Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет информатики и

Радиоэлектроники

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

по курсу “Средства и методы защиты информации в интеллектуальных системах”

Вариант 3

Выполнил:

Студент гр. 321702 Рублевская Е. А.

Проверил: Захаров В. В.

Минск

2025

**Тема:** Простейшие криптографические преобразования

**Задание для самостоятельного выполнения:**

1. Реализовать в виде программы шифр (зашифрование и расшифрование) в соответствии с вариантом. Язык исходного текста русский или английский по выбору исполнителя.

- Использовать шифр Скитала;

1. Реализовать в виде программы атаку полным перебором ключа, используя для оценки правильности выбора ключа визуальный метод или исходный текст для автоматического сравнения результата дешифрования.
2. Оценить криптографическую стойкость реализованного шифра.
3. Предложить варианты усложнения шифра. Предложенные варианты оформить в виде алгоритма.

**Выполнение:**

**Код программы:**

**package org.example;**

**public class ScytaleCipher {**

**// Метод шифрования**

**public static String encrypt(String plaintext, int rows, int columns) {**

**plaintext = plaintext.replaceAll("\\s+", "").toUpperCase(); // Удаляем пробелы и приводим к верхнему регистру**

**char[][] table = new char[rows][columns];**

**int index = 0;**

**// Заполнение таблицы по строкам**

**for (int i = 0; i < rows; i++) {**

**for (int j = 0; j < columns; j++) {**

**if (index < plaintext.length()) {**

**table[i][j] = plaintext.charAt(index++);**

**} else {**

**table[i][j] = 'X'; // Заполнение пустых мест символом 'X'**

**}**

**}**

**}**

**// Считывание текста по столбцам для получения криптотекста**

**StringBuilder ciphertext = new StringBuilder();**

**for (int j = 0; j < columns; j++) {**

**for (int i = 0; i < rows; i++) {**

**ciphertext.append(table[i][j]);**

**}**

**}**

**return ciphertext.toString();**

**}**

**// Метод расшифрования**

**public static String decrypt(String ciphertext, int rows, int columns) {**

**char[][] table = new char[rows][columns];**

**int index = 0;**

**// Заполнение таблицы по столбцам**

**for (int j = 0; j < columns; j++) {**

**for (int i = 0; i < rows; i++) {**

**if (index < ciphertext.length()) {**

**table[i][j] = ciphertext.charAt(index++);**

**}**

**}**

**}**

**// Считывание текста по строкам для получения открытого текста**

**StringBuilder plaintext = new StringBuilder();**

**for (int i = 0; i < rows; i++) {**

**for (int j = 0; j < columns; j++) {**

**plaintext.append(table[i][j]);**

**}**

**}**

**return plaintext.toString().replaceAll("X+$", ""); // Убираем символы-заполнители**

**}**

**public static void main(String[] args) {**

**String plaintext = "Katyusha is the funniest girl";**

**int rows = 4;**

**int columns = 8;**

**// Шифрование**

**String ciphertext = encrypt(plaintext, rows, columns);**

**System.out.println("Зашифрованный текст: " + ciphertext);**

**// Расшифрование**

**String decryptedText = decrypt(ciphertext, rows, columns);**

**System.out.println("Расшифрованный текст: " + decryptedText);**

**}**

**}**

**package org.example;**

**public class ScytaleAttack {**

**// Метод для атаки с перебором размеров таблицы**

**public static void bruteForceAttack(String ciphertext) {**

**int length = ciphertext.length();**

**System.out.println("Атака полным перебором размеров таблицы:");**

**for (int rows = 1; rows <= length; rows++) {**

**if (length % rows == 0) {**

**int columns = length / rows;**

**String decryptedText = ScytaleCipher.decrypt(ciphertext, rows, columns);**

**System.out.println("Попытка с размером таблицы (" + rows + " x " + columns + "): " + decryptedText);**

**}**

**}**

**}**

**public static void main(String[] args) {**

**String ciphertext = "Katyusha is the funniest girl";**

**bruteForceAttack(ciphertext);**

**}**

**}**

**Основные функции:**

**Класс ScytaleAttack:**

Класс отвечает за выполнение атак на зашифрованные тексты, основанных на методе полного перебора размеров таблицы.

* **bruteForceAttack(String ciphertext)**  
  Метод выполняет атаку путём перебора всех возможных комбинаций строк и столбцов до расшифровки текста. Он определяет длину ciphertext, перебирает все возможные сочетания размеров таблицы (все делители длины текста).Для каждого варианта производит расшифровку с помощью метода ScytaleCipher.decrypt, при этом замеряя и выводя время выполнения для каждой итерации.

**Класс ScytaleCipher:**

Класс реализует функционал шифрования и расшифровки сообщений с использованием алгоритма Скиталы.

* **encrypt(String plaintext, int rows, int columns)**  
  Метод выполняет шифрование открытого текста методом Скиталы. Удаляет пробелы из plaintext и переводит текст в верхний регистр, потом создаёт таблицу размерами rows x columns, заполняет ее символами текста по строкам. Если текста недостаточно, оставшиеся ячейки таблицы заполняются символом **'X'**. Формирует зашифрованный текст считыванием символов из таблицы по столбцам.
* **decrypt(String ciphertext, int rows, int columns)**  
  Метод выполняет расшифровку ciphertext методом Скиталы. Создаёт таблицу размерами rows x columns, заполняет ее символами текста по столбцам, формирует открытый текст считыванием символов из таблицы по строкам и убирает символы-заполнители 'X', оставшиеся в конце текста.

**Результат программы:**

**Задание 1:**

****

Рисунок 1. Получение сообщения для зашифровки.

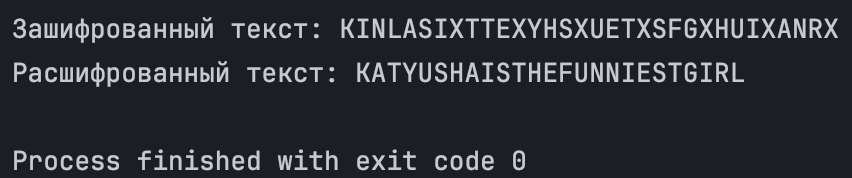
****

Рисунок 2. Зашифровка и дешифровка сообщения шифром Скитала.

**Задание 2:**

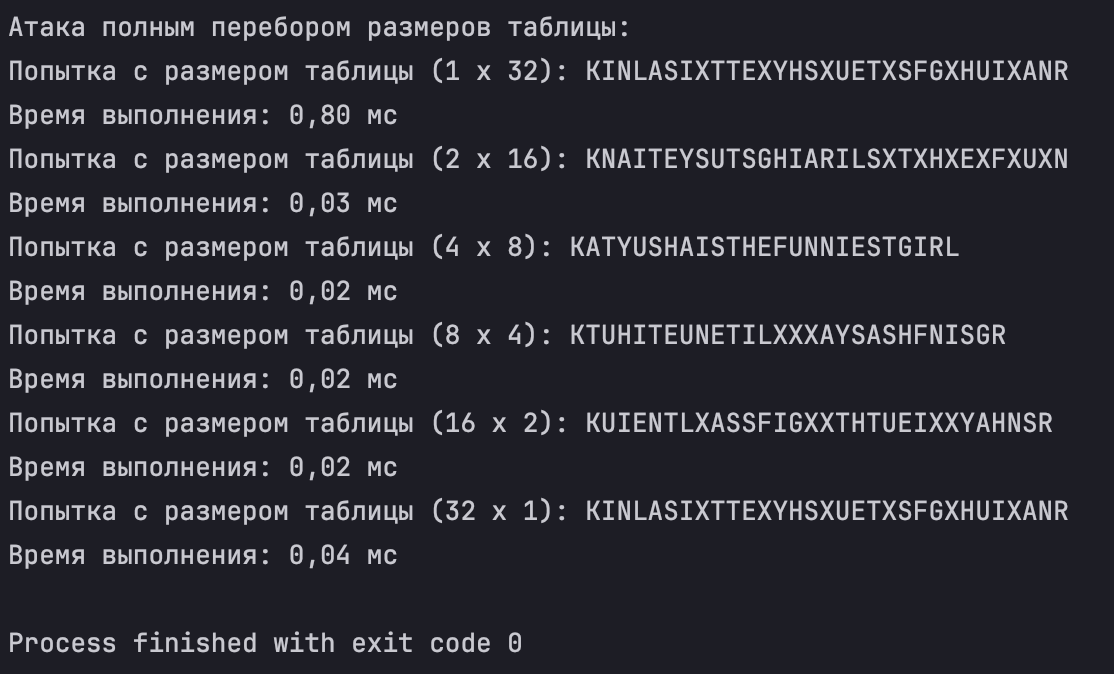
****

Рисунок 3. Атака полным перебором размеров таблицы.

**Задание 3:**

Шифр Скитала обладает довольно низкой криптографической стойкостью, поскольку он имеет довольно много слабых сторон:

* Одной из ключевых слабостей шифра Скитала является возможность атаки полным перебором. Для шифра, реализованного в моей лабораторной работе, число возможных ключей (варианты количества строк и столбцов таблицы) ограничено делителями длины зашифрованного текста. Это означает, что при длине текста в 32 символа существует всего 6 возможных комбинаций размерности таблицы (например, 1x32, 2x16, 4x8 и т.д.). Современные вычислительные средства позволяют быстро перебрать все возможные варианты, что делает шифр крайне уязвимым к взлому.
* Шифр Скитала переносит символы открытого текста в зашифрованный текст без изменения их частоты появления. Это значит, что если частотная характеристика языка (например, английского) известна атакующему, он может провести частотный анализ и быстро восстановить исходный текст, даже без полной атаки перебором.
* Сам метод шифрования Скитала подразумевает расположение символов по строкам и простое считывание по столбцам. Эта упрощённая структура создает предсказуемость, которая значительно снижает стойкость алгоритма.  
  Как и в случае с другими классическими шифрами, шифр Скитала давно изучен и разобран, что делает его ещё более уязвимым. Любые атакующие с базовыми знаниями в криптографии могут эффективно и быстро взломать такой шифр, не прибегая к сложным методам анализа.

Как итог, Шифр Скитала не рекомендуется использовать для защиты конфиденциальной информации. Причины этого заключаются в лёгкости перебора всех возможных ключей, простоте частотного анализа текста и предсказуемости самого алгоритма. Это делает шифр практически бесполезным в условиях современных вычислительных возможностей. Однако, шифр Скитала может быть полезен для образовательных целей, изучения основ перестановочных шифров и понимания базовых принципов работы с криптографическими алгоритмами.

**Задание 4:**

### ****Скитала с перемешиванием строк****

* 1. Выберите ключ, который задает порядок перестановки строк.
  2. После заполнения таблицы Скиталы представьте строки в порядке, заданным ключем. Например, если ключ – 2, 3, 1, то первая строка станет третьей, вторая строка станет первой, а третья – второй.
  3. Считайте шифртекст постолбцово из переставленной таблицы.

### ****Скитала с дополнительным шифром каждый N-й символ****

### Выполните стандартное шифрование Скитала.

### После получения промежуточного шифртекста выберите период времени N, например каждые 3 символа.

### Примените дополнительный шифр к каждому N-му символу текста. Например, к каждому третьему символу применить сдвиг с помощью шифра Цезаря или заменить символы, используя заранее заданный словарь.

### Комбинируйте шифртекст, добавляя зашифрованные N-е символы в него.

### ****Скитала с динамическим порядком заполнения таблицы****

### Вместо стандартного заполнения таблицы строками заполняйте таблицу по определённым паттернам, например зигзаг, диагональ.

* 1. Для каждого шифрования выбирайте случайный паттерн, который передаётся в качестве части ключа.
  2. Считывайте таблицу в зашифрованном порядке, основываясь на выбранном паттерне.

### ****Скитала с дополнительным зеркальным преобразованием таблицы****

### Выполните стандартное шифрование Скитала.

* 1. Перед считыванием таблицы выполните преобразование, например переверните ее на 180/90 градусов.
  2. После преобразования считайте шифртекст по обычному принципу.

### ****Скитала с вложенными таблицами****

* 1. Разделите зашифрованный текст на несколько частей
  2. Для каждой части выполните шифрование Скитала с разными размерами таблиц, например для первой части будет таблица 4х5, для второй 3х6.
  3. Объедините шифртексты всех частей, сохранив последовательность.

**Вывод:**

В результате данной лабораторной работы была разработана специальная программа на языке Java, которая способна шифровать сообщения английского языка с помощью шифра Скитала. Также данная программа способна проводить атаку на зашифрованные сообщения методом подбора возможных размеров таблицы. По результатам работы данной программы можно сделать вывод, что шифр Скитала обладает низкой криптографической стойкостью в связи с простотой его алгоритма. Поэтому данный шифр не рекомендуется использовать для шифрования важных сообщений, поскольку они с высокой вероятностью могут быть дешифрованы злоумышленником с использованием современных вычислительных средств. Данный шифр можно применять в учебных целях для ознакомления людей с основами перестановочного шифрования и криптографии в целом.

Также на криптостойкость данного шифра сильно влияют такие параметры, как мощность ключевого поля (число возможных размеров таблицы) и время, необходимое для подбора ключа. На примере моей программы можно убедиться, что шифр Скитала расшифровывается менее чем за несколько миллисекунд, что также указывает на его низкую криптографическую стойкость.

Математическая модель шифра Скитала:

Шифрование:

1. Разделить зашифровываемый текст на блоки размером **rows x columns**.
2. Заполнить таблицу посимвольно по строкам.
3. Прочитать символы из таблицы постолбцово, формируя шифртекст.

Дешифрование:

1. Определить размер таблицы (**rows x columns**) исходя из длины текста и возможных делителей.
2. Заполнить таблицу посимвольно постолбцово.
3. Прочитать символы из таблицы построчно, восстанавливая исходный текст.

Данную математическую модель можно легко реализовать и применить для шифрования и дешифрования текстов различной длины. Однако из-за простоты перестановок текст можно восстанавливать методом перебора всех возможных размеров таблицы.

Для повышения криптографической стойкости шифра Скитала рекомендуется вносить модификации в алгоритм. Например использование ключевого слова для упорядочивания столбцов, ведение нелинейного порядка заполнения таблицы (например, по диагонали или зигзагом), применение дополнительных шифров (например, шифра Цезаря) к символам перед шифрованием таблицы.

Для более высокой криптостойкости не рекомендуется использовать шифры моноалфавитной замены. Лучше использовать шифры полиалфавитной замены, поскольку они более совершенны.

Одним из таких шифров является **шифр Вижинера**. Он является простой формой многоалфавитной (полиалфавитной) замены. Многоалфавитная замена определяется ключом, содержащим не менее двух вариантов замен. В шифре Виженера в качестве ключа используется секретное слово. В полиалфавитном шифре несколько одноалфавитных шифров применяются циклически.

Также одним из сильных устройств шифрования сообщений являлось семейство переносных электромеханических роторных машин – **«Энигма»**. Данные машины имели различные параметры:

– расположение роторов: выбор роторов и их расположение;

– первоначальные позиции роторов: выбранные оператором, различные для каждого сообщения;

– настройка колец: позиция алфавитного кольца, совпадающая с роторной схемой;

– настройки штепселей: соединения штепселей на коммутационной панели.

Однако данные машины несмотря на свою высокую криптостойкость все равно были дешифрованы несмотря на длительный промежуток времени.

Данные примеры могут говорить о том, что ни один шифр не является на 100% криптостойким. А ни одна защита не является на 100% безопасной. Поэтому для защиты своих данных мы должны применять самые новые методы защиты и шифрования, а также на основе уже найденных уязвимостей дорабатывать существующие алгоритмы и придумывать новые, которые будут более совершенными.