МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №4**

**ИЗУЧЕНИЕ УСТРОЙСТВА И ФУНКЦИОНАЛЬНЫХ ОСОБЕННОСТЕЙ ШИФРОВАЛЬНОЙ МАШИНЫ «ЭНИГМА»**

Выполнил:

Cтудент 3 курса 1 группы

Парибок И. А.

Вариант 5

Минск 2023

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации перестановочных шифров (работа рассчитана на 4 часа аудиторных занятий).

**Теоретическая часть:**

Машина «Энигма» – это электромеханическое устройство. Как и другие роторные машины, «Энигма» состоит из комбинации механических и электрических подсистем.

Механическая часть включает в себя клавиатуру, набор вращающихся дисков – роторов, которые расположены вдоль вала и прилегают к нему, и ступенчатого механизма, двигающего один или несколько роторов при каждом нажатии на клавишу. Электрическая часть, в свою очередь, состоит из электрической схемы, соединяющей между собой клавиатуру, коммутационную панель, лампочки и роторы (для соединения роторов использовались скользящие контакты).

На рис. 1 показана фотография одной из моделей «Энигмы» с указанием месторасположения основных модулей машины. Как видно на этом рисунке, «Энигма» состоит из 5 основных блоков:

• панели механических клавиш 1 (дают сигнал поворота роторных дисков);

• трех (или более) роторных дисков 2, каждый имеет контакты по сторонам, по 26 на каждую, которые коммутируют в случайном порядке; по окружности нанесены буквы латинского алфавита либо числа;

• рефлектора 3 (имеет контакты с крайним слева ротором);

• коммутационной панели 4 (служит для того, чтобы дополнительно менять местами электрические соединения (контакты) двух букв);

• панели в виде электрических лампочек 5; индикационная панель с лампочками служит индикатором выходной буквы в процессе шифрования.

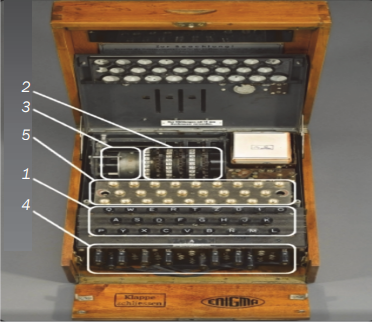


Рисунок 1 – Одна из моделей (трехроторная) «Энигмы»

Конкретный механизм мог быть разным, но общий принцип был таков: при каждом нажатии на клавишу самый правый ротор сдвигался на одну позицию, а при определенных условиях сдвигались и другие роторы. Движение роторов приводило к различным криптографическим преобразованиям при каждом следующем нажатии на клавишу на клавиатуре, т. е. зашифрование/расшифрование сообщений основано на выполнении ряда замен (подстановок) одного символа другим. Идея А. Шербиуса состояла в том, чтобы добиться этих подстановок электрическими связями.

Механические части двигались и, замыкая контакты, образовывали меняющийся электрический контур. При нажатии на клавишу клавиатуры контур замыкается, ток проходит через созданную (для зашифрования/расшифрования одного конкретного символа сообщения) цепь и в результате включает одну из набора лампочек, отображающую искомую букву шифртекста (или расшифрованного сообщения). На рис. 2 показаны упрощенная конструкция ротора (*а*) и рефлектора (*б*). Замыкание цепи происходило за счет рефлектора. На рис. 3 схематично показано, как некоторая буква (например, «а») будет зашифрована другой буквой (например, «g»), а следующая за ней буква сообщения (также «а») – уже буквой «с»

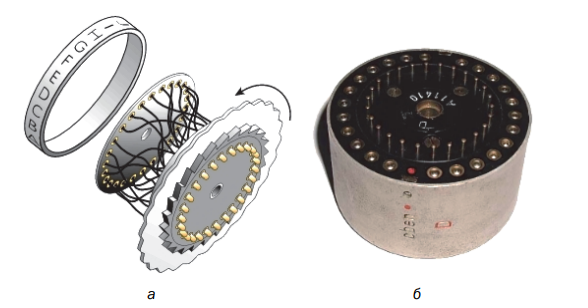


Рисунок 2 – Упрощенная конструкция ротора (*а*) и рефлектора (*б*) «Энигмы»

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 3 – Пояснение к принципу шифрования путем формирования электрической цепи

Отметим, что на рис. 3 электрическая цепь не представлена в виде замкнутой, поскольку не показаны части коммутационной панели и электрическая лампочка. Замкнутые электрические цепи хорошо иллюстрирует рис. 4.

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

Рисунок 4 – Пояснение к принципу формирования зашифрованного символа с помощью замкнутой электрической цепи

Замкнутую цепь составляют: батарея *1* (это могут быть и иные источники питания), нажатая двунаправленная буквенная клавиша *2*, разъем коммутационной панели *3* (как видим, в одном случае – буква «а» – коммутационного перехода на другую букву нет), входной разъем (входное колесо) роторного модуля *4*, роторный модуль *5* (состоит из трех роторов, как в версии «Энигмы» для Вермахта, *Wehrmacht Enigma М3*, или четырех – в версии «Энигмы» для военно-морского флота, *Kriegsmarine Enigma M4*), рефлектор *6*. Последний возвращает ток (цепь) по другому пути через те же узлы, «зажигая» на ламповой панели букву «D», к другому полюсу батареи. Обратим внимание, что обратная часть цепи уже проходит с учетом выполненной коммутации (*7* и *8*).

Отметим также, что клавиатура соответствовала немецкой раскладке *QWERTZ.*

Во время Второй мировой войны немецкие операторы использовали специальную (тайную) шифровальную книгу для установки роторов и настроек колец.

Операторы «Энигмы» (шифровальщики и дешифровальщики) выполняли следующие основные операции.

***Пример 1.***

Зашифрование сообщения.

1. Установить начальную стартовую позицию роторов (предположим, их 3) согласно текущей кодовой таблице (коду дня), например WZA.

2. Выбрать случайный ключ сообщения, например SXT. Затем оператор устанавливал роторы в стартовую позицию WZA.

3. Зашифровать ключ сообщения SXT. Предположим, что в результате зашифрования ключа получится UHL.

4. Далее оператор ставил ключ сообщения (SXT) как начальную позицию роторов и зашифровывал собственно сообщение. После этого он отправлял стартовую позицию (WZA) и зашифрованный ключ (UHL) вместе с сообщением.

Расшифрование сообщения.

1. Установить стартовые позиции роторов в соответствии с первой трехграммой (WZA).

2. Расшифровать вторую треграмму (UHL) и извлечь исходный ключ (SXT).

3. Далее получатель использовал этот ключ как стартовую позицию для расшифрования шифртекста. Обычно срок действия ключей составлял одни сутки.

***Пример 2.***

Зашифрование сообщения.

1. Установить стартовую позицию роторов согласно коду дня. Например, если код был «HUA», роторы должны быть инициализированы на «H», «U» и «A» соответственно.

2. Выбрать случайный код с тремя буквами, например ACF.

3. Зашифровать текст «ACFACF» (повторный код), используя начальную установку роторов шага 1. Например, предположим, что зашифрованный код – «OPNABT».

4. Установить стартовые позиции роторов к OPN (половина зашифрованного кода).

5. Присоединить зашифрованные шесть букв, полученных на шаге 2 (OPNABT), в конец к начальному сообщению.

6. Зашифровать сообщение, включая код с 6 буквами. Передать зашифрованное сообщение.

Расшифрование сообщения.

1. Получить сообщение и отделить первые шесть букв.

2. Установить стартовую позицию роторов согласно коду дня.

3. Расшифровать первые шесть букв сообщения, используя начальную установку шага 2.

4. Установить позиции роторов на первую половину расшифрованного кода.

5. Расшифровать сообщение (без первых шести букв).

Более сложная система использует случайный ряд символов для нижнего алфавита. Принцип, положенный в основу этой «случайности», имеет много общего с перестановочными шифрами. Например, ниже показан принцип подстановки, основанный на взаимной перестановке во втором (нижнем) алфавите в 13 парах символов, расположенных случайным образом:



Этот принцип случайности использовался и при изготовлении роторов и рефлекторов для «Энигмы». Всего за время Второй мировой войны немцами было изготовлено восемь роторов и четыре рефлектора, но одновременно могло использоваться ровно столько, на сколько была рассчитана машина.

Ниже на рис. 5 и 6 представлены спецификации соответственно на роторы и на рефлекторы.

Изображение выглядит как календарь

Автоматически созданное описание

Рисунок 5 – Спецификация на роторы «Энигмы»

Изображение выглядит как текст, стол

Автоматически созданное описание

Рисунок 6 – Спецификация на рефлекторы «Энигмы»

Рассмотрим пример использования приведенных спецификаций.

***Пример 3***. В этом примере мы рассмотрим процедуру зашифрования только одной буквы («G»).

Предположим, что «Энигма» оснащена роторами I, II, III (см. рис. 5). Таким образом, правым ротором (R) является III приведенной спецификации. Предположим также, что каждый ротор находится в своем положении A, когда выполняется шифрование. Если взять информацию из рис. 5 и 6, указав фактическую разводку ротора, это означает, что правый ротор R производит подстановку в соответствии с переставленными буквами исходного алфавита, т. е. буква «G» будет заменена буквой «C»:



Центральный ротор (М) или II заменяет букву «С» на букву «D»:



Левый ротор (L) или III – соответственно букву «D» на букву «F»:



Предположим далее, что используется рефлектор В (спецификация на рис. 6 – первая строка):



Обратим внимание на то, что рефлектор имеет только 13 соединений, т. е. имеется 13 пар подстановок: A – Y, В – R и т. д. В нашем примере произошла подстановка F → S.

Ток теперь проходит обратный путь через три ротора в последовательности L → M → R.

Эффект преобразования левого ротора (обратный):



соответственно – среднего ротора (обратный):



и, наконец, правого ротора (обратный):



После всех (в данном случае – 7) подстановок буква «G» будетзашифрована буквой «P».

Процедура расшифрования шифртекстов предусматривала настройку отражателя, роторов и коммутационной панели машины в соответствии с таблицами (книгами) и использованными при зашифровании паролями.

**Пример 4**

Рассмотрим пример шифрования символа на конкретном примере с настройками роторов и рефлектора, данными в лабораторной работе.

1. На клавиатуре вводим букву P, она попадает в правый ротор(Gamma), на выходе из него кодируется в букву C
2. Буква C попадает на средний ротор(Beta), на выходе из него кодируется в букву Y
3. Буква Y попадает на левый ротор(I), на выходе из него кодируется в букву C
4. Буква C попадает на рефлектор где меняется на букву U
5. Буква U попадает снова на левый ротор(I), на выходе из него кодируется в букву A
6. Буква A попадает на средний ротор(Beta), на выходе из него кодируется в букву L
7. Буква L попадает на правый ротор(Gamma),  на выходе из него кодируется в букву B

**Практическое задание:**

1. Ознакомиться с функционалом хотя бы одного (по согласованию с преподавателем) симулятора «Энигмы». Произвести зашифрование сообщения (собственные имя, отчество, фамилия) при 8–10 различных настройках машины-симулятора. Оценить частотные свойства символов в шифртекстах и сравнить этот параметр с частотными свойствами символов для исходного текста.
2. Разработать приложение-симулятор шифровальной машины, состоящей из клавиатуры, трех роторов и отражателя. Типы роторов (L – M – R) и отражателя Re следует выбрать из рис. 5 и 6 в соответствии со своим вариантом, представленным в таблице 1. Крайний правый столбец этой таблицы показывает, на какое число шагов (букв, i) перемещается соответствующий ротор при зашифровании одного (текущего) символа; число 0 означает перемещение соответствующего ротора на один шаг при условии, что расположенный правее ротор совершит один оборот.

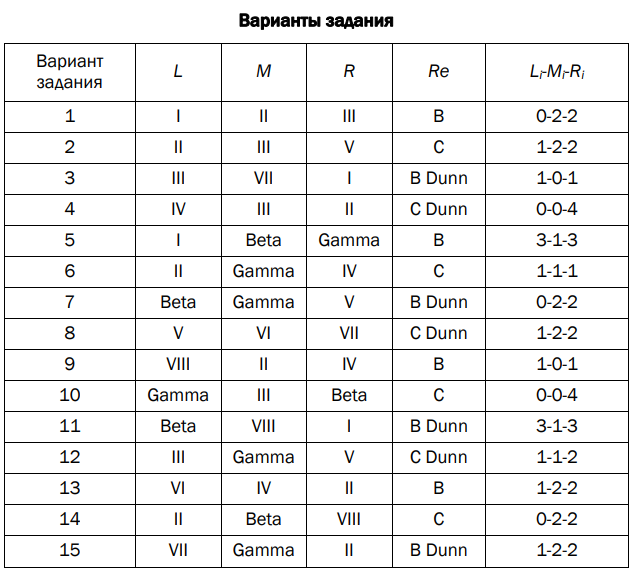


Таблица 1 – Варианты заданий

С помощью разработанного приложения зашифровать сообщение в соответствии с п. 1 практического задания, применив не менее 5 вариантов начальных установок роторов.

Оценить криптостойкость вашего варианта машины.

3. Результаты оформить в виде отчета по установленным правилам.

**Выполнение работы:**

На рисунках 7-9 представлены результаты шифрования текста с помощью симулятора машины энигмы. Во всех примерах использовались различные настройки машины-симулятора.

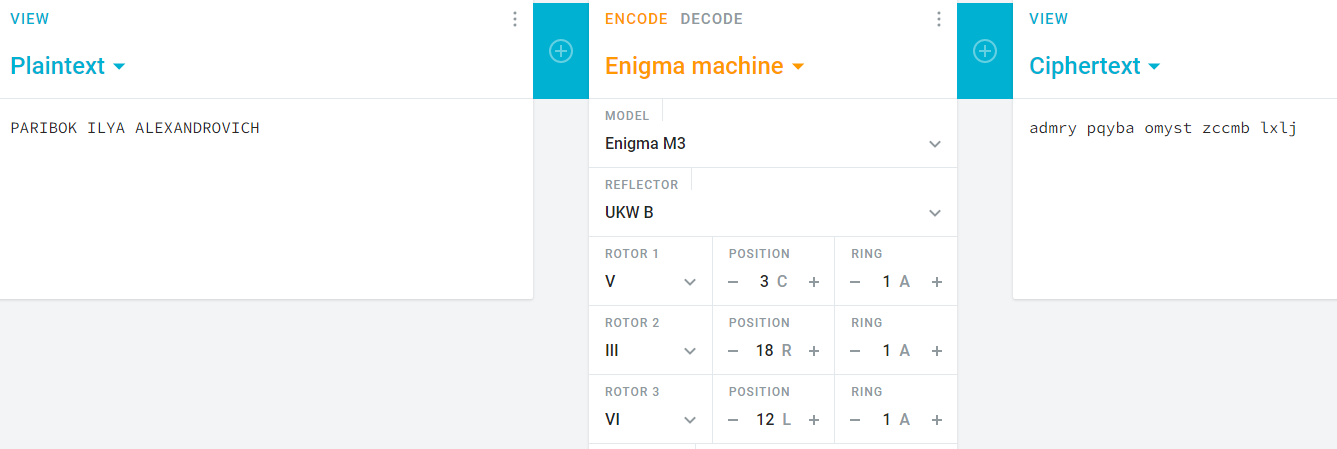


Рисунок 7 – Настройки симулятора и значение шифртекста (пример 1)

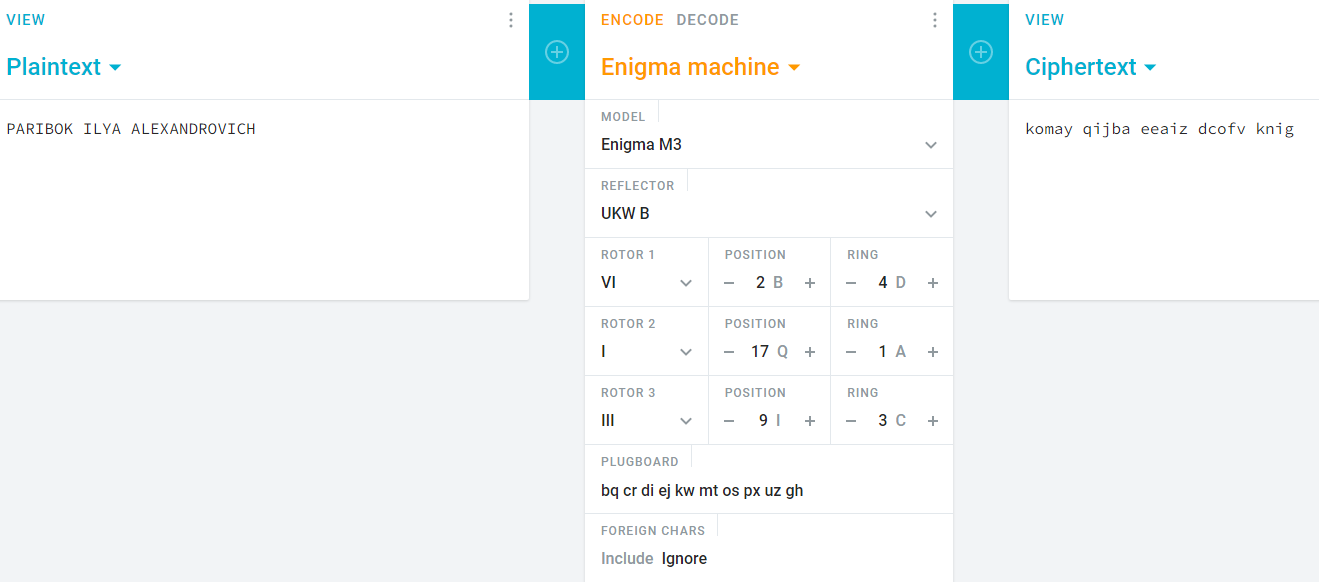


Рисунок 8 – Настройки симулятора и значение шифртекста (пример 2)

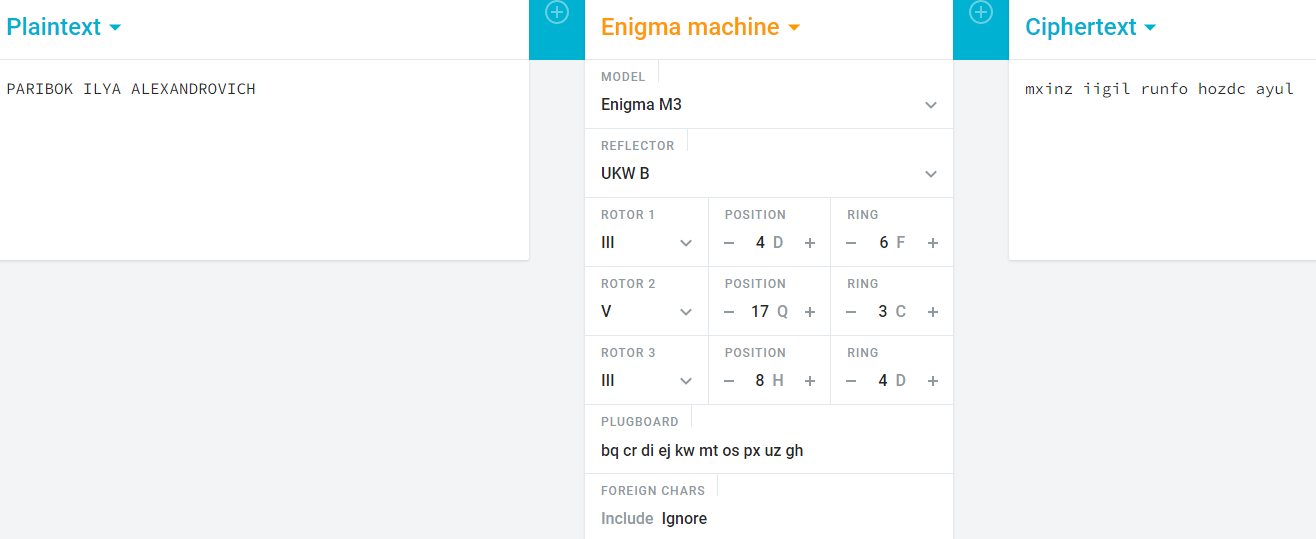


Рисунок 9 – Настройки симулятора и значение шифртекста (пример 3)

**Изображение выглядит как диаграмма

Автоматически созданное описание**

Рисунок 10 –Частотная характеристика исходного и зашифрованного сообщения

Частотные характеристики исходного и зашифрованного сообщений, полученные с помощью Enigma (рисунок10), могут дать некоторую информацию о структуре сообщений.

В целом, частотные характеристики зашифрованного сообщения будут отличаться от частотных характеристик исходного сообщения. Это связано с тем, что Enigma использует перестановки и замены символов, чтобы зашифровать сообщения, что сильно искажает исходную структуру сообщения.

На рисунках 11-13 представлены примеры выполнения приложения с различными настройками машины. Настройки машины для различных примеров: Листинг программы изложен в приложение А.

**Пример 1**:  *L–M–R*:3–1–3

**Пример 2**:  *L–M–R*:1–1–3

**Пример 3**:  *L–M–R*:3–1–1



Рисунок 11 – Пример 1 шифрования и дешифрования



Рисунок 12 – Пример 2 шифрования и дешифрования



Рисунок 13 – Пример 3 шифрования и дешифрования

**Вывод:** В результате проведенной лабораторной работы мы познакомились с устройством и функциональными особенностями шифровальной машины «Энигма», а также приобрели практические навыки оценки криптостойкости перестановочных и подстановочных шифров на платформе «Энигма».

**Контрольные вопросы**

**1. Дать пояснение к структуре шифровальных машин «Энигма».**

– при каждом нажатии на клавишу самый правый ротор сдвигался на одну позицию, а при определенных условиях сдвигались и другие роторы;

– движение роторов приводило к различным криптографическим преобразованиям при каждом следующем нажатии на клавишу на клавиатуре, т. е. зашифрование/расшифрование сообщений основано на выполнении ряда замен (подстановок) одного символа другим.

– при нажатии на клавишу, контур замыкается, включается одна из набора лампочек, показывающая искомую букву шифртекста

**2. На основе каких шифров строится машина «Энигма»?**

«Энигма» строится на основе подстановочных шифров, подобных шифру Цезаря, в котором, как известно, ключ сообщения, который должен знать получатель, – это просто смещение между двумя алфавитами. Принято считать, что в основе шифра «Энигмы» лежит динамический шифр Цезаря.

**3. Дать пояснение к принципам зашифрования сообщений.**

Более сложная система использует случайный ряд символов для нижнего алфавита. Принцип, положенный в основу этой «случайности», имеет много общего с перестановочными шифрами.

Принцип подстановки, основанный на взаимной перестановке во втором (нижнем) алфавите в 13 парах символов, расположенных случайным образом.

**4. Дать характеристику криптостойкости шифровальной машины Энигма.**

Преобразование «Энигмы» для каждой буквы может быть определено математически как результат подстановок.

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.

Немецкие криптологии полагали, что один ротор может быть подключен 4\*10^26 различными способами. Сочетание трех роторов и отражателя позволяет получить астрономические цифры возможных вариантов подстановок. Для союзников, которые знали конструкции роторов, число различных вариантов существенно уменьшалось.

Проблема криптоанализа шифров «Энигмы» была экстраординарной (с учетом электромеханических конструкций устройств для криптоанализа, применяемых в то время). Исчерпывающий поиск всех возможных 1,07 · 10^23 настроек был невозможен в 1940-х гг., а его сопоставимый 77-битный ключ огромен даже для современных электронных систем. Чтобы дать представление о размере этого числа, представим, что у нас есть 1,07 · 10^23 листов бумаги толщиной около 1 мм. Из этих листов можно сложить примерно 70 000 000 стопок бумаги, каждая из которых простирается от Земли до Солнца. Кроме того, 1,07 · 10^23 дюйма равно 288 500 световых лет.

**5. Дать характеристику криптостойкости машины-симулятора на основе разработанного приложения.**

Модель Энигмы имеет 3 различных ротора, пронумерованных римскими цифрами.

При шифровании роторы можно располагать в любой последовательности, что для трех роторов дает 6 разных комбинаций.

Помимо этого, каждый ротор может быть установлен в одной из 26 возможных стартовых позиций. Т.е. начальное положение роторов имеет всего 6\*26^3=105456 комбинаций.

Количество всех возможных соединений на коммутационной панели вычисляется по формуле n! /((n-2m)! m! 2^m), где n — количество букв алфавита, m — количество соединенных пар.

Для 26 букв английского алфавита и 10 пар это составляет 150738274937250=2^47 различных комбинаций.

Таким образом базовая версия Энигмы с тремя роторами имеет солидное даже по современным меркам пространство ключей:

150738274937250\*105456=15,896,255,521,782,636,000≈2^64.

Частоты символов в исходном тексте и зашифрованном сообщении сильно отличаются: некоторые символы вообще не появляются в шифре, а некоторые хоть и встречаются с такой же частотой, на находятся в совершенно других местах. Таким образом, криптостойкость разработанного симулятора такая же, как и у оригинальной машины.

**6. Пояснить основные принципы расшифрования сообщений «Энигмы».**

Процедура расшифрования шифртекстов предусматривала настройку отражателя, роторов и коммутационной панели машины в соответствии с таблицами (книгами) и использованными при зашифровании паролями.

**Приложение А**

using System;

namespace LAB\_06

{

class Program

{

static void Main()

{

Enigma enigma = new Enigma();

string encoded = enigma.Crypt("PARIBOK", 3, 1,1);

Console.WriteLine(encoded);

Console.WriteLine(enigma.Decrypt(encoded, 3, 1, 1));

}

}

} }

namespace LAB\_06

{

public class Rotor

{

private readonly char[] \_rotorChar;

private int \_currentIndex;

public Rotor(string rotorString, int startIndex)

{

\_rotorChar = rotorString.ToCharArray();

\_currentIndex = startIndex >= rotorString.Length ? 0 : startIndex;

}

public char this[int index]

{

get

{

return \_rotorChar[(index + \_currentIndex) % \_rotorChar.Length];

}

}

public int IndexOf(char symbol)

{

int index = \_rotorChar.ToList().IndexOf(symbol);

// shows how much the shift to the right has occurred

int rightOffset = \_rotorChar.Length - \_currentIndex;

int offsetRotorIndex = (index + rightOffset) % \_rotorChar.Length;

return offsetRotorIndex;

}

// Return true if the Rotor has made a full revolution

public bool MoveRotor(int offset)

{

\_currentIndex = \_currentIndex + offset;

if (\_currentIndex >= \_rotorChar.Length)

{

\_currentIndex = \_currentIndex % \_rotorChar.Length;

return true;

}

return false;

}

public char CurrentRotor()

{

return \_rotorChar[\_currentIndex];

}

public void Reset()

{

\_currentIndex = 0;

}

}

namespace LAB\_06

{

public class Enigma

{

private static readonly string \_alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

private static readonly string \_rotor1 = "EKMFLGDQVZNTOWYHXUSPAIBRCJ";

private static readonly string \_rotorBeta = "LEYJVCNIXWPBQMDRTAKZGFUHOS";

private static readonly string \_rotorGamma = "FSOKANUERHMBTIYCWLQPZXVGJD";

private static readonly string[] \_reflectorB = { "AY", "BR", "CU", "DH", "EQ", "FS", "GL", "IP", "JX", "KN", "MO", "TZ", "VW" };

public string Crypt(string text, int posL, int posM, int posR)

{

var rotorR = new Rotor(\_rotorGamma, posR);

var rotorM = new Rotor(\_rotorBeta, posM);

var rotorL = new Rotor(\_rotor1, posL);

StringBuilder result = new StringBuilder(text.Length);

foreach (var ch in text)

{

Console.Write(ch);

char symbol = rotorR[\_alphabet.IndexOf(ch)];

LogToConsole(symbol);

symbol = rotorM[\_alphabet.IndexOf(symbol)];

LogToConsole(symbol);

symbol = rotorL[\_alphabet.IndexOf(symbol)];

LogToConsole(symbol);

symbol = \_reflectorB.First(x => x.Contains(symbol)).First(x => !x.Equals(symbol));

LogToConsole(symbol);

symbol = rotorL[\_alphabet.IndexOf(symbol)];

LogToConsole(symbol);

symbol = rotorM[\_alphabet.IndexOf(symbol)];

LogToConsole(symbol);

symbol = rotorR[\_alphabet.IndexOf(symbol)];

LogToConsole(symbol);

Console.WriteLine();

result.Append(symbol);

rotorR.MoveRotor(posR);

rotorM.MoveRotor(posM);

rotorL.MoveRotor(posL);

}

return result.ToString();

}

public string Decrypt(string text, int posL, int posM, int posR)

{

var rotorR = new Rotor(\_rotorGamma, posR);

var rotorM = new Rotor(\_rotorBeta, posM);

var rotorL = new Rotor(\_rotor1, posL);

StringBuilder result = new StringBuilder(text.Length);

foreach (var ch in text)

{

Console.Write(ch);

char symbol = \_alphabet[rotorR.IndexOf(ch)];

LogToConsole(symbol);

symbol = \_alphabet[rotorM.IndexOf(symbol)];

LogToConsole(symbol);

symbol = \_alphabet[rotorL.IndexOf(symbol)];

LogToConsole(symbol);

symbol = \_reflectorB.First(x => x.Contains(symbol)).First(x => !x.Equals(symbol));

LogToConsole(symbol);

symbol = \_alphabet[rotorL.IndexOf(symbol)];

LogToConsole(symbol);

symbol = \_alphabet[rotorM.IndexOf(symbol)];

LogToConsole(symbol);

symbol = \_alphabet[rotorR.IndexOf(symbol)];

LogToConsole(symbol);

Console.WriteLine();

result.Append(symbol);

rotorR.MoveRotor(posR);

rotorM.MoveRotor(posM);

rotorL.MoveRotor(posL);

}

return result.ToString();

}

public void LogToConsole(char symbol)

{

Console.Write(" > " + symbol);

}

}

} }