Лабораторная работа №2

Математические основы защиты информации и информационной безопасности

Минов Кирилл Вячеславович

НПМмд-02-23 27 сентября 2023

Российский университет дружбы народов

Москва, Россия

Цель лабораторной работы

1) Реализовать на языке программирования маршрутное шифрование, шифрование с помощью решеток и таблицу Виженера

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов.

Маршрутное шифрование разработал французский математик Франсуа Виет. Открытый текст записывают в некоторую геометрическую фигуру (обычно прямоугольник) по некоторому пути, а затем, выписывая символы по другому пути, получают шифр текст.

Шифрование с помощью решеток — это метод криптографического шифрования, в котором текст сообщения разбивается на сегменты, представляющие собой сетку (решетку). Эта сетка может быть представлена в виде квадратной или прямоугольной матрицы. Создана австрийским криптографом Эдуардом Флейснером в 1881 году.

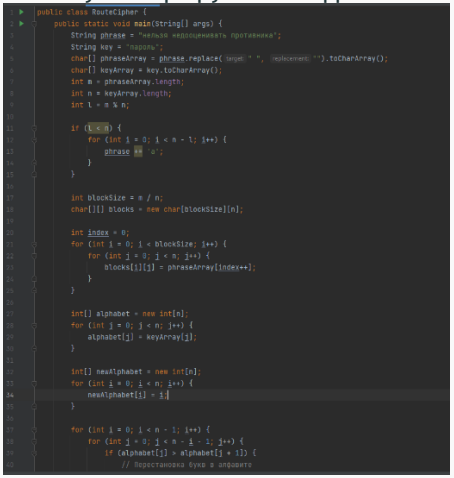
Теоретическое введение

В 1585 году французский криптограф Блез Виженер опубликовал свой метод шифрования в «Трактате о шифрах», а именно "Таблица Виженера".

Таблица Виженера, также известная как Табличный шифр Виженера, представляет собой инструмент для шифрования и дешифрования текста с использованием метода шифра Виженера. Этот метод шифрования основан на использовании ключевого слова или фразы, которое повторяется, чтобы зашифровать или дешифровать сообщение.

Ход выполнения лабораторной работы

- Реализуем маршрутное шифрование



```
public class RouteCipher {
    public static void main(String[] args) {
        String phrase = "нельзя недооценивать противника";
        String key = "пароль";
        char[] phraseArray = phrase.replace( target: " ",  replacement: "").toCharArray();
        char[] keyArray = key.toCharArray();
        int m = phraseArray.length;
        int n = keyArray.length;
        int l = m % n;

        if (l > 0) {
            for (int i = 0; i < n - l; i++) {
                phrase += 'а';
            }
        }

        int blockSize = m / n;
        char[][] blocks = new char[blockSize][n];

        int index = 0;
        for (int i = 0; i < blockSize; i++) {
            for (int j = 0; j < n; j++) {
                blocks[i][j] = phraseArray[index++];
            }
        }

        int[] alphabet = new int[n];
        for (int j = 0; j < n; j++) {
            alphabet[j] = keyArray[j];
        }

        int[] newAlphabet = new int[n];
        for (int i = 0; i < n; i++) {
            newAlphabet[i] = i;
        }

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (alphabet[j] > alphabet[j + 1]) {
                    // Перестановка букв в алфавите
```

Рис.1 Маршрутное шифрование(1)

Ход выполнения лабораторной работы

- Реализуем маршрутное шифрование

```
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (alphabet[j] > alphabet[j + 1]) {
                    // Перестановка букв в алфавите
                    int temp = alphabet[j];
                    alphabet[j] = alphabet[j + 1];
                    alphabet[j + 1] = temp;

                    // Перестановка индексов в новом алфавите
                    temp = newAlphabet[j];
                    newAlphabet[j] = newAlphabet[j + 1];
                    newAlphabet[j + 1] = temp;
                }
            }
        }

        StringBuilder result = new StringBuilder();
        for (int g = 0; g < n; g++) {
            for (int h = 0; h < blockSize; h++) {
                result.append(blocks[h][newAlphabet[g]]);
            }
        }

        System.out.println(result.toString());
    }
}
```

Рис.2: Маршрутное шифрование

Ход выполнения лабораторной работы

- Реализуем шифрование с помощью решеток

```java
public class FleissnerCipher {
    public static void main(String[] args) {
        String phrase = "Договор подписали";
        String key = "шифр";

        String encryptedText = encryptFleissner(phrase, key);
        System.out.println("Зашифрованный текст: " + encryptedText);
    }

    public static String encryptFleissner(String phrase, String key) {
        int keyLength = key.length();
        int phraseLength = phrase.length();
        char[] encryptedText = new char[phraseLength];

        for (int i = 0; i < phraseLength; i++) {
            int keyIndex = i % keyLength;
            char keyChar = key.charAt(keyIndex);
            int shift = keyChar % 32; // Диапазон символов кириллицы

            char encryptedChar = (char) (phrase.charAt(i) + shift);
            if (encryptedChar > 'я') {
                encryptedChar = (char) (encryptedChar - 32); // Обертывание по алфавиту
            }
            encryptedText[i] = encryptedChar;
        }

        return new String(encryptedText);
    }
}
```

Рис.3: Шифрование с помощью решеток

Ход выполнения лабораторной работы

- Реализуем таблицу Виженера

```
public class VigenereCipher {
    public static void main(String[] args) {
        String phrase = "криптография серьезная наука";
        String key = "математика";
        phrase = phrase.replaceAll( regex: " ",  replacement: "");
        char[] phraseArray = phrase.toCharArray();
        char[] keyArray = key.toCharArray();

        char[] alphabet = new char[32];
        for (int i = 0; i < 32; i++) {
            alphabet[i] = (char) (1072 + i);
        }

        char[][] table = new char[32][32];
        for (int i = 0; i < 32; i++) {
            for (int j = 0; j < 32; j++) {
                table[i][j] = alphabet[(j + i) % 32];
            }
        }

        int k = phraseArray.length / keyArray.length;
        StringBuilder keyList = new StringBuilder();
        for (int j = 0; j < k; j++) {
            keyList.append(key);
        }
        String partKey = key.substring(0, phraseArray.length % keyArray.length);
        keyList.append(partKey);

        char[] keyListArray = keyList.toString().toCharArray();
        char[] cipher = new char[phraseArray.length];

        for (int g = 0; g < phraseArray.length; g++) {
            int rowIndex = new String(alphabet).indexOf(phraseArray[g]);
            int colIndex = new String(alphabet).indexOf(keyListArray[g]);
            cipher[g] = table[rowIndex][colIndex];
        }

        System.out.println(new String(cipher));
    }
}
```

Рис.4: Таблица Виженера

Вывод

- В ходе выполнения данной лабораторной работы были реализованы маршрутное шифрование, шифрование с помощью решеток и таблица Виженера