Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Информационная безопасность**

Студент: Козека Е. М.

ФИТ 3 курс 4 группа

Преподаватель: Нистюк О. А.

Минск 2025

**Лабораторная работа №6. Изучение устройства и функциональных особенностей шифровальной машины «Энигма»**

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации подстановочно-перестановочных шифров.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости подстановочно-перестановочных шифров.
2. Изучить структуру, принципы функционирования, реализацию процедур зашифрования сообщений в машинах семейства «Энигма».
3. Приобрести практические навыки выполнения криптопреобразований информации на платформе «Энигма», реализованной в виде симуляторов.
4. Получить практические навыки оценки криптостойкости подстановочных и перестановочных шифров на платформе «Энигма».
5. Результаты выполнения лабораторной работы оформить в виде отчета о проведенных исследованиях, методике выполнения практической части задания и оценке криптостойкости шифров.

**Теоретические сведения**

Первая шифровальная машина, «Enigma A», появилась в 1923 году и весила около 50 кг. Вскоре была представлена «Enigma B», похожая на первую модель, но обе оказались слишком громоздкими для военного использования. Во время Второй мировой войны было произведено более 100 тысяч таких машин.

«Энигма» – это электромеханическое устройство, состоящее из механической и электрической частей. Механическая часть включает клавиатуру, роторы и ступенчатый механизм, который поворачивает роторы при каждом нажатии клавиши. Электрическая часть включает клавиатуру, коммутационную панель, лампочки и роторы.

Основные блоки «Энигмы»:

* Панель механических клавиш, дающая сигнал для поворота роторов.
* Три или более роторных дисков, каждый с 26 контактами, коммутируемыми в случайном порядке.
* Рефлектор, соединенный с крайним левым ротором.
* Коммутационная панель для изменения электрических соединений двух букв.
* Панель с электрическими лампочками, показывающими выходную букву.

При каждом нажатии клавиши самый правый ротор сдвигается на одну позицию, а при определенных условиях сдвигаются и другие роторы. Это приводит к различным криптографическим преобразованиям, заменяя один символ другим. Механические части замыкают контакты, создавая электрический контур. При нажатии клавиши контур замыкается, ток проходит через цепь и включает лампочку, показывая букву шифртекста. Рефлектор замыкает цепь, возвращая ток по другому пути и зажигая лампочку.

*EK = f (m, В, Re, L, M, R)*.

Чтобы оценить криптостойкость шифра, нужно учитывать все возможные настройки машины. Для этого необходимо рассмотреть следующие свойства «Энигмы»:

* выбор и порядок роторов;
* разводку (коммутацию) роторов;
* настройку колец на каждом из роторов;
* начальное положение роторов в начале сообщения;
* отражатель;
* настройки коммутационной панели.

**Ход работы**

**Задание 1.** Разработать приложение-симулятор шифровальной машины, состоящей из клавиатуры, трех роторов и отражателя.

Метод rotorEncrypt принимает на вход шифруемый символ алфавита, алфавит из которого будет осуществляться преобразования, алфавит в который будет осуществляться преобразование (алфавит ротора) и текущее смещение ротора.

rotorEncrypt(letter, originalAlphabet, encryptionAlphabet,

currentOffset) {

const originalIndex = originalAlphabet.indexOf(letter);

return encryptionAlphabet[(originalIndex + currentOffset) %

this.ROTOR\_LENGTH];

}

Листинг 6.1 – Функция rotorEncrypt

Далее были разработаны методы directPath и reversePath, которые представляют собой прямой и обратный пути преобразования соответственно.

directPath(letter, operation) {

let afterRight;

let afterMiddle;

switch (operation) {

case Operation.ENCRYPT:

afterRight = this.rotorEncrypt(letter, this.ALPHABET,

this.R\_ROTOR, this.rCurrentPosition);

afterMiddle = this.rotorEncrypt(afterRight,

this.ALPHABET, this.M\_ROTOR, this.mCurrentPosition);

return this.rotorEncrypt(afterMiddle, this.ALPHABET,

this.L\_ROTOR, this.lCurrentPosition);

case Operation.DECRYPT:

afterRight = this.rotorDecrypt(letter, this.ALPHABET,

this.R\_ROTOR, this.rCurrentPosition);

afterMiddle = this.rotorDecrypt(afterRight,

this.ALPHABET, this.M\_ROTOR, this.mCurrentPosition);

return this.rotorDecrypt(afterMiddle, this.ALPHABET,

this.L\_ROTOR, this.lCurrentPosition);

}

}

reversePath(letter, operation) {

let afterLeft;

let afterMiddle;

switch (operation) {

case Operation.ENCRYPT:

afterLeft = this.rotorEncrypt(letter, this.L\_ROTOR,

this.ALPHABET, this.lCurrentPosition);

afterMiddle = this.rotorEncrypt(afterLeft, this.M\_ROTOR,

this.ALPHABET, this.mCurrentPosition);

return this.rotorEncrypt(afterMiddle, this.R\_ROTOR,

this.ALPHABET, this.rCurrentPosition);

case Operation.DECRYPT:

afterLeft = this.rotorDecrypt(letter, this.L\_ROTOR,

this.ALPHABET, this.lCurrentPosition);

afterMiddle = this.rotorDecrypt(afterLeft, this.M\_ROTOR,

this.ALPHABET, this.mCurrentPosition);

return this.rotorDecrypt(afterMiddle, this.R\_ROTOR,

this.ALPHABET, this.rCurrentPosition);

}

}

Листинг 6.2 – Функции directPath и reversePath

Метод passThroughReflector принимает текущий символ и осуществляет подстановку на основе C Dunn рефлектора.

this.REFLECTOR = {

a: 'r', b: 'd', c: 'o', d: 'b', e: 'j', f: 'n',

g: 't', h: 'k', i: 'v', j: 'e', k: 'h', l: 'm',

m: 'l', n: 'f', o: 'c', p: 'w', q: 'z', r: 'a',

s: 'x', t: 'g', u: 'y', v: 'i', w: 'p', x: 's',

y: 'u', z: 'q'

};

passThroughReflector(letter) {

    return this.REFLECTOR[letter];

}

Листинг 6.3 – Функция passThroughReflector

Метод shiftRotors выполняется после зашифрования/расшифрования одного символа и меняет положения каждого из роторов.

shiftRotors() {

this.lCurrentPosition = (this.lCurrentPosition + this.L\_SHIFT) %

this.ROTOR\_LENGTH;

this.mCurrentPosition = (this.mCurrentPosition + this.M\_SHIFT) %

this.ROTOR\_LENGTH;

this.rCurrentPosition = (this.rCurrentPosition + this.R\_SHIFT) %

this.ROTOR\_LENGTH;

}

Листинг 6.4 – Функция shiftRotors

Итоговый метод шифрования encrypt принимает на вход оригинальный текст и выполняет шифрование с помощью метода прямого пути, метода рефлектора и метода обратного пути. После каждой итерации осуществляется смещение роторов.

encrypt(text) {

let result = "";

for (let s of text) {

if (this.ALPHABET.includes(s)) {

let afterDirect = this.directPath(s, Operation.ENCRYPT);

let afterReflector =

this.passThroughReflector(afterDirect);

result += this.reversePath(afterReflector,

Operation.ENCRYPT);

this.shiftRotors();

}

}

return result;

}

Листинг 6.5 – Функция encrypt

Метод расшифрования decrypt принимает на вход зашифрованный текст. Он выполняет аналогичные методу encrypt операции, только на прямом и обратном пути осуществляет расшифрование.

decrypt(encryptedText) {

    let result = "";

    for (let s of encryptedText) {

        if (this.ALPHABET.includes(s)) {

            let afterDirect = this.directPath(s, Operation.DECRYPT);

            let afterReflector =

this.passThroughReflector(afterDirect);

            result += this.reversePath(afterReflector,

Operation.DECRYPT);

            this.shiftRotors();

        }

    }

    return result;

}

Листинг 6.6 – Функция decrypt

Кроме того, класс Enigma позволяет устанавливать начальные положения роторов в своём конструкторе.

Примеры зашифрования 1 символа сообщения «kozekaelizavetamaksimovna», когда L – V, M – VI, R – VII, Re – C Dunn, Li-Mi-Ri = 1-2-2:

1. Шифруемый символ: k, исходный алфавит: abcdefghijklmnopqrstuvwxyz;
2. Ротор R в положении 2: jhgrcxmyswboufaivlpekqdtnz. Положение символа k в исходном алфавите: 10. Производим замену k 🡪 b;
3. Ротор M в положении 2: gvoumfyqbenhzrdkasxlictwjp. Положение символа b в исходном алфавите: 1. Производим замену b 🡪 v;
4. Ротор L в положении 1: zbrgityupsdnhlxawmjqofeckv. Положение символа v в исходном алфавите: 21. Производим замену v 🡪 f;
5. Далее символ f проходит через рефлектор. Производим замену: f 🡪 n;
6. Обратный ход через ротор L. Положение символа n в алфавите ротора L: 12. Производим замену n 🡪 n;
7. Обратный ход через ротор M. Положение символа n в алфавите ротора M: 12. Производим замену n 🡪 o;
8. Обратный ход через ротор R. Положение символа o в алфавите ротора R: 13. Производим последнюю замену o 🡪 p.
9. Далее роторы смещаются и начинается шифрование следующего символа.

Результат работы приложения для роторов в позиции 1-2-2 можно увидеть на рисунке 6.1.

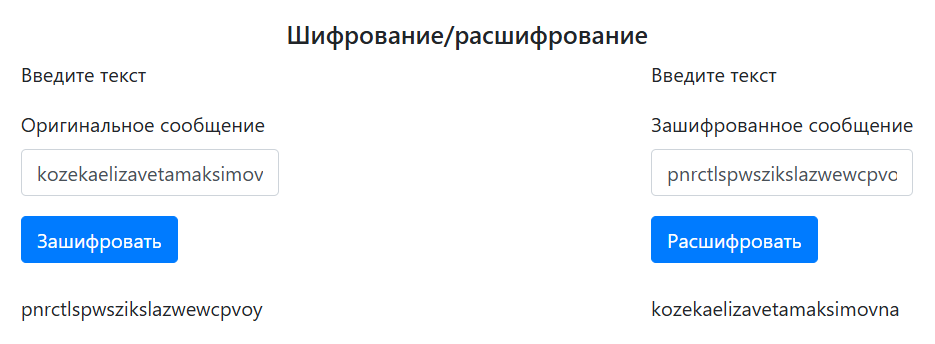


Рисунок 6.1 – Результат шифрования и дешифрования сообщения для роторов в позиции 1-2-2

Результат работы приложения для роторов в позиции 0-0-0 можно увидеть на рисунке 6.2.

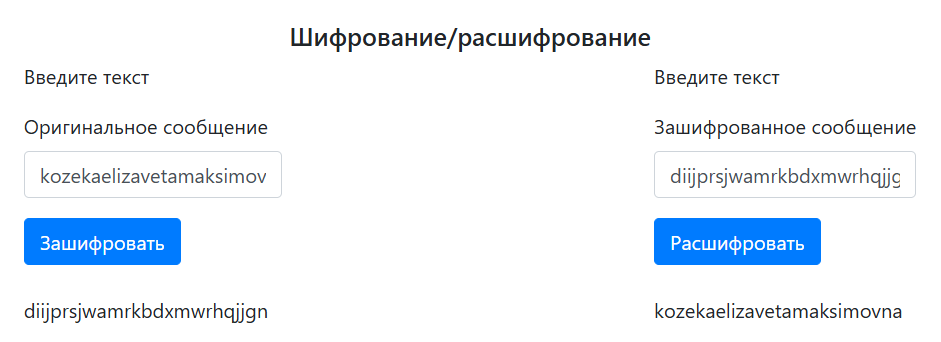


Рисунок 6.2 – Результат шифрования и дешифрования сообщения для роторов в позиции 0-0-0

Результат работы приложения для роторов в позиции 2-1-3 можно увидеть на рисунке 6.3.

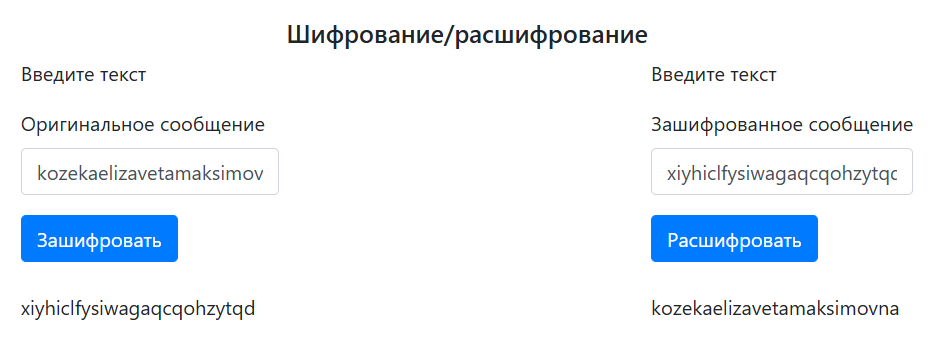


Рисунок 6.3 – Результат шифрования и дешифрования сообщения для роторов в позиции 2-1-3

Результат работы приложения для роторов в позиции 4-0-3 можно увидеть на рисунке 6.4.

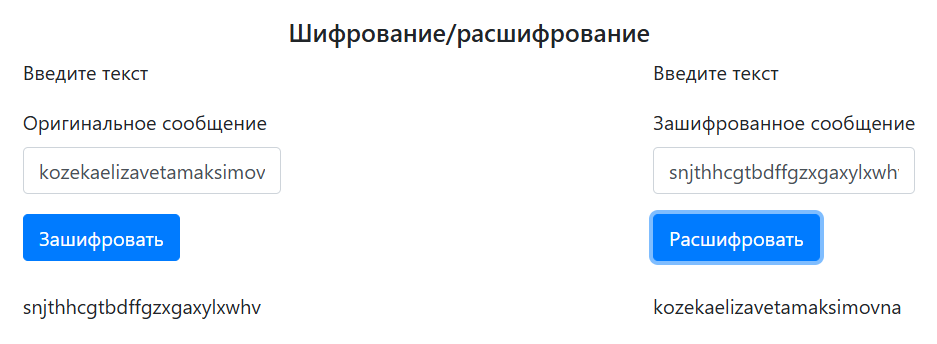


Рисунок 6.4 – Результат шифрования и дешифрования сообщения для роторов в позиции 4-0-3

Результат работы приложения для роторов в позиции 2-3-4 можно увидеть на рисунке 6.5.

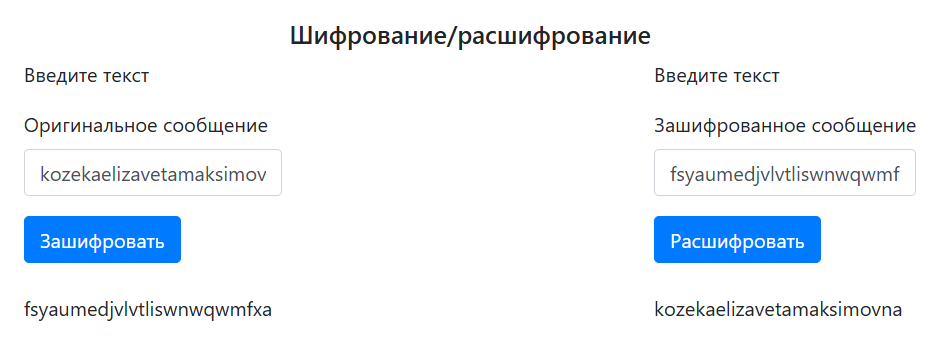


Рисунок 6.5 – Результат шифрования и дешифрования сообщения для роторов в позиции 2-3-4

**Задание 2.** Сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений. Оценить время выполнения операций зашифрования/расшифрования.

Гистограмма частот появления символов исходного сообщения приведена на рисунке 6.6.

Рисунок 6.6 – Гистограмма частот появления символов исходного сообщения

***Роторы в позиции 1-2-2:***

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 6.7.

Рисунок 6.7 – Гистограмма частот появления символов зашифрованного сообщения

***Роторы в позиции 0-0-0:***

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 6.8.

Рисунок 6.8 – Гистограмма частот появления символов зашифрованного сообщения

***Роторы в позиции 2-1-3:***

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 6.9.

Рисунок 6.9 – Гистограмма частот появления символов зашифрованного сообщения

***Роторы в позиции 4-0-3:***

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 6.10.

Рисунок 6.10 – Гистограмма частот появления символов зашифрованного сообщения

***Роторы в позиции 2-3-4:***

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 6.11.

Рисунок 6.11 – Гистограмма частот появления символов зашифрованного сообщения

Можно заметить, что частоты появления не совпадают, что значительно увеличивает криптостойкость шифра машины «Энигма».

Количество возможных комбинаций роторов: 3 \* 2 \* 1 = 6.

Возможные положения роторов: 26 \* 26 \* 26 = 17576.

Начальные положения роторов: 26 \* 26 = 676.

6 \* 17576 \* 676 = 71288256.

**Вывод:** в ходе выполнения лабораторной работы были изучены принципы функционирования машин семейства «Энигма». Также было разработано приложение для реализации зашифрования и расшифрования с помощью симулятора машины «Энигма». Было выполнено исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях.