Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

з лабораторної роботи № 5

з дисципліни «Захист інформації в комп’ютерних системах»

на тему: «Симетричні блокові шифри на основі мережі Фейстеля»

Виконав: ст. гр. КІ-302

Радевич-Винницький Я.А.

Перевірив:

Муляревич О.В.

**Мета роботи:** ознайомитися з методом побудови алгоритмів симетричного блокового шифрування на прикладі мережі Фейстеля.

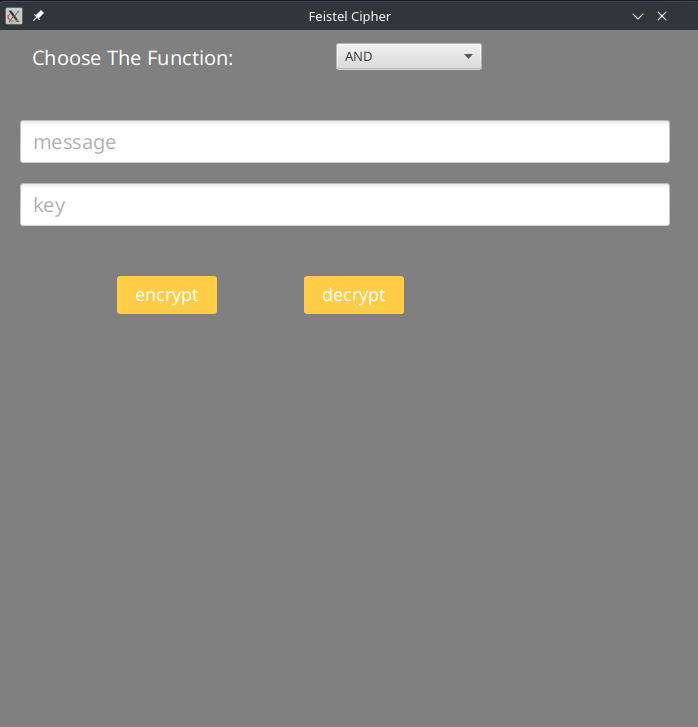
**Завдання:**

Створити програму, що реалізує симетричний блоковий алгоритм на основі мережі Фейстеля.

**Виконання завдання:**

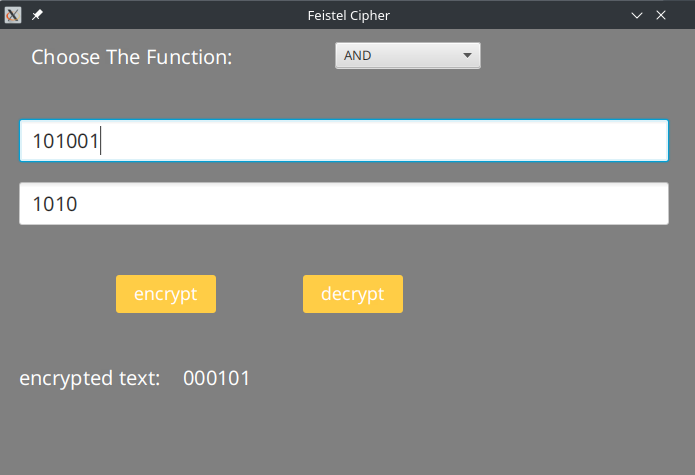
Для виконання завдання було вибрано мову Java та засоби Java FX для створення графічного інтерфейсу. Програмний код наведено в додатку.

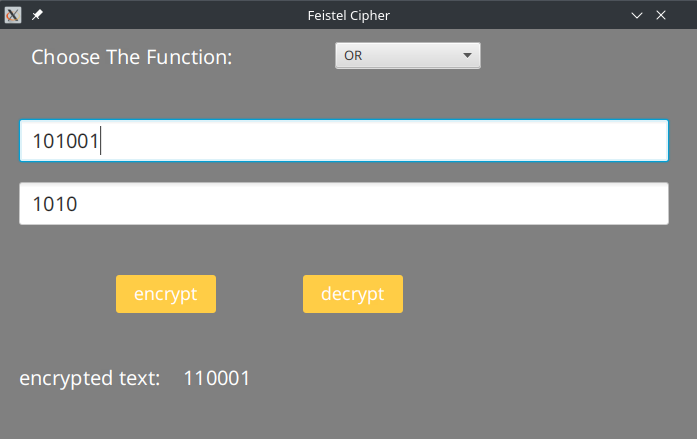
Демонстрація роботи програми:



*Рис. 1 – вікно програми*

Вікно програми пропонує користувачеві на вибір дві функції перетворення підблока на ключі: AND та OR.

*Рис. 2 – шифрування за допомогою функції AND*

*Рис. 3 – шифрування за допомогою функції OR*

**Висновок:** у ході виконання лабораторної роботи було вивчено вивчено методи побудови алгоритмів симетричного блокового шифрування на прикладі мережі Фейстеля. За допомогою мови Java та набору інструментів з платформи Java FX було створено програму, що реалізує такий алгоритм.

**Додаток**

Код файлу *FeistelCipher.java*:

Лістинг 1

|  |
| --- |
| package sample.lab5;  import java.util.ArrayList;  public class FeistelCipher {  private int initialKey;  private String functionOperator;  private int totalRound;  private ArrayList<String> keys = new ArrayList<>();  FeistelCipher(int round) {  this.totalRound = round;  keys.add("1110");  keys.add("0100");  keys.add("1101");  keys.add("0001");  keys.add("0010");  keys.add("1111");  keys.add("1011");  keys.add("1000");  keys.add("0011");  keys.add("1010");  keys.add("0110");  keys.add("1100");  keys.add("0101");  keys.add("1001");  keys.add("0000");  keys.add("0111");  }  public void setInitialKey(int initialKey) {  this.initialKey = initialKey;  }  public void setFunctionOperator(String functionOperator) {  this.functionOperator = functionOperator;  }  public String encrypt(String message) {  int messageMid = message.length() / 2;  String left = message.substring(0, messageMid);  String right = message.substring(messageMid);  for (int roundIndex = 0; roundIndex < totalRound; roundIndex++) {  String temp = right;  String functionText = function(right, roundIndex);  right = XOR(left, functionText);  left = temp;  }  return left + "" + right;  }  public String decrypt(String encryptedMessage) {  int messageMid = encryptedMessage.length() / 2;  String left = encryptedMessage.substring(0, messageMid);  String right = encryptedMessage.substring(messageMid);  for (int roundIndex = 0; roundIndex < totalRound; roundIndex++) {  String temp = left;  String functionText = function(left, totalRound - roundIndex - 1);  left = XOR(right, functionText);  right = temp;  }  return left + "" + right;  }  private String function(String right, int roundIndex) {  String currentKey = getSubKey(roundIndex);  String encryptedText = "";  switch (functionOperator) {  case "AND":  encryptedText = AND(right, currentKey);  break;  case "OR":  encryptedText = OR(right, currentKey);  break;  }  return encryptedText;  }  private String getSubKey(int roundIndex) {  int x = (initialKey + roundIndex) % 16;  return keys.get(x);  }  private String AND(String left, String right) {  StringBuilder stringBuilder = new StringBuilder();  for (int i = 0; i < left.length(); i++) {  stringBuilder.append((left.charAt(i) - '0') & (right.charAt(i) - '0'));  }  return stringBuilder.toString();  }  private String OR(String left, String right) {  StringBuilder stringBuilder = new StringBuilder();  for (int i = 0; i < left.length(); i++) {  stringBuilder.append((left.charAt(i) - '0') | (right.charAt(i) - '0'));  }  return stringBuilder.toString();  }  private String XOR(String left, String right) {  StringBuilder stringBuilder = new StringBuilder();  for (int i = 0; i < left.length(); i++) {  stringBuilder.append((left.charAt(i) - '0') ^ (right.charAt(i) - '0'));  }  return stringBuilder.toString();  }  } |

Код файлу *Controller.java*:

Лістинг 2

|  |
| --- |
| package sample.lab5;  import javafx.fxml.FXML;  import javafx.scene.control.Button;  import javafx.scene.control.ChoiceBox;  import javafx.scene.control.Label;  import javafx.scene.control.TextField;  public class Controller {  public ChoiceBox choiceBox;  public TextField messageTextField;  public TextField keyTextField;  public Button encryptButton;  public Button decryptButton;  public Label headerLabel;  public Label encryptedLabel;  FeistelCipher feistelCipher;  @FXML  private void initialize() {  feistelCipher = new FeistelCipher(16);  encryptButton.setOnMouseClicked(mouseEvent -> handleEncryptButton());  decryptButton.setOnMouseClicked(mouseEvent -> handleDecryptedButton());  }  private void handleEncryptButton() {  String functionOperator = choiceBox.getValue().toString();  String message = messageTextField.getText();  String InitialKey = keyTextField.getText();  String encryptedMessage = "";  feistelCipher.setInitialKey(Integer.parseInt(InitialKey, 2));  feistelCipher.setFunctionOperator(functionOperator);  encryptedMessage = feistelCipher.encrypt(message);  showEncryptedMessage(encryptedMessage);  }  private void handleDecryptedButton() {  String encryptedMessage = encryptedLabel.getText();  String decryptedMessage = "";  decryptedMessage = feistelCipher.decrypt(encryptedMessage);  showDecryptedMessage(decryptedMessage);  }  private void showEncryptedMessage(String encryptedMessage) {  headerLabel.setVisible(true);  encryptedLabel.setText(encryptedMessage);  messageTextField.clear();  }  private void showDecryptedMessage(String decryptedMessage) {  headerLabel.setVisible(false);  messageTextField.setText(decryptedMessage);  encryptedLabel.setText("");  }  } |

Код файлу *Main.java*:

Лістинг 3

|  |
| --- |
| package sample.lab5;  import javafx.application.Application;  import javafx.fxml.FXMLLoader;  import javafx.scene.Parent;  import javafx.scene.Scene;  import javafx.stage.Stage;  public class Main extends Application {  @Override  public void start(Stage primaryStage) throws Exception{  FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource("sample.fxml"));  Scene scene = new Scene(fxmlLoader.load(), 700, 700);  primaryStage.setTitle("Feistel Cipher");  primaryStage.setScene(scene);  primaryStage.setResizable(false);  primaryStage.show();  }  public static void main(String[] args) {  launch(args);  }  } |

Код файлу *sample.fxml:*

Лістинг 4

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?import java.lang.\*?>  <?import javafx.collections.\*?>  <?import javafx.scene.control.\*?>  <?import javafx.scene.layout.\*?>  <?import javafx.scene.text.\*?>  <VBox maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="458.0" prefWidth="547.0" style="-fx-background-color: gray;" xmlns="http://javafx.com/javafx/10.0.2-internal" xmlns:fx="http://javafx.com/fxml/1" fx:controller="sample.lab5.Controller">  <children>  <Pane layoutY="50.0" prefHeight="50.0" prefWidth="843.0">  <children>  <Label layoutX="32.0" layoutY="13.0" text="Choose The Function:" textFill="WHITE">  <font>  <Font size="20.0" />  </font>  </Label>  <ChoiceBox fx:id="choiceBox" layoutX="336.0" layoutY="13.0" prefHeight="26.0" prefWidth="146.0" value="AND">  <items>  <FXCollections fx:factory="observableArrayList">  <String fx:value="AND" />  <String fx:value="OR" />  </FXCollections>  </items>  </ChoiceBox>  </children>  </Pane>  <AnchorPane>  <children>  <TextField fx:id="messageTextField" layoutY="22.0" prefHeight="39.0" prefWidth="497.0" promptText="message" AnchorPane.bottomAnchor="10.0" AnchorPane.leftAnchor="20.0" AnchorPane.rightAnchor="30.0" AnchorPane.topAnchor="40.0">  <font>  <Font size="20.0" />  </font>  </TextField>  </children>  </AnchorPane>  <AnchorPane>  <children>  <TextField fx:id="keyTextField" layoutX="20.0" layoutY="35.0" prefHeight="39.0" prefWidth="497.0" promptText="key" AnchorPane.bottomAnchor="30.0" AnchorPane.leftAnchor="20.0" AnchorPane.rightAnchor="30.0" AnchorPane.topAnchor="10.0">  <font>  <Font size="20.0" />  </font>  </TextField>  </children>  </AnchorPane>  <AnchorPane>  <children>  <Button fx:id="encryptButton" contentDisplay="CENTER" layoutX="117.0" prefHeight="26.0" prefWidth="100.0" style="-fx-background-color: #FFCD46;" text="encrypt" textFill="white" AnchorPane.bottomAnchor="30.0" AnchorPane.leftAnchor="117.0" AnchorPane.topAnchor="20.0">  <font>  <Font size="18.0" />  </font>  </Button>  <Button fx:id="decryptButton" contentDisplay="CENTER" layoutX="304.0" prefHeight="26.0" prefWidth="100.0" style="-fx-background-color: #FFCD46;" text="decrypt" textFill="white" AnchorPane.bottomAnchor="30.0" AnchorPane.leftAnchor="304.0" AnchorPane.topAnchor="20.0">  <font>  <Font size="18.0" />  </font>  </Button>  </children>  </AnchorPane>  <AnchorPane>  <children>  <Label fx:id="headerLabel" prefHeight="50.0" prefWidth="560.0" text="encrypted text: " textFill="white" visible="false" AnchorPane.bottomAnchor="10.0" AnchorPane.leftAnchor="20.0" AnchorPane.rightAnchor="20.0" AnchorPane.topAnchor="10.0">  <font>  <Font size="20.0" />  </font>  </Label>  <Label fx:id="encryptedLabel" layoutX="184.0" prefHeight="50.0" prefWidth="343.0" textFill="WHITE" AnchorPane.bottomAnchor="10.0" AnchorPane.leftAnchor="184.0" AnchorPane.rightAnchor="20.0" AnchorPane.topAnchor="10.0">  <font>  <Font size="20.0" />  </font>  </Label>  </children>  </AnchorPane>  </children>  </VBox> |