

Module 7: Ethical Hacking Lab

1. Using Cryptool to encrypt and decrypt password

Perform encryption and decryption of text using cryptool2. Using cryptool2 to perform following

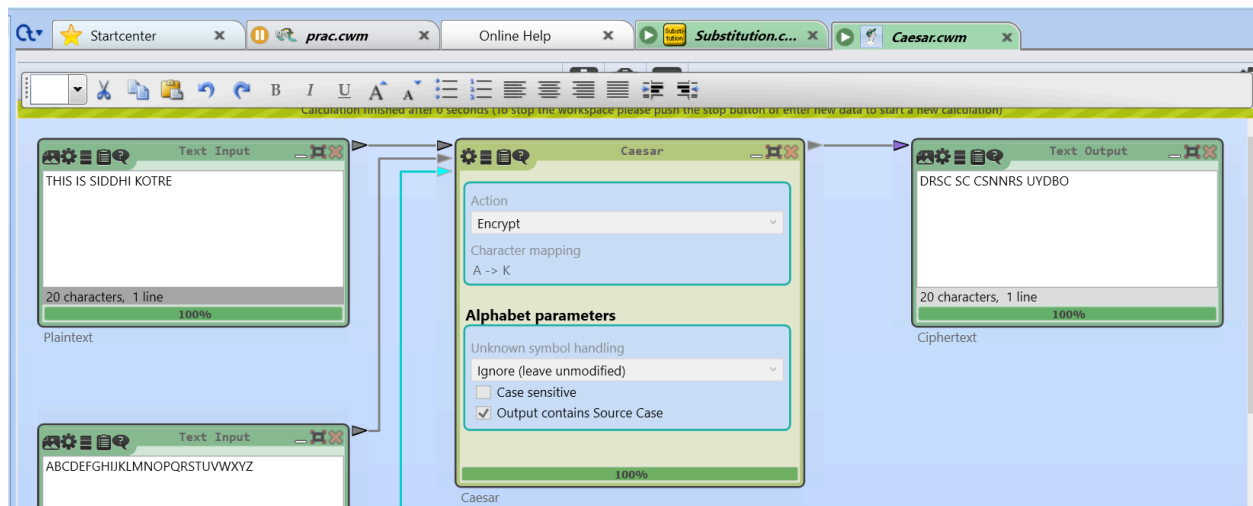
CrypTool 2 (CT2) is an educational software platform that allows users to explore and experiment with various cryptographic techniques and algorithms interactively. It supports a wide range of encryption, decryption, cryptanalysis, and cryptographic functions, making it ideal for students, educators, and enthusiasts who want to learn about classical and modern cryptography in a hands-on environment.

1. Caesar cipher

Theory:

The Caesar Cipher is a basic encryption technique that shifts each letter in the plaintext by a fixed number of positions down or up the alphabet, with the shift value serving as the key. Named after Julius Caesar, who reportedly used it to protect his messages, it's simple but easily broken due to its small key space. This cipher is primarily educational today, highlighting foundational ideas in cryptography.

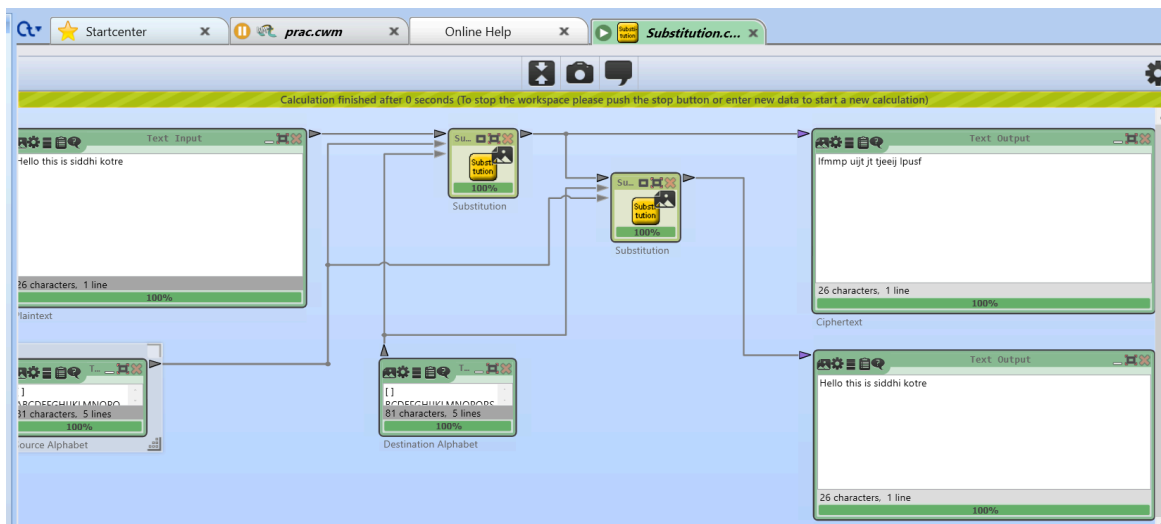
Output:



2.Substitution cipher

Theory:

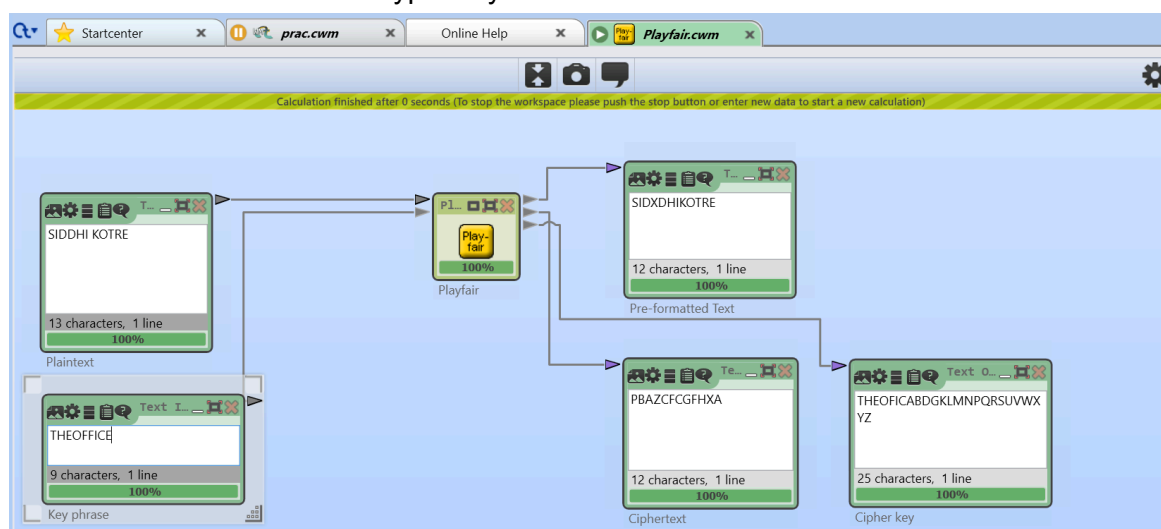
The Substitution Cipher replaces each letter in the plaintext with another letter according to a substitution alphabet. Unlike Caesar's fixed shift, this cipher uses an arbitrary mapping, providing a larger key space and thus greater security. However, because it preserves letter frequencies, it remains vulnerable to frequency analysis, where common letters in the language can reveal patterns in the ciphertext.



3.Playfair cipher

Theory:

The Playfair Cipher encrypts pairs of letters (digraphs) rather than single letters, using a 5x5 matrix created from a keyword. By encrypting in pairs, it obscures single-letter frequencies, making it more secure than a simple substitution cipher. Though still susceptible to digraph frequency analysis, the Playfair Cipher represents an important step in making substitution ciphers more resistant to basic cryptanalysis.



1. **Aim:** Write a code in java for encryption and decryption using caesar cipher.

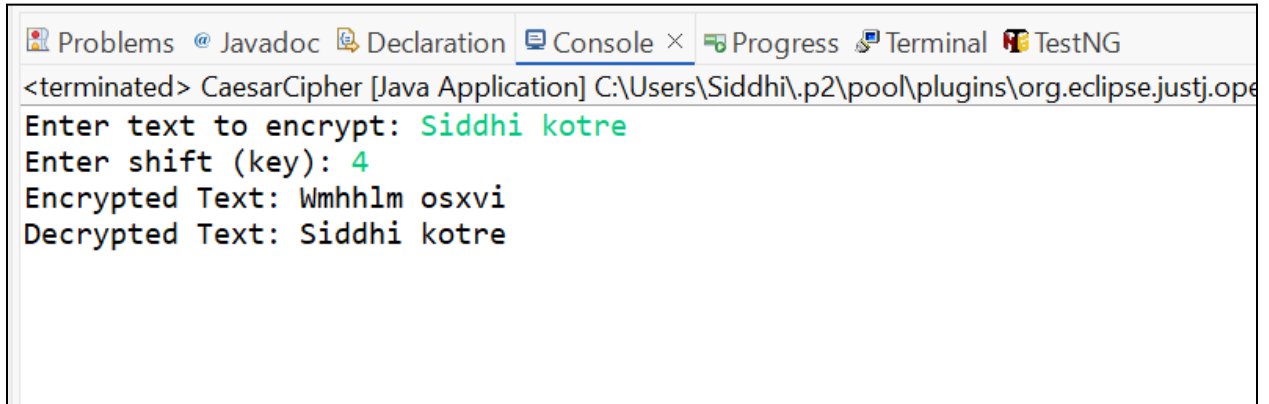
Code:

```
package practical;
import java.util.Scanner;
public class CaesarCipher {
    public static String encrypt(String text, int shift) {
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < text.length(); i++) {
            char ch = text.charAt(i);

            if (Character.isUpperCase(ch)) {
                char encryptedChar = (char) (((ch - 'A' + shift) % 26) + 'A');
                result.append(encryptedChar);
            }

            else if (Character.isLowerCase(ch)) {
                char encryptedChar = (char) (((ch - 'a' + shift) % 26) + 'a');
                result.append(encryptedChar);
            } else {
                result.append(ch);
            }
        }
        return result.toString();
    }
    public static String decrypt(String text, int shift) {
        return encrypt(text, 26 - shift);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter text to encrypt: ");
        String text = scanner.nextLine();
        System.out.print("Enter shift (key): ");
        int shift = scanner.nextInt();
        String encryptedText = encrypt(text, shift);
        System.out.println("Encrypted Text: " + encryptedText);
        String decryptedText = decrypt(encryptedText, shift);
        System.out.println("Decrypted Text: " + decryptedText);
        scanner.close();
    }
}
```

Output:



The screenshot shows the Eclipse IDE's console window. The title bar includes tabs for Problems, Javadoc, Declaration, Console (active), Progress, Terminal, and TestNG. The console output for the application 'CaesarCipher [Java Application]' is as follows:

```
<terminated> CaesarCipher [Java Application] C:\Users\Siddhi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre\bin\java.exe
Enter text to encrypt: Siddhi kotre
Enter shift (key): 4
Encrypted Text: Wmhhlm osxvi
Decrypted Text: Siddhi kotre
```