Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

з лабораторної роботи № 2

з дисципліни «Захист інформації в комп’ютерних системах»

на тему: «Перестановочний шифр»

Виконав: ст. гр. КІ-302

Радевич-Винницький Я.А.

Перевірив:

Муляревич О.В.

**Мета роботи:** ознайомитись з основами перестановочної техніки шифрування.

**Завдання:**

1. Створити програму, що реалізує довільний перестновочний алгоритм шифрування.

**Варіант:** 22.

**Виконання завдання:**

Алгоритм роботи перестановочного шифру:

Алгоритм використовує перетворення «драбинки». Відкритий текст записується уздовж похилих рядків певної довжини, а потім зчитується построчно по горизонталі. Наприклад, щоб зашифрувати повідомлення “computer engineering” по методу драбинки зі сходами довжиною 2, запишемо це повідомлення у вигляді

c m u e e g n e i g

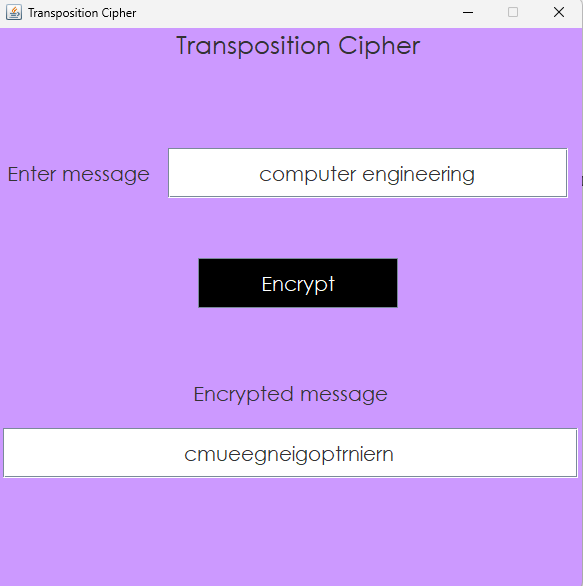
o p t r n i e r n

Шифроване повідомлення буде мати такий вигляд:

CMUEEGNEIGOPTRNIERN

Для виконання завдання було вибрано мову Java та бібліотеку Swing для створення графічного інтерфейсу додатку.

*Програма – Transposition Cipher*



*Рис. 2 – вікно програми*

Код файлу *TranspositionCipher.java*, у якому міститься реалізація заданого алгоритму шифрування:

Лістинг 1

|  |
| --- |
| package application.encryptor;  public class TranspositionCipher {  private static final int *LAYER* = 2;  private static final int *FIRST\_BUILDER* = 0;  private static final int *SECOND\_BUILDER* = 1;   public String encrypt(String message) {  StringBuilder[] stringBuilders = new StringBuilder[*LAYER*];  for (int i = 0; i < *LAYER*; i++) {  stringBuilders[i] = new StringBuilder();  }  char[] messageCharArray = message.replaceAll("\\s", "").toCharArray();  for (int i = 0; i < messageCharArray.length; i++) {  if (i % 2 == 0) {  stringBuilders[*FIRST\_BUILDER*].append(messageCharArray[i]);  } else {  stringBuilders[*SECOND\_BUILDER*].append(messageCharArray[i]);  }  }  return stringBuilders[*FIRST\_BUILDER*].append(stringBuilders[*SECOND\_BUILDER*]).toString();  } } |

Код файлу *Frame.java*, у якому міститься реалізація код графічного інтерфейсу програми:

Лістинг 2

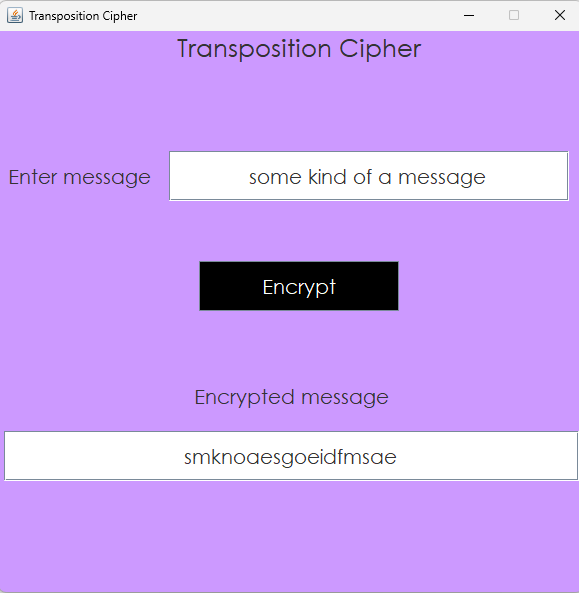
|  |
| --- |
| package application.gui;  import application.encryptor.TranspositionCipher;  import javax.swing.\*; import java.awt.\*; import java.awt.event.ActionEvent; import java.awt.event.ActionListener;  public class Frame extends JFrame implements ActionListener {  private TranspositionCipher transpositionCipher;   private static final String *FRAME\_TITLE* = "Transposition Cipher";  private static final int *DIMENSION* = 600;   private JFrame frame;   private JLabel headLabel;  private JLabel inputMessageLabel;  private JLabel outputMessageLabel;   private JTextField inputTextField;  private JTextField outputTextField;   private JButton encryptionButton;   public Frame() {  headLabel = new JLabel();  adjustHeadLabelSettings(headLabel);   inputMessageLabel = new JLabel();  adjustInputMessageLabelSettings(inputMessageLabel);   inputTextField = new JTextField();  adjustInputTextFieldSettings(inputTextField);   encryptionButton = new JButton();  adjustEncryptButtonSettings(encryptionButton);   outputMessageLabel = new JLabel();  adjustOutputMessageLabelSettings(outputMessageLabel);   outputTextField = new JTextField();  adjustOutputTextFieldSettings(outputTextField);   frame = new JFrame();  adjustFrameSettings(frame);   frame.add(headLabel);  frame.add(inputMessageLabel);  frame.add(inputTextField);  frame.add(encryptionButton);  frame.add(outputMessageLabel);  frame.add(outputTextField);  }   private void adjustFrameSettings(JFrame frame) {  frame.setTitle(*FRAME\_TITLE*);  frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  frame.setResizable(false);  frame.setSize(*DIMENSION*,*DIMENSION*);  frame.getContentPane().setBackground(new Color(204,153,255));  frame.setLayout(null);  frame.setVisible(true);  }   private void adjustHeadLabelSettings(JLabel headLabel) {  headLabel.setText(*FRAME\_TITLE*);  headLabel.setFont(new Font("Century Gothic", Font.*PLAIN*, 25));  headLabel.setVerticalAlignment(JLabel.*TOP*);  headLabel.setHorizontalAlignment(JLabel.*CENTER*);  headLabel.setBounds(0,0, *DIMENSION*,50);  }   private void adjustInputMessageLabelSettings(JLabel inputMessageLabel) {  inputMessageLabel.setText("Enter message");  inputMessageLabel.setFont(new Font("Century Gothic", Font.*PLAIN*, 20));  inputMessageLabel.setHorizontalAlignment(JLabel.*CENTER*);  inputMessageLabel.setBounds(5, 120, 150, 50);  }   private void adjustInputTextFieldSettings(JTextField inputTextField) {  inputTextField.setPreferredSize(new Dimension(250,40));  inputTextField.setBounds(170, 120, 400, 50);  inputTextField.setFont(new Font("Century Gothic", Font.*PLAIN*, 20));  inputTextField.setHorizontalAlignment(JLabel.*CENTER*);  }   private void adjustEncryptButtonSettings(JButton encryptionButton) {  encryptionButton.setBounds(200, 230, 200, 50);  encryptionButton.setText("Encrypt");  encryptionButton.setFont(new Font("Century Gothic", Font.*PLAIN*, 20));  encryptionButton.setForeground(Color.*WHITE*);  encryptionButton.setFocusable(false);  encryptionButton.setBackground(Color.*black*);  encryptionButton.addActionListener(this);  }   private void adjustOutputMessageLabelSettings(JLabel outputMessageLabel) {  outputMessageLabel.setText("Encrypted message");  outputMessageLabel.setFont(new Font("Century Gothic", Font.*PLAIN*, 20));  outputMessageLabel.setHorizontalAlignment(JLabel.*CENTER*);  outputMessageLabel.setBounds(5, 340, 575, 50);  }   private void adjustOutputTextFieldSettings(JTextField outputTextField) {  outputTextField.setPreferredSize(new Dimension(575,50));  outputTextField.setBounds(5, 400, 575, 50);  outputTextField.setFont(new Font("Century Gothic", Font.*PLAIN*, 20));  outputTextField.setHorizontalAlignment(JLabel.*CENTER*);  }   @Override  public void actionPerformed(ActionEvent e) {  if (e.getSource().equals(encryptionButton)) {  transpositionCipher = new TranspositionCipher();  String encryptedMessage = transpositionCipher.encrypt(inputTextField.getText());  outputTextField.setText(encryptedMessage);  }  } } |

Код головного файлу програми - Main.java:

Лістинг 3

|  |
| --- |
| package application;  import application.gui.Frame;  public class Main {  public static void main(String[] args) {  Frame frame = new Frame();  } } |

Результат роботи програми:



*Рис. 3 – результат роботи програми*

**Висновок:** у ході виконання лабораторної роботи було вивчено основи перестановочної техніки шифрування. Було створено програму, що реалізує шифрування вихідного повідомлення за допомогою перестановочного шифру «драбинки» мовою програмування Java.