CIPHER:**

**Introduction:**

It is also called shift cipher or additive cipher. Julius caeser used additive cipher to communicate with his officer. For this reason additive cipher are sometimes called caeser cipher. He used key of 3 for communication.

**Working Mechanism:**

It works by shifting a letters in the plaintext message by certain positions known as shift or key. For Example with a shift of 2 A would be replaced by C or with a shift 3 Z would be replaced by C.



**Encryption: (convert plaintext into ciphertext)**

**C = E(P, K) = (P+K) MOD 26**

**Decryption: (covert ciphertext into plaintext)**

**P = D(C,K) = (C+K) MOD 26**

### ****Features of Caesar Cipher:****

**Substitution cipher:** The Caesar cipher is a type substitution cipher, where each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

**Fixed key:** The Caesar cipher uses a fixed key, which is the number of positions by which the letters are shifted. This key is known to both the sender and the receiver.

**Symmetric encryption:** The Caesar cipher is a symmetric encryption technique, meaning that the same key is used for both encryption and decryption.

**Limited keyspace:** The Caesar cipher has a very limited keyspace of only 26 possible keys, as there are only 26 letters in the English alphabet.

**Vulnerable to brute force attacks:** The Caesar cipher is vulnerable to  brute force attack, as there are only 26 possible keys to try.

**Easy to implement:** The Caesar cipher is very easy to implement and requires only simple arithmetic operations, making it a popular choice for simple encryption tasks.

**IMPLEMENTATION**

#include <iostream>

#include <string>

#include <map>

using namespace std;

map<char,int> mp;

//Encode msg

string encryption(string plain\_text,int key){

string cipher\_text="";

for(char c: plain\_text){

if (mp.find(c)!=mp.end()){

int encrypted\_value=(mp[c]+key)%26;

for(auto value:mp){

if(value.second==encrypted\_value){

cipher\_text+=value.first;

break;

}

}

}else{

cipher\_text+=c;

};

}

cout<<"Encoded message is: "<<cipher\_text<<endl;

return cipher\_text;

}

//Decode msg

void decryption(string cipher\_text,int key){

string plain\_text="";

for (char c:cipher\_text){

if (mp.find(c)!=mp.end()){

int decrypted\_value=((mp[c]-key)+26)%26;

for(auto value:mp){

if(value.second==decrypted\_value){

plain\_text+=value.first;

break;

}

}

}else{

plain\_text+=c;

}

}

cout<<"Decoded message is: "<<plain\_text;

}

//define map to store data in key:value pair

void alphabet() {

mp['a'] = 0;

mp['b'] = 1;

mp['c'] = 2;

mp['d'] = 3;

mp['e'] = 4;

mp['f'] = 5;

mp['g'] = 6;

mp['h'] = 7;

mp['i'] = 8;

mp['j'] = 9;

mp['k'] = 10;

mp['l'] = 11;

mp['m'] = 12;

mp['n'] = 13;

mp['o'] = 14;

mp['p'] = 15;

mp['q'] = 16;

mp['r'] = 17;

mp['s'] = 18;

mp['t'] = 19;

mp['u'] = 20;

mp['v'] = 21;

mp['w'] = 22;

mp['x'] = 23;

mp['y'] = 24;

mp['z'] = 25;

}

int main(){

// input string and shift key

string input\_;

int shift;

cout << "Enter Plain text: ";

getline(cin,input\_);

cout << "Enter shift key: ";

cin >> shift;

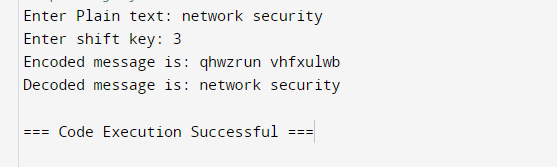
alphabet();

decryption(encryption(input\_,shift),shift);

return 0;

}

**OUTPUT:**

****