**IMPLEMENTATION OF CRYPTOGRAPHIC SOLUTIONS USING**

**SYMMETRIC ENCRYPTION**

**AIM**: To implement Caesar Cipher, Shift Cipher and the Brute Force attack on Shift Cipher.

**CODE**:

CAESAR CIPHER

#include <iostream>

#include <string>

using namespace std;

string Encrypt(string message, int key) {

    string encryptedMessage = "";

    for (char ch : message) {

        if (isupper(ch)) {

            encryptedMessage += char(int(ch + key - 65) % 26 + 65);

        }

        else if (islower(ch)) {

            encryptedMessage += char(int(ch + key - 97) % 26 + 97);

        }

        else {

            encryptedMessage += ch;

        }

    }

    return encryptedMessage;

}

int main() {

    string plaintext;

    int key;

    cout << "Enter the plaintext message: ";

    getline(cin, plaintext);

    //Setting key to be 3 (for Ceasar Cipher)

    key=3;

    string ciphertext = Encrypt(plaintext, key);

    cout << "Encrypted ciphertext: " << ciphertext << endl;

    return 0;

}

OUTPUT:



SHIFT CIPHER

#include <iostream>

#include <string>

using namespace std;

string encrypt(string plaintext, int shift) {

    string encryptedText = "";

    for (size\_t i = 0; i < plaintext.length(); i++) {

        if (isupper(plaintext[i]))

            encryptedText += char(int(plaintext[i] + shift - 65) % 26 + 65);

        else if (islower(plaintext[i]))

            encryptedText += char(int(plaintext[i] + shift - 97) % 26 + 97);

        else

            encryptedText += plaintext[i];

    }

    return encryptedText;

}

string decrypt(string ciphertext, int shift) {

    string decryptedText = "";

    for (size\_t i = 0; i < ciphertext.length(); i++) {

        if (isupper(ciphertext[i]))

            decryptedText += char(int(ciphertext[i] - shift - 65 + 26) % 26 + 65);

        else if (islower(ciphertext[i]))

            decryptedText += char(int(ciphertext[i] - shift - 97 + 26) % 26 + 97);

        else

            decryptedText += ciphertext[i];

    }

    return decryptedText;

}

int main() {

    int shiftKey;

    string message;

    cout << "Enter the message to encrypt: ";

    getline(cin, message);

    cout << "Enter the shift key: ";

    cin >> shiftKey;

    // Encryption part

    string encryptedMessage = encrypt(message, shiftKey);

    cout << "Encrypted message: " << encryptedMessage << endl;

    // Decryption part

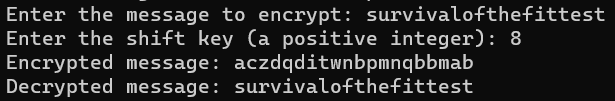
    string decryptedMessage = decrypt(encryptedMessage, shiftKey);

    cout << "Decrypted message: " << decryptedMessage << endl;

    return 0;

}

OUTPUT:



BRUTE FORCE ATTACK (ON SHIFT CIPHER)

#include <iostream>

#include <string>

using namespace std;

string decrypt(string ciphertext, int shift) {

    string decryptedText = "";

    for (size\_t i = 0; i < ciphertext.length(); i++) {

        if (isupper(ciphertext[i]))

            decryptedText += char(int(ciphertext[i] - shift - 65 + 26) % 26 + 65);

        else if (islower(ciphertext[i]))

            decryptedText += char(int(ciphertext[i] - shift - 97 + 26) % 26 + 97);

        else

            decryptedText += ciphertext[i];

    }

    return decryptedText;

}

int main() {

    string encryptedMessage;

    cout << "Enter the encrypted message: ";

    getline(cin, encryptedMessage);

    // Perform brute-force attack

    cout << "Check the results:" << endl;

    for (int shiftKey = 1; shiftKey <= 25; shiftKey++) {

        string decryptedMessage = decrypt(encryptedMessage, shiftKey);

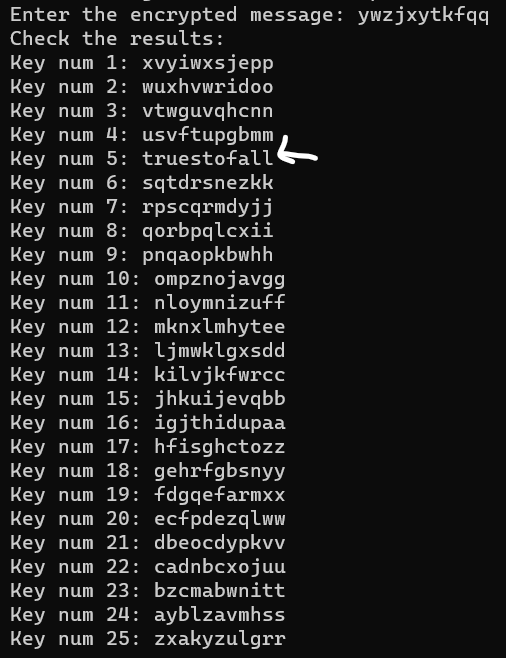
        cout << "Key num " << shiftKey << ": " << decryptedMessage << endl;

    }

    return 0;

}

OUTPUT:



RESULT:

Thus, Caesar Cipher, Shift Cipher and Brute Force attack have been implemented accordingly.