

Отчет по лабораторной работе по предмету Математические основы защиты информации и информационной безопасности

Лабораторная работа №2. Шифры перестановки

Никита Андреевич Топонен

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Маршрутное шифрование	7
3.2	Шифрование с помощью решеток	7
3.3	Таблица Виженера	8
4	Выполнение лабораторной работы	10
4.1	Маршрутное шифрование	10
4.2	Таблица Виженера	13
5	Выводы	16
	Список литературы	17

Список иллюстраций

3.1	Шифрование шифром Цезаря	8
4.1	Маршрутное шифрование	13
4.2	Шифрование шифром Атбаш	15

Список таблиц

1 Цель работы

Цель работы — познакомиться с шифрами перестановки.

2 Задание

1. Реализовать маршрутное шифрование.
2. Реализовать шифрование с помощью решеток.
3. Реализовать шифрование с помощью таблицы Виженера.

3 Теоретическое введение

3.1 Маршрутное шифрование

Пусть m и n – некоторые натуральные (т.е. целые положительные) числа, каждое больше 1. Открытый текст последовательно разбивается на части (блоки) с длиной, равной произведению mn (если в последнем блоке не хватает букв, можно дописать до нужной длины произвольный их набор). Блок вписывается построчно в таблицу размерности $m \times n$ (т.е. m строк и n столбцов). Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Этот маршрут вместе с числами m и n составляет ключ шифра.

Чаще всего буквы выписывают по столбцам, которые упорядочиваются в соответствии с паролем: под таблицей подписывается слово, состоящее из n неповторяющихся букв, и столбцы таблицы нумеруются по алфавитному порядку букв пароля.

3.2 Шифрование с помощью решеток

Этот способ шифрования предложил в 1881 году австрийский криптограф Эдуард Флейснер. Выбирается натуральное число $k > 1$, и квадрат размерности $k \times k$ построчно заполняется числами $1, 2, \dots, k$. Для примера возьмем $k = 2$.

Квадрат поворачивается по часовой стрелке на 90° и размещается вплотную к предыдущему квадрату. Аналогичные действия совершаются еще два раза, так чтобы в результате из четырех малых квадратов образовался один большой с

длиной стороны $2k$.

Далее из большого квадрата вырезаются клетки с числами от 1 до k^2 , для каждого числа одна клетка. Процесс шифрования происходит следующим образом. Сделанная решетка (квадрат с прорезями) накладывается на чистый квадрат $2k \times 2k$ и в прорези по строчкам (т.е. слева направо и сверху вниз) вписываются первые буквы открытого текста. Затем решетка поворачивается на 90° по часовой стрелке и накладывается на частично заполненный квадрат, вписывание продолжается.

После третьего поворота, наложения и вписывания все клетки квадрата будут заполнены. Правило выбора прорезей гарантирует, что при заполнении квадрата буква на букву никогда не попадет. Из заполненного квадрата буквы можно выписать по столбцам, выбрав подходящий пароль.

Например, с использованием изображенной выше решетки и пароля ш и ф р открытый текст договор подписали переводится в криптограмму за пять шагов:

-	-	-	д	-	-	-	д	-	о	-	д	ш	и	ф	р
-	-	-	-	-	в	-	-	а	в	п	-	с	о	а	д
-	о	-	г	о	о	-	г	о	о	-	г	д	в	п	л
-	-	о	-	-	р	о	п	и	р	о	п	о	о	и	г
												и	р	о	п

Рис. 3.1: Шифрование шифром Цезаря

3.3 Таблица Виженера

Французский криптограф Блез Виженер (1523-1596) опубликовал свой метод в «Трактате о шифрах» в 1585 году. С тех пор на протяжении трех столетий шифр Виженера считался нераскрываемым, пока с ним не справился австриец Фридрих Казиски (в 1863 году). При этом способе шифрования открытый текст разбивается на блоки некоторой длины n . Задается ключ – последовательность из n

натуральных чисел: a_1, a_2, \dots, a_n . Затем в каждом блоке первая буква циклически сдвигается вправо по алфавиту на a_1 позиций, вторая буква – на a_2 позиций, ..., последняя – на a_n шагов.

Для лучшего запоминания, в качестве ключа обычно берут осмысленное слово, и алфавитные номера составляющих его букв используют для вычислений, связанных со сдвигами.

Из-за нехватки опытных шифровальщиков шифр Виженера с длиной блока, равной всего лишь 3, применялся в низовых звеньях русской армии в 1916 году, во время наступления Юго-Западного фронта против австро-венгерской армии – знаменитого брусиловского прорыва. Противник легко читал русские оперативные шифровки, что, в конце концов, и не позволило генералу Брусилову добиться стратегического успеха в блестяще задуманной операции.

4 Выполнение лабораторной работы

4.1 Маршрутное шифрование

В рамках данной лабораторной работы я реализовал маршрутное шифрование на языке Java. Ниже приведен код с подробными комментариями:

```
public class Routing {  
    public static void main(String[] args) {  
        // Вводим начальные данные  
        int n = 6;  
        int m = 5;  
        String message = "нельзя недооценивать противника";  
        String password = "пароль";  
  
        // Кодируем сообщение и выводим его для проверки  
        String encryptedMessage = encrypt(n, m, password, message);  
        System.out.println(encryptedMessage);  
  
        // Расшифровываем сообщение и выводим для проверки  
        String decryptedMessage = decrypt(encryptedMessage, n, m, password);  
        System.out.println(decryptedMessage);  
    }  
  
    // =====
```

```

// = Implementation
// =====

// Метод для шифрования
private static String encrypt(int n, int m, String password,
    String message) {
    // Убираем пробелы
    String preparedMessage = message.replaceAll(" ", "");
    // Добавляем символы, так чтобы последняя строка была длиной m
    int charactersToAdd = preparedMessage.length() % (n * m);
    preparedMessage = preparedMessage + "a".repeat(charactersToAdd);

    // Создаем таблицу, ключи которой упорядочены по алфавитному порядку
    // В значения будем записывать строки столбцов таблицы для ключа
    Map<Character, String> encryptionTable = new TreeMap<>();
    for (int i = 0; i < n; i++) {
        encryptionTable.put(password.charAt(i),
            getCharactersByPosition(preparedMessage, i, n, m));
    }

    // Возвращаем "склеенные" строки зашифрованной таблицы
    return String.join("", encryptionTable.values());
}

// Метод для получения столбца таблицы
private static String getCharactersByPosition(String string,
    int position, int n, int m) {
    StringBuilder result = new StringBuilder();
    for (int i = 0; i < m; i++) {

```

```

        result.append(string.charAt(i * n + position));
    }
    return result.toString();
}

// Метод расшифровки сообщения
private static String decrypt(String encryptedMessage, int n, int m,
    String password) {
    // Подготавливаем таблицу для расшифровки
    char[] route = password.toCharArray();
    Arrays.sort(route);
    Map<Character, String> decryptionMap = new HashMap<>();
    for (int i = 0; i < password.length(); i++) {
        decryptionMap.put(route[i],
            encryptedMessage.substring(i * m, i * m + m));
    }

    // Проходим по таблице и восстанавливаем сообщение
    // по полученной таблице расшифровки
    StringBuilder result = new StringBuilder();
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            result.append(decryptionMap.get(password.charAt(j)).charAt(i));
        }
    }

    return result.toString();
}
}

```

Результаты выполнения программы на иллюстрации (рис. 4.1).

```
C:\Users\Toponen\.jdk\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
ееепнпзотэаьтовокньнеьвдирияцтя
нелзянедооцениватьпротивникаа
|
Process finished with exit code 0
```

Рис. 4.1: Маршрутное шифрование

4.2 Таблица Виженера

В рамках данной лабораторной работы я реализовал шифрование с помощью таблицы Виженера на языке Java. Ниже приведен код с подробными комментариями:

```
public class Vigenere {  
    // Сдвиг. Для английского алфавита 97, для русского 1072  
    private static int bias = 0;  
    // Количество букв в алфавите. Для английского алфавита 26, для русского 33  
    private static int letters = 0;  
  
    public static void main(String[] args) {  
        // Задаем сдвиг и количество букв для английского алфавита  
        bias = 97;  
        letters = 26;  
  
        // Задаем сообщение и ключ  
        String message = "test message for vigenere encryption method";  
        String key = "mathematics";  
  
        // Кодировем сообщение и выводим его  
        String encryptedMessage = encrypt(message, key);  
    }  
}
```

```

        System.out.println(encryptedMessage);

        // Декодируем сообщение и выводим его
        String decryptedMessage = decrypt(encryptedMessage, key);
        System.out.println(decryptedMessage);
    }

    // =====
    // = Implementation
    // =====

    // Метод для кодирования
    private static String encrypt(String text, String key) {
        StringBuilder encrypt = new StringBuilder();
        int keyLen = key.length();
        // Для каждого символа в сообщении
        for (int i = 0; i < text.length(); i++) {
            // Пропускаем пробелы
            if (text.charAt(i) == ' ') {
                encrypt.append(' ');
                continue;
            }
            // Находим символ, сдвинутый вправо на соответствующий номер
            // символа ключа в алфавите, дописываем его в результат
            encrypt.append((char) (((text.charAt(i)
                + key.charAt(i % keyLen) - 2 * bias) % letters) + bias));
        }
        // Возвращаем результат
        return encrypt.toString();
    }

```

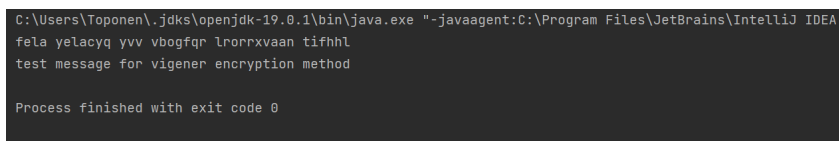
```

    }

    // Метод для декодирования
    private static String decrypt(String cipher, String key) {
        StringBuilder decrypt = new StringBuilder();
        int keyLen = key.length();
        // Для каждого символа в шифре
        for (int i = 0; i < cipher.length(); i++) {
            // Пропускаем пробелы
            if (cipher.charAt(i) == ' ') {
                decrypt.append(' ');
                continue;
            }
            // Находим символ, сдвинутый влево на соответствующий номер
            // символа ключа в алфавите, дописываем его в результат
            decrypt.append((char) (((cipher.charAt(i) -
                key.charAt(i % keyLen) + letters) % letters) + bias));
        }
        // Возвращаем результат
        return decrypt.toString();
    }
}

```

Результаты выполнения программы на иллюстрации (рис. 4.2).



```

C:\Users\Toponen\.jdk\openjdk-19.0.1\bin\java.exe --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
feLa yelacyq yvv vbogfqr lrorrxxvaan tifhhl
test message for vigenere encryption method

Process finished with exit code 0

```

Рис. 4.2: Шифрование шифром Атбаш

5 Выводы

В рамках данной лабораторной работы я познакомился с шифрами перестановки, такими как маршрутное шифрование, шифрование с помощью решеток и таблица Виженера. Также реализовал шифры на языке Java.

Список литературы