**PANDIT DEENDAYAL ENERGY UNIVERSITY**

**SCHOOL OF TECHNOLOGY**



**Course: Information Security Lab**

**Course Code: 20CP304P**

**LAB MANUAL**

**B.Tech. (Computer Science and Engineering)**

**Semester 5**

|  |  |
| --- | --- |
| **Submitted To:** | **Submitted By:** |
| Dr. Hargeet Kaur | Ravi Jamanbhai Makwana |
|  | 21BCP418 |
|  | G12 batch |

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **List of experiments** | **Date** | **Sign** |
| **1** | Download and practice Cryptool. | 7/8/23 |  |
| **2** | Study and Implement program for Caeser Cipher with Encryption, Decryption functions. | 7/8/23 |  |
| **3** | Study and implement a program for Transposition (Columnar) Cipher to encrypt and decrypt the message. | 14/8/23 |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |
| **7** |  |  |  |
| **8** |  |  |  |
| **9** |  |  |  |
| **10** |  |  |  |

**Experiment-1**

**Aim:** Download and Practice Cryptool.

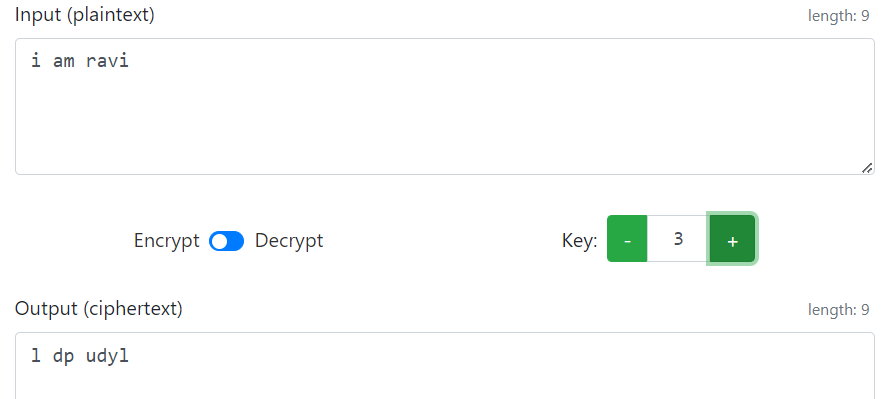
**Introduction:**

Cryptool is an open-source project that is a free e-learning software for illustrating cryptographic and cryptanalytic concepts.

Cryptool stands as a dynamic and adaptable software suite, positioned at the forefront of modern cryptography investigation and analysis. Its intuitive interface and meticulously crafted cryptographic tools empower both newcomers and seasoned professionals to immerse themselves in the captivating realm of encryption, decryption, and cryptographic analysis. Whether one's goal is to master the foundational concepts of cryptography, evaluate algorithm security, or untangle intricate historical ciphers, Cryptool serves as an essential platform for hands-on exploration, educational pursuits, and scholarly research. By seamlessly melding theoretical understanding with practical engagement, Cryptool emerges as a vital companion for unravelling the intricacies of cryptographic techniques, reinforcing digital security, and cultivating a deeper comprehension of the intricate craft of code crafting and deciphering.

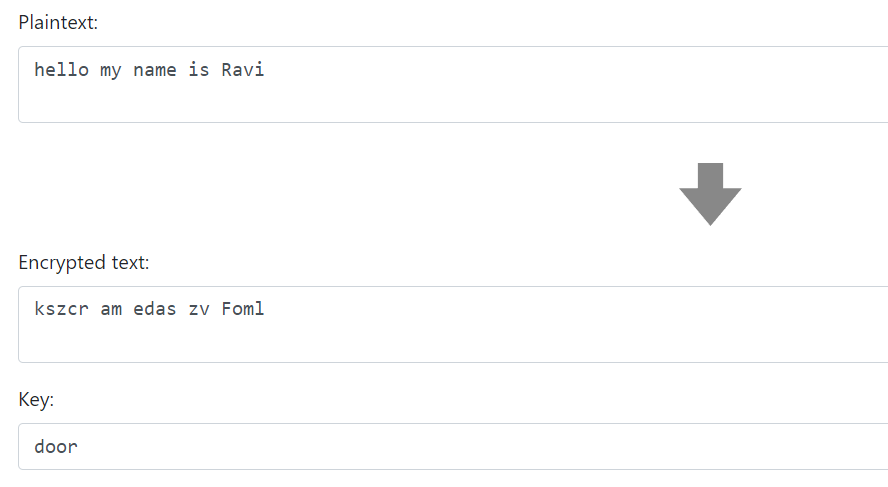
**PRACTICE**

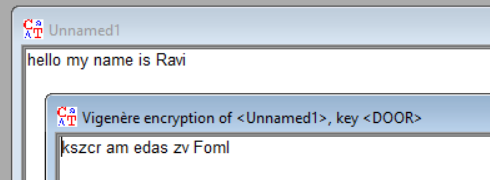
1. **Ceasar Cipher:**



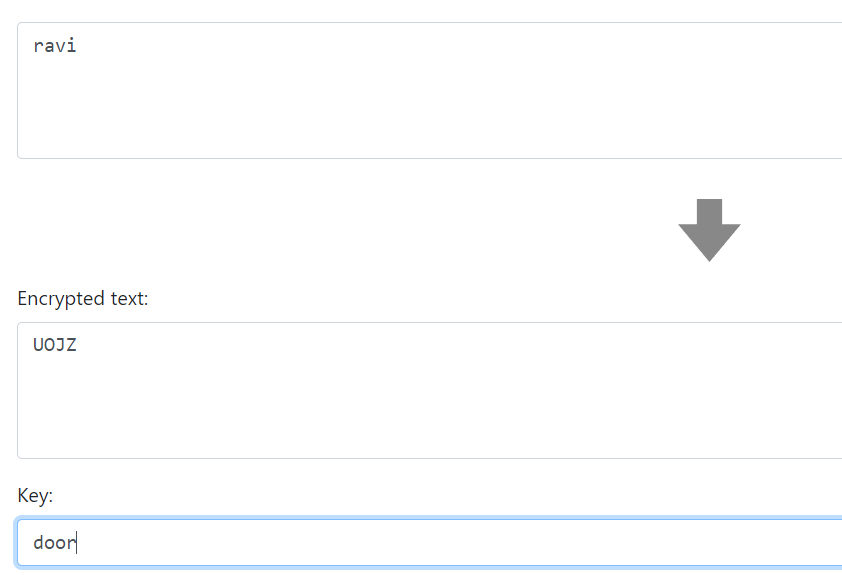


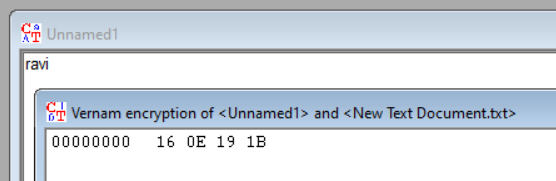
**2) Vigenère Cipher:**



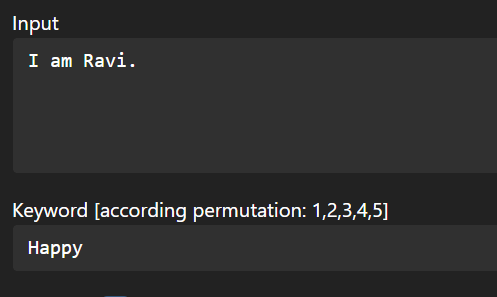


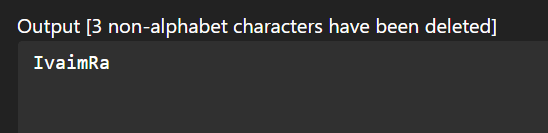
**3) Vernam Cipher:**

****

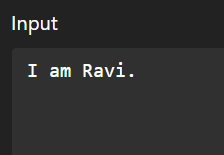
****

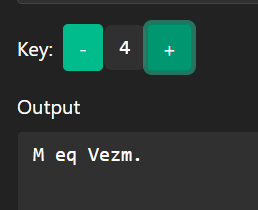
**4) Simple Columnar Transposition Cipher:**





**5) Mono alphabetic Substitution Cipher:**





**Experiment-2**

**Aim:** Study and Implement program for Caeser Cipher with Encryption, Decryption functions.

**Introduction:**

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code, or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

It works as:

* **Key (Shift Value):** A key is chosen, which determines the number of positions each letter is shifted. For example, if the key is 3, then each letter in the plaintext will be replaced with the letter that is 3 positions down the alphabet.
* **Encryption:** To encrypt a message, each letter in the plaintext is replaced by the letter located a certain number of positions down the alphabet. If the shift value is positive (as in the original Caesar Cipher), the letters wrap around from the end of the alphabet to the beginning if necessary.

For example, using a key of 3:

'A' becomes 'D’, ‘B' becomes 'E', 'C' becomes 'F' ….'Z' becomes 'C'

* **Decryption:** To decrypt a message encrypted with the Caesar Cipher, you perform the reverse process. Each letter in the ciphertext is shifted back by the same key value to reveal the original plaintext.

**Program (Source Code):**

#include <bits/stdc++.h>

using namespace std;

string cipher(string P, int key){

    string C;

    // converting original text to lowercase

    for (int i=0;i<P.length();i++){

        P[i] = tolower(P[i]);

    }

    // converting to cipher text

    for (int i=0;i<P.length();i++){

        char ch;

        if (P[i] != ' '){

            ch = (int(P[i]) + (key % 26) - 97) % 26 + 97;

        }

        else{

            ch = ' ';

        }

        C += ch;

    }

    return C;

}

string decipher(string P, int key){

    string C;

    // converting original text to lowercase

    for (int i=0;i<P.length();i++){

        P[i] = tolower(P[i]);

    }

    // converting to cipher text

    for (int i=0;i<P.length();i++){

        char ch;

        if (P[i] != ' '){

            ch = (int(P[i]) - (key % 26) - 97 + 26) % 26 + 97;

        }

        else{

            ch = ' ';

        }

        C += ch;

    }

    return C;

}

int main(){

    string plainText = "my name is ravi";

    int key = 3;

    string cipherText = cipher(plainText, key);

    cout<<"Encrypted Text: "<<cipherText;

    cout<<"\n";

    string decyp = decipher(cipherText, key);

    cout<<"Decrypted Text: "<<decyp;

    return 0;

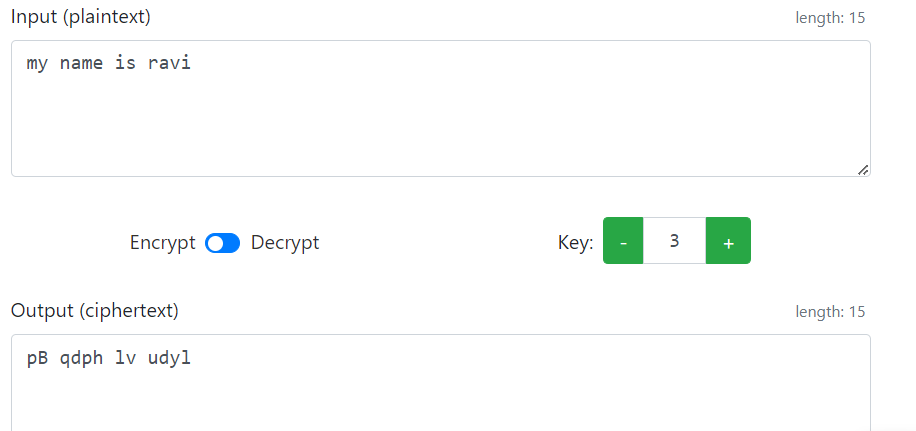
}

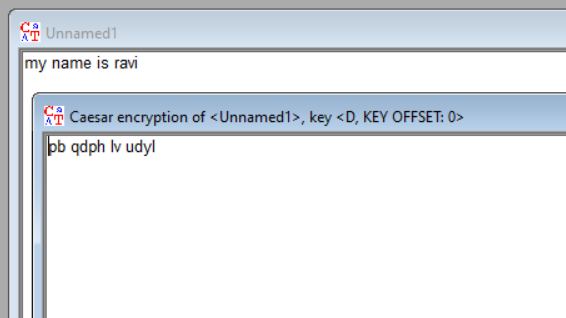
**Output (Program):**



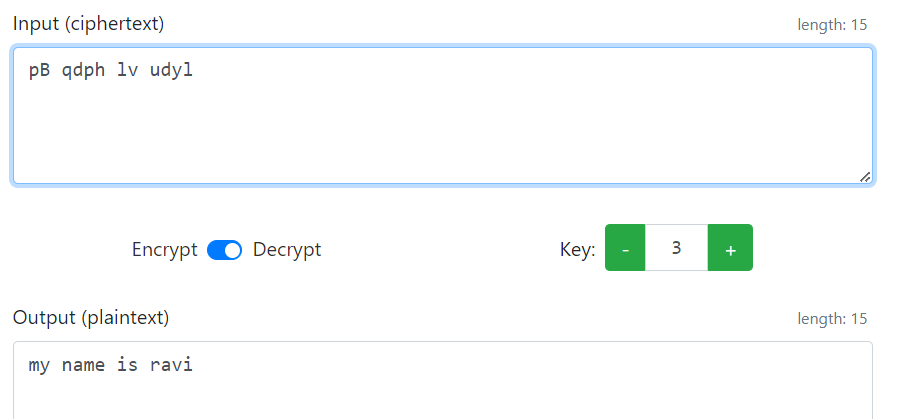
**Output (Cryptool):**

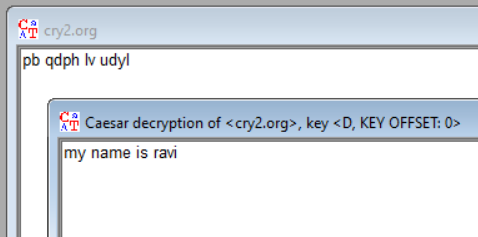
**Encryption:**





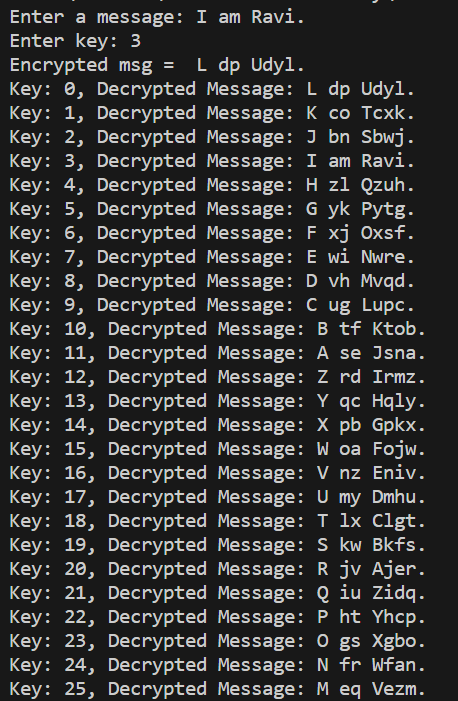
**Decryption:**

****



**Cryptanalysis:**

**Brute force Analysis:**



**2. Frequency Analysis:** Even though the Caesar cipher obscures the original letters, it does not change the frequency distribution of letters in the text. For example, in English, the letter "E" is the most used letter. An attacker can analyze the frequency of letters in the ciphertext and make educated guesses about the key based on the known frequency distribution of letters in the language.

**3. Known-Plaintext Attack:** If the attacker has some knowledge of the plaintext content or can make educated guesses about parts of the original text, they can use this information to narrow down possible keys. By aligning the known plaintext with the ciphertext, they can deduce the key.

**Applications:**

The Caesar cipher, although not suitable for strong security, can still have applications in:

* Education: It is a great tool to introduce beginners to the concepts of encryption and decryption in cryptography.
* Puzzles and Challenges: In recreational settings, it can add an element of mystery to puzzles, games, or coding challenges.
* Historical Context: It can be used in historical re-enactments to demonstrate ancient cryptographic techniques.
* Introductory Coding: It can be used to teach basic programming skills involving strings and loops.
* Steganography: As a component in larger steganographic methods where information is hidden within other media.

**References:**

1. W3Schools
2. GeeksforGeeks