МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования

«БЕЛОРУССКИЙ

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Направление специальности Информационные системы и технологии \_\_

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**ПО ДИСЦИПЛИНЕ**

**«ЗАЩИТА ИНФОРМАЦИИ**

**И НАДЕЖНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент 3 курса 1 группы

Кашперко Василиса Сергеевна

Проверил: ассистент

Нистюк Ольга Александровна

Минск 2023

**Лабораторная работа № 2**

**Исследование криптографических шифров на основе подстановки символов**

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации подстановочных шифров.

**Теоретическая часть:**

Сущность подстановочного шифрования состоит в том, что исходный текст (из множества М) и зашифрованный текст (из множества С) основаны на использовании одного и того же или разных алфавитов, а тайной или ключевой информацией является алгоритм подстановки.

В моноалфавитных шифрах операция замена производится только над каждым одиночным символом сообщения Мi. Для наглядной демонстрации шифра простой замены достаточно выписать под заданным алфавитом тот же алфавит, но в другом порядке или, например, со смещением. Записанный таким образом алфавит называют алфавитом замены.

Максимальное количество ключей для любого шифра этого вида не превышает N!, где N – количество символов в алфавите.

В шифре **Цезаря** шифрование:

y = x + k (mod N),

а расшифрование: х = у – k (mod N).

Диск **Альберти.** Он состоял из двух дисков – внешнего неподвижного и внутреннего подвижного дисков, на которые были нанесены буквы алфавита. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замене ее на букву с внутреннего диска, стоящую под ней.

Таблица **Трисемуса**. При зашифровании первая буква открытого текста заменяется на букву, стоящую в первой строке, вторая – на букву, стоящую во второй строке, и т.д. После использования последней строки вновь возвращаются к первой.

Шифр **Виженера**. Основная идея заключается в следующем. Создается таблица размером N·N, в первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущего циклического сдвига последней на 1 символ влево.

**Практическая часть:**

В данной лабораторной работе необходимо разработать авторское приложение для шифрования и расшифрования с использованием английского алфавита:

1. Шифр Цезаря с ключевым словом, ключевое слово – собственная фамилия

2. Таблица Трисемуса, ключевое слово – собственное имя

**Вывод программы:**

В ходе разработки программы был использован Windows Presentation Foundation — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework, использующая язык XAML. На рисунке 1 изображен интерфейс программы.

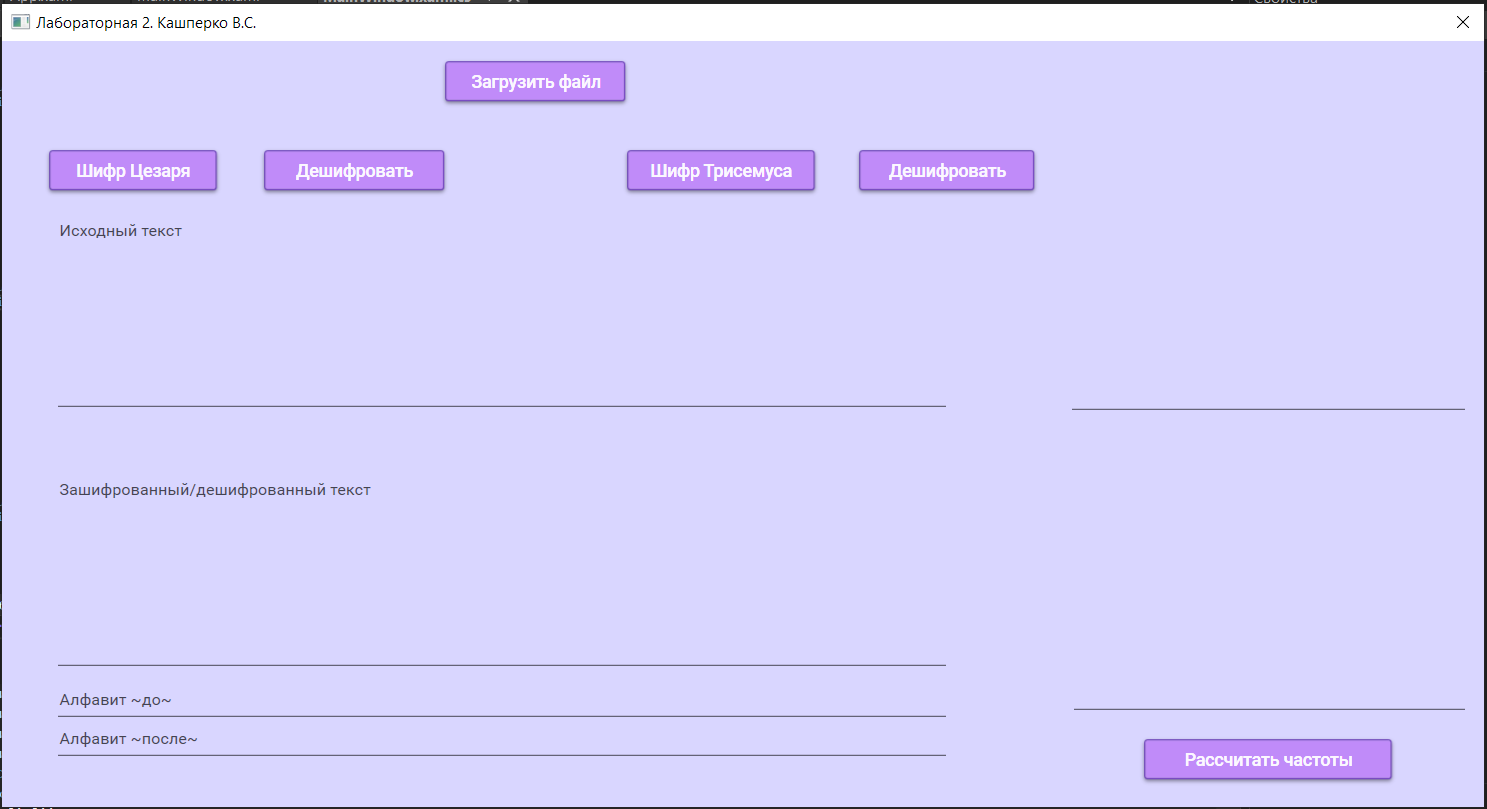


Рисунок 1 – Интерфейс программы

Для удобства был встроен инструмент OpenFileDialog для того, чтобы легко выбирать документ для работы (Рисунок 2).

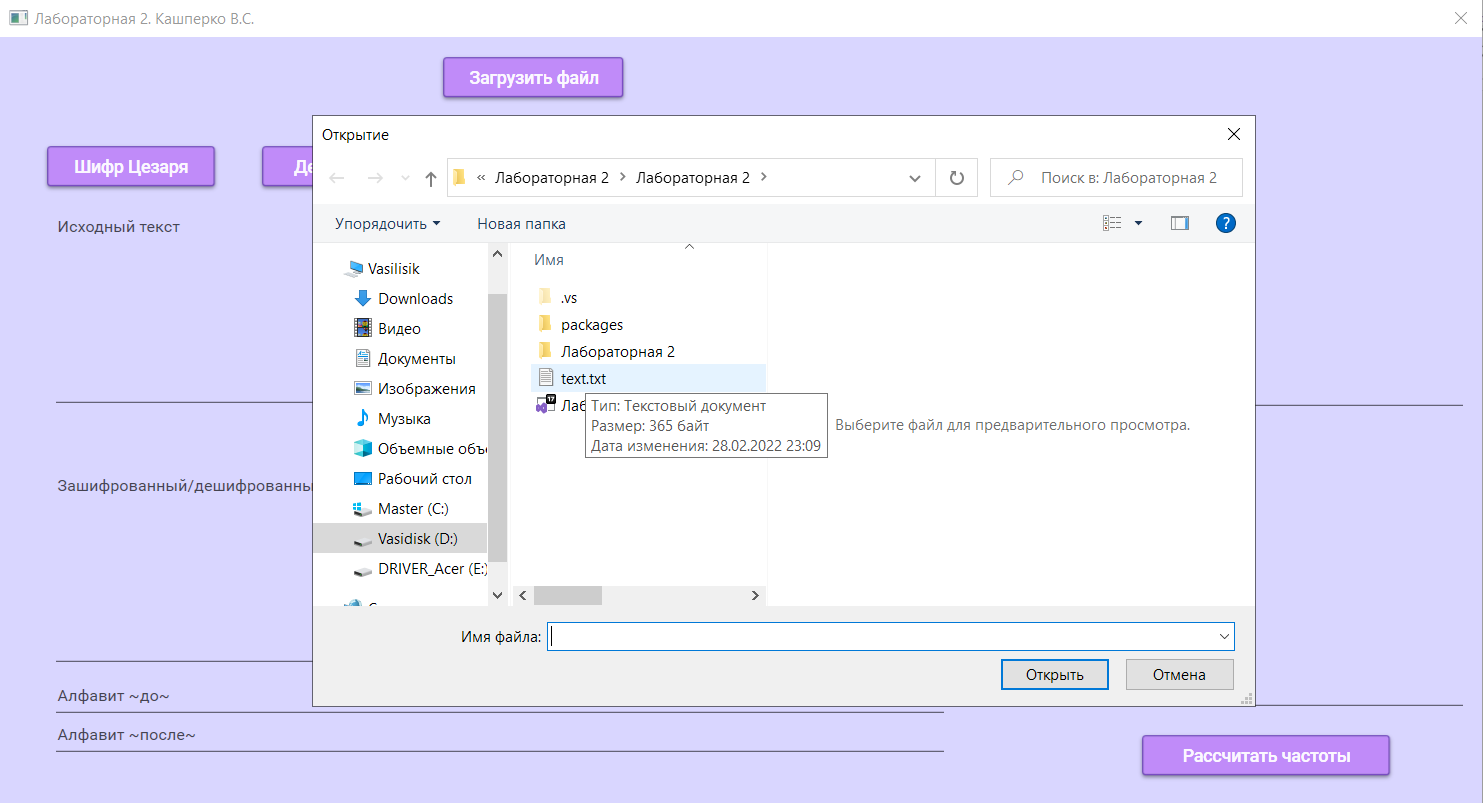


Рисунок 2 – Работа инструмента OpenFileDialog

После загрузки файла – текст передается в текстовое поле для дальнейшей работы (Рисунок 3).

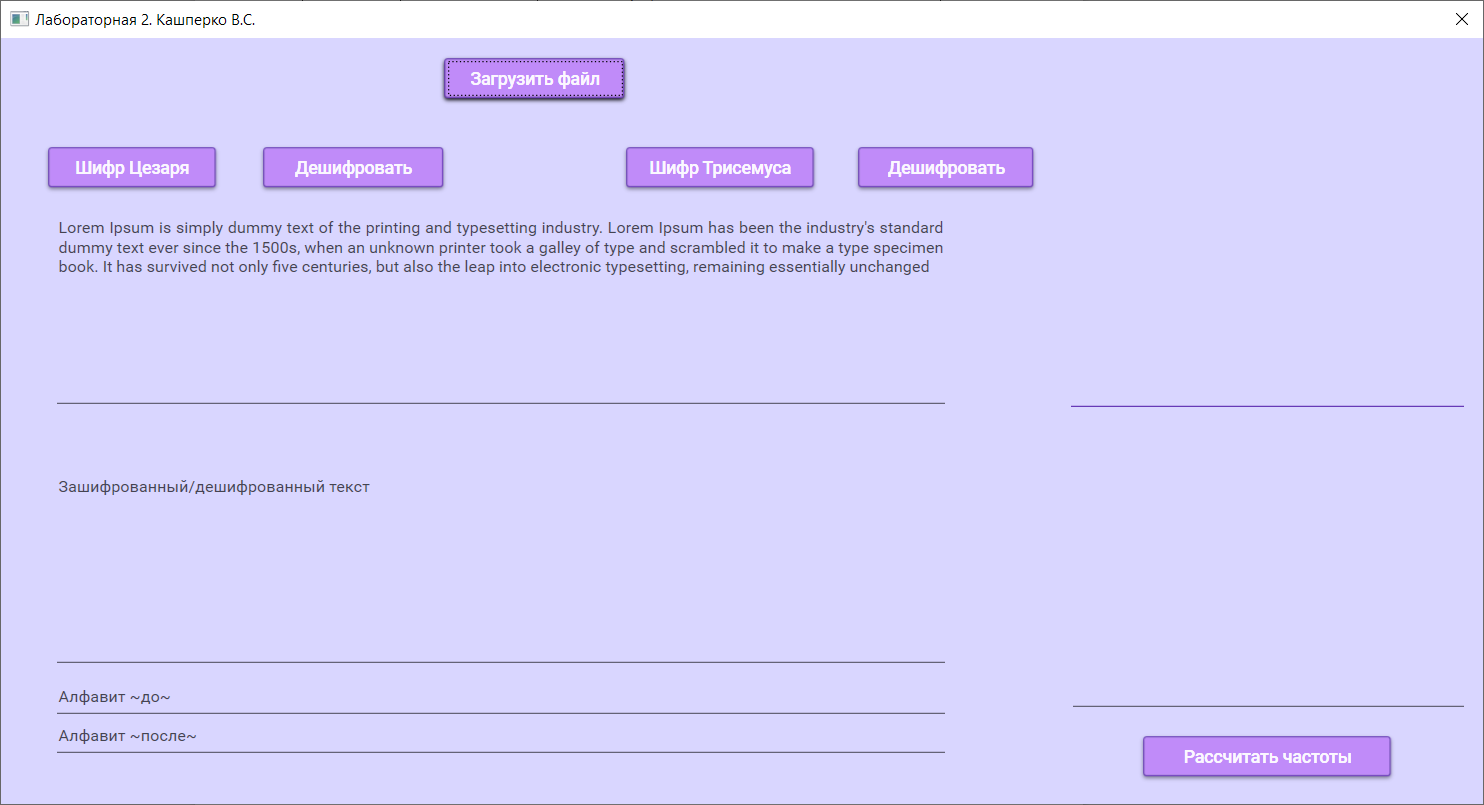


Рисунок 3 – Текст файла загружен в TextBox

Далее, по нажатию на кнопку «Шифр Цезаря» мы можем выполнить зашифрование исходного текста и изучить затраченное время. На рисунке 4 изображен результат, после нажатия на кнопку «Цифр Цезаря». Также приведен листинг 1 с алгоритмом шифрования.

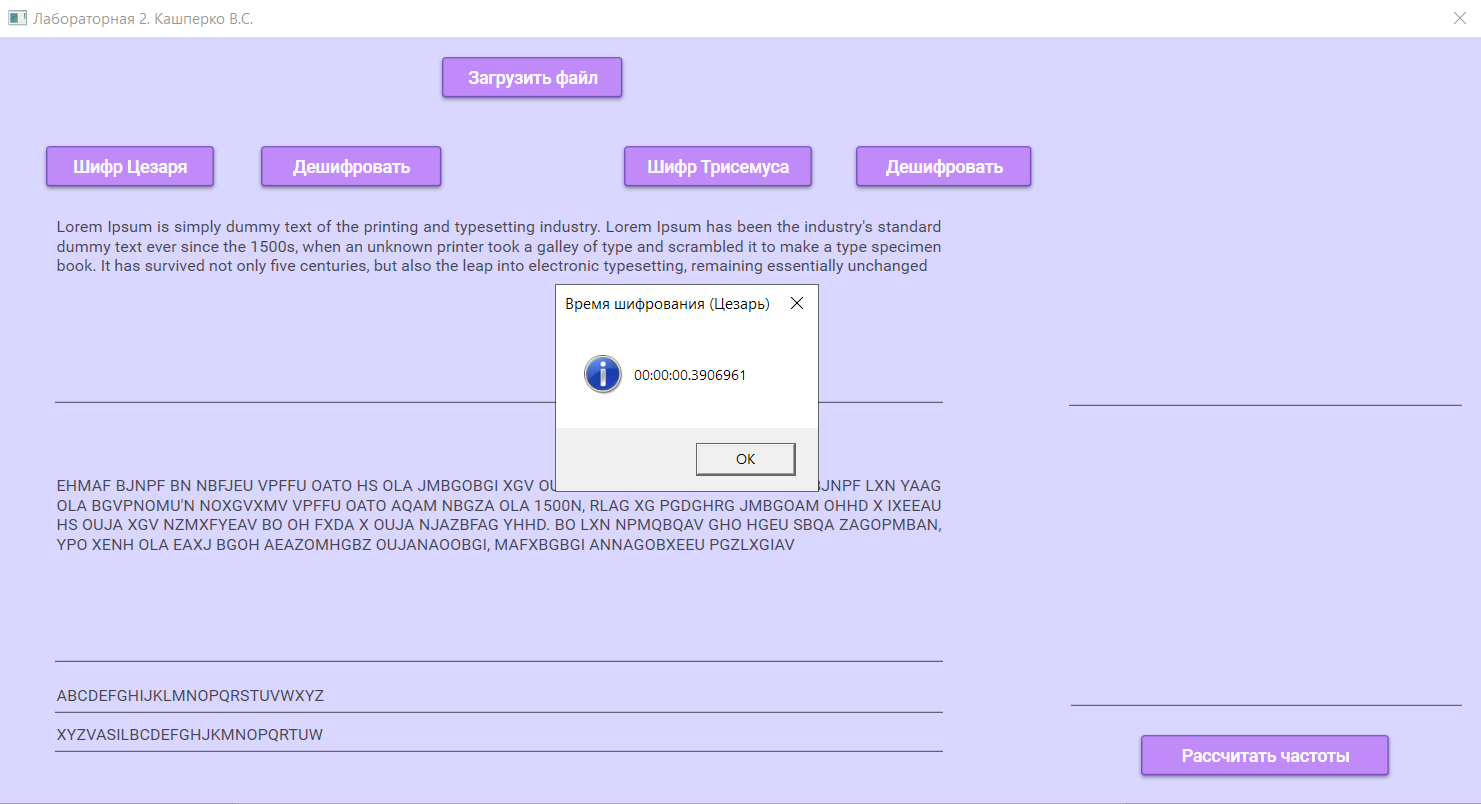


Рисунок 4 – Зашифрование исходного текста после нажатия на кнопку «Шифр Цезаря»

|  |
| --- |
| void CesarEncrypt(string text)  {  if (text != null)  {  text = text.ToUpper();  char[] a = alphabet.ToCharArray();  char[] b = alphabetCaesar.ToCharArray();  char[] c = text.ToCharArray();  for (int i = 0; i < c.Length; i++)  {  if (alphabet.Contains(c[i]))  {  int pos = alphabet.IndexOf(c[i]);  c[i] = b[pos];  }  outputTextBox.Text += c[i];  }  }  } |

Листинг 1 – Реализация алгоритма шифрования Цезаря

Система шифрования Цезаря с ключевым словом (лозунгом). Также является одноалфавитной системой подстановки. Особенностью этой системы является использование ключевого слова (лозунга) для смещения и изменения порядка символов в алфавите подстановки (желательно, чтобы все буквы ключевого слова были различными).

После чего можно выполнить дешифрование зашифрованного текста и также увидеть затраченное на эту операцию время (Рисунок 5). В листинге 2 содержится функция дешифрования.



Рисунок 5 – Дешифрование исходного текста после нажатия на кнопку «Дешифровать»

|  |
| --- |
| void CesarDecrypt(string text)  {  if (text != null)  {  text = text.ToUpper();  char[] a = alphabet.ToCharArray();  char[] b = alphabetCaesar.ToCharArray();  char[] c = text.ToCharArray();  for (int i = 0; i < c.Length; i++)  {  if (alphabet.Contains(c[i]))  {  int pos = alphabetCaesar.IndexOf(c[i]);  c[i] = a[pos];  }  inputTextBox.Text += c[i];  }  }  } |

Листинг 2 – Реализация алгоритм дешифрования Цезаря

Далее по нажатию на кнопку «Шифр Трисемуса» (рисунок 6) реализуется алгоритм шифрования, содержащийся в листинге 3. После нажатия на кнопку «Дешифровать» (рисунок 7) реализуется алгоритм дешифрования, также содержащийся в листинге 3.

Зашифрование осуществляется так: заготавливается таблица подстановки (так называемая «таблица Трисемуса» – таблица со стороной, равной N, где N – мощность алфавита), где первая строка – это алфавит, вторая – алфавит, сдвинутый на один символ, и т. д. При зашифровании первая буква открытого текста заменяется на букву, стоящую в первой строке, вторая – на букву, стоящую во второй строке, и т. д. После использования последней строки вновь возвращаются к первой.

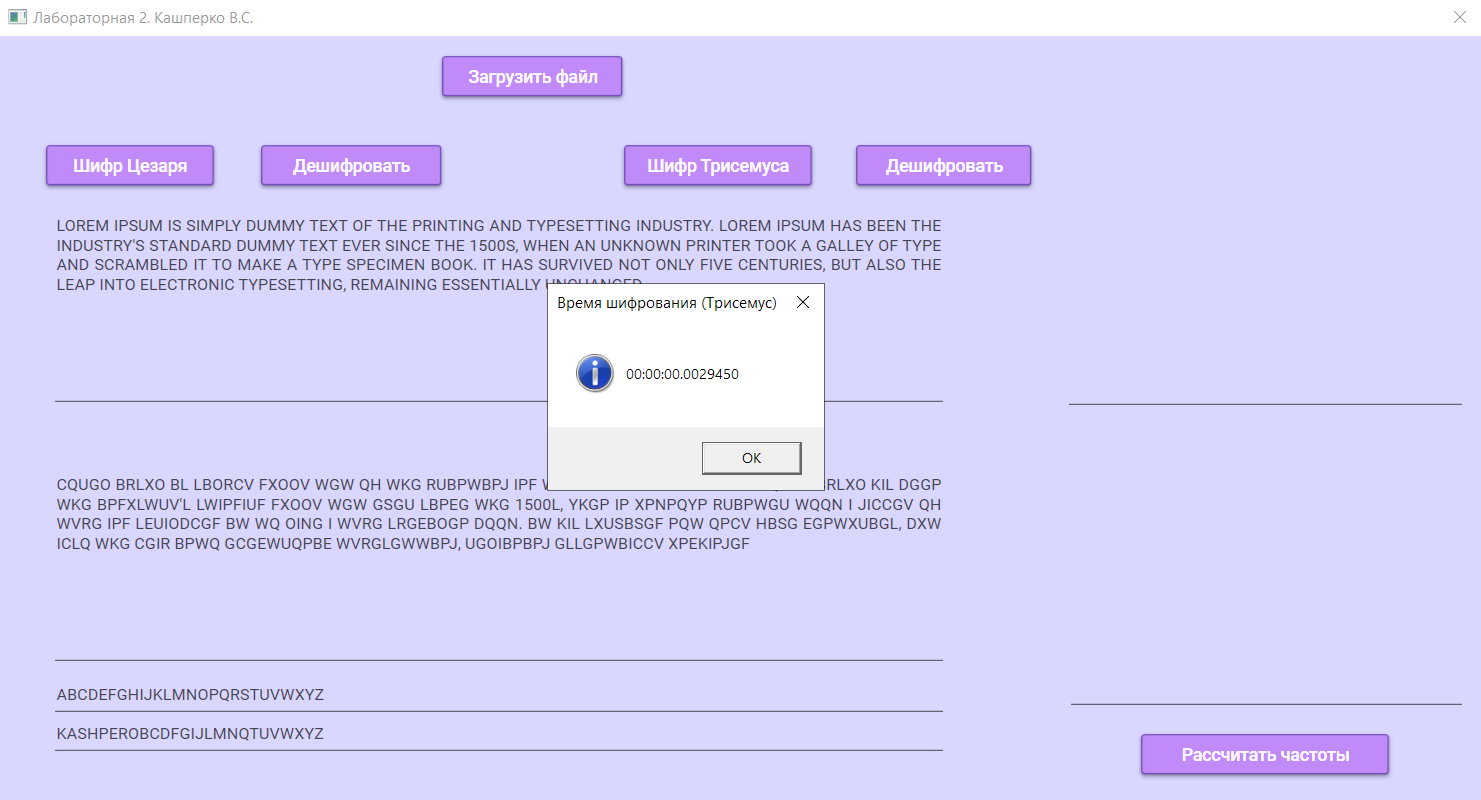


Рисунок 6 – Зашифрование исходного текста после нажатия на кнопку «Шифр Трисемуса»

|  |
| --- |
| void TrisemusEncrypt(string text)  {  if (text != null)  {  char[] alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYX".ToCharArray();  int rows = 13, columns = 2;  String ke = "vasilisa";  char[] keyWord = ke.ToUpper().Distinct().ToArray();  var table = new char[rows, columns];  for (var i = 0; i < keyWord.Length; i++)  {  table[i / columns, i % columns] = keyWord[i];  }  alphabet = alphabet.Except(keyWord).ToArray();  for (var i = 0; i < alphabet.Length; i++)  {  int position = i + keyWord.Length;  table[position / columns, position % columns] = alphabet[i];  }  message = text.ToUpper();  StringBuilder sb = new StringBuilder();  for (int l = 0; l < message.Length - l; l += 255)  {  message.Substring(l, l + 255);  var result = new char[message.Length];  for (var k = 0; k < message.Length; k++)  {  char symbol = message[k];  for (var i = 0; i < rows; i++)  {  for (var j = 0; j < columns; j++)  {  if (symbol == table[i, j])  {  symbol = table[(i + 1) % rows, j];  i = rows;  break;  }  }  }  result[k] = symbol;  }  ress = sb.Append(result).ToString();  break;  }  outputTextBox.Text = ress;  }  }  void TrisemusDecrypt(string text)  {  char[] alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYX".ToCharArray();  int rows = 13, columns = 2;  String ke = "eni";  char[] keyWord = ke.ToUpper().Distinct().ToArray();  var table = new char[rows, columns];  for (var i = 0; i < keyWord.Length; i++)  {  table[i / columns, i % columns] = keyWord[i];  }  alphabet = alphabet.Except(keyWord).ToArray();  for (var i = 0; i < alphabet.Length; i++)  {  int position = i + keyWord.Length;  table[position / columns, position % columns] = alphabet[i];  }  messages = text.ToUpper();  StringBuilder sbs = new StringBuilder();  for (int m = 0; m < messages.Length - m; m += 255)  {  messages.Substring(m, m + 255);  var results = new char[messages.Length];  for (var k = 0; k < messages.Length; k++)  {  char symbol = messages[k];  for (var i = 0; i < rows; i++)  {  for (var j = 0; j < columns; j++)  {  if (symbol == table[i, j])  {  symbol = table[(i + 10) % rows, j];  i = rows;  break;  }  }  }  results[k] = symbol;  }  deress = sbs.Append(results).ToString();  break;  }  } |

Листинг 3 – Реализация алгоритма шифрования и дешифрования Трисемуса

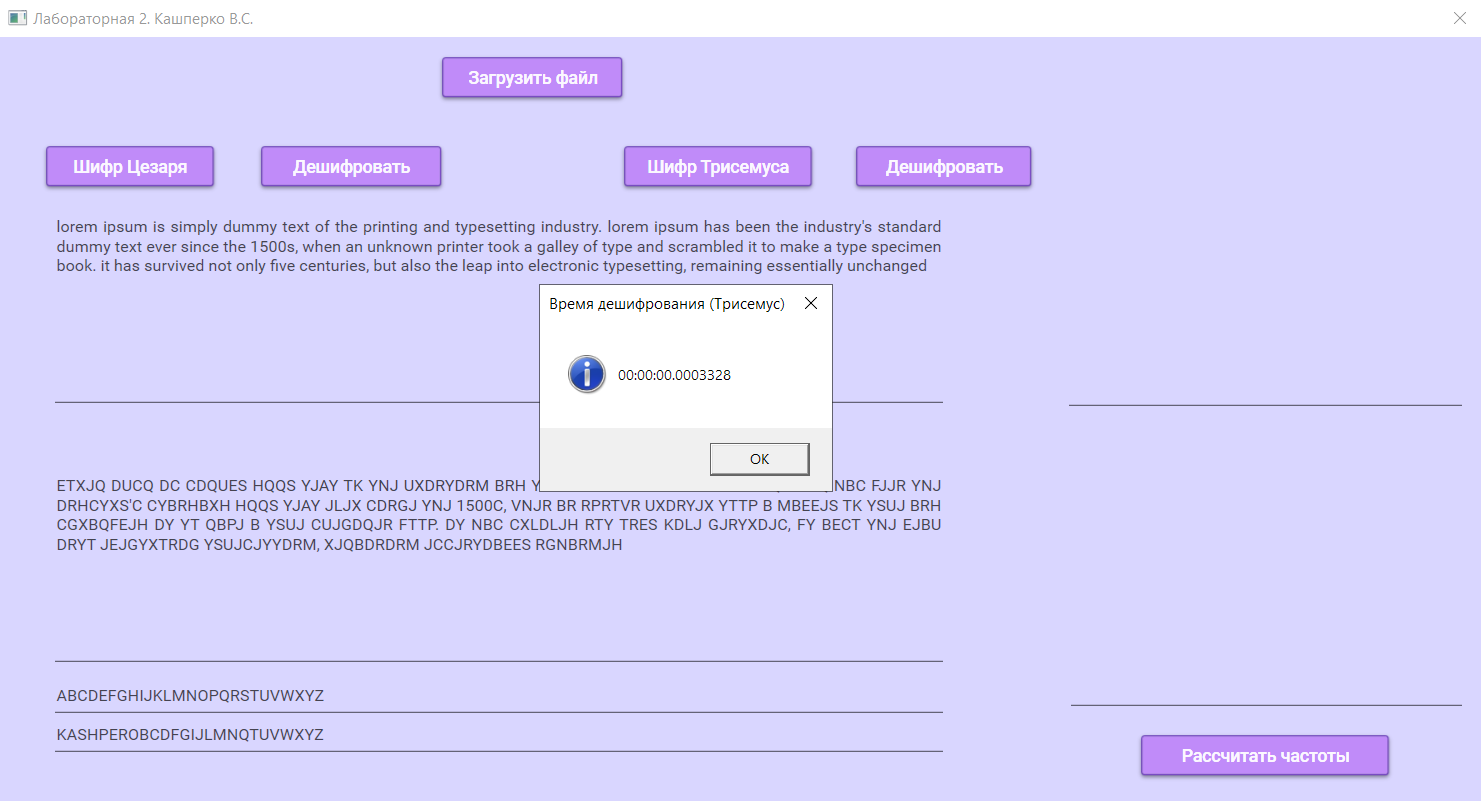


Рисунок 7 – Интерфейс программы

Для того, чтобы сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений существует кнопка «Рассчитать частоты» – после нажатия на нее в текстовых полях формируются словари типа Dictionary ключ-значение. Ключ – буква алфавита, значение – частота появления символа (рисунок 8).

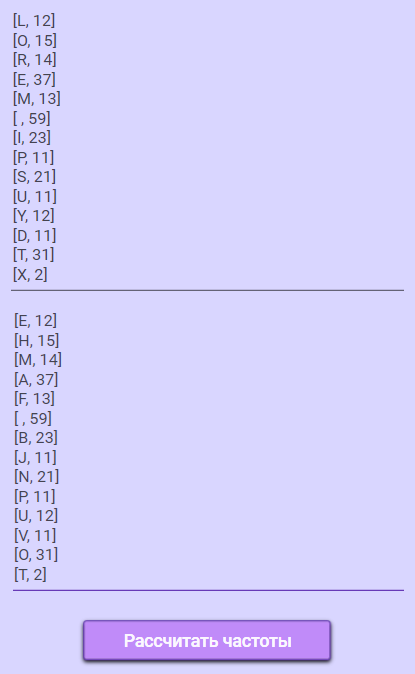


Рисунок 8 – Расчёт частот появления символов

На рисунке 9 представлены гистограммы частот появления символов.

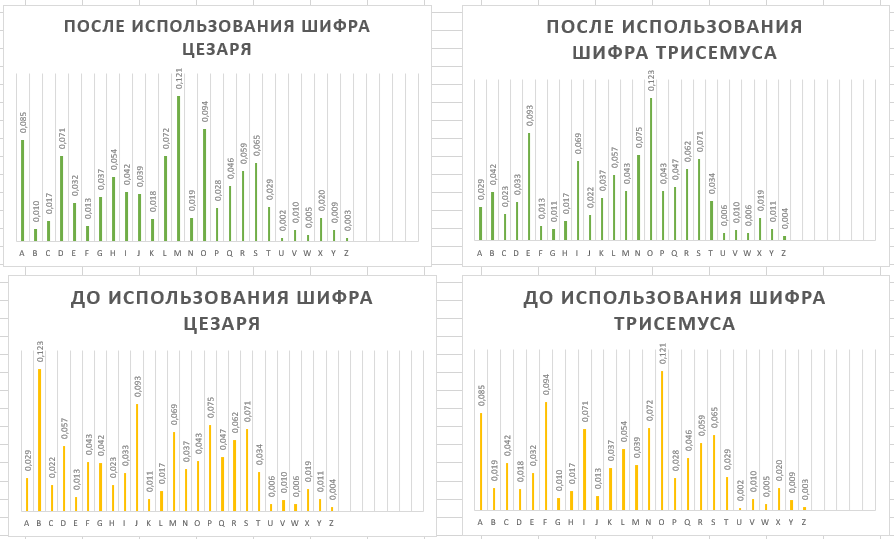


Рисунок 9 – Гистограммы появления частот символов

Ответы на контрольные вопросы:

1. **В чем заключается основная идея криптографических преобразований на основе шифров замены?**

Одни символы (или группа символов) заменяются на другие по какому-либо алгоритму.

1. **Привести классификационные признаки и дать сравнительную характеристику разновидностям подстановочных шифров**

Подстановочные шифры делятся на моноалфавитные, полиграммные, омофонические и полиалфавитные. В моноалфавитных символы сообщения заменяются на другие символы того же алфавита. Для полиграммных шифров характерна замена группы символов и на символ того же или другого алфавита. В омофонических шифрах производят замену сходных по звучанию частей сообщения. В полиалфавитных для шифрования используется 2 и более алфавита.

Наиболее сложными для реализации, но также наиболее криптостойкими, являются омофонические шифры, так как сложно создать алгоритм, определяющий созвучие слов или групп символов. Наименее криптостойкими являются моноалфавитные шифры, так как легко проанализировать частоту встречаемости символов, за счёт чего сообщение и можно расшифровать.

1. **Сколько разновидностей шифров, подобных шифру Цезаря, можно составить для алфавитов русского и белорусского языков?**

Существует 4 разновидности шифра Цезаря, но так как мы можем комбинировать несколько шифров, необходимо найти сумму сочетаний, умноженную на 2, так как 2 алфавита:

1. **Можно ли использовать в качестве ключевого в шифре Виженера слово, равное по длине открытому тексту?**

Да

1. **По какому признаку можно определить, что текст зашифрован шифром Плейфера?**

Так как шифр Плейфера работает с биграммами, в зашифрованном сообщении будут в среднем чаще встречаться двойные повторения символов (например, AA или VV). Частотный анализ большого шифра также покажет, что распределение вероятностей встречи символов по числам значительно отличается от алфавитного, что не позволяет установить соответствие между символом шифра и символом алфавита по вероятности его встречи в тексте.

1. **Имеются ли предпочтения в выборе размеров таблицы Трисемуса для виртуального алфавита мощностью 40: 4×10; 10×4; 5×8; 8×5; 2×20; 20×2?**

Разницы нет, но стоит учитывать, что при выборе разных размерностей таблиц получатся разные шифры.

1. **Охарактеризовать основные виды атак на шифры**

**Атака с известным шифротекстом** – известен алгоритм, имеет образец шифра, неизвестен ключ.

**Атака с выбором шифротекста** – имеется большое количество шифротекста, из которого можно выбрать только нужную часть.

**Адаптивная атака с выбором шифротекста** - имеется возможность выбирать новые шифрограммы для расшифрования с учетом того, что известна некоторая информация из предыдущих сообщений.

**Атака с известным открытым шифротекстом** - то же, что и предыдущая, но для некоторых шифрограмм имеются соответствующие им открытые тексты.

**Атака с выбором открытого текста** – имеется несколько открытых текстов и соответствующих им шифротекстов.

**Адаптивная атака с выбором открытого текста** – имеется возможность выбирать новые шифрограммы и соответствующий им открытый текст, при этом известна некоторая информация из предыдущих сообщений.

**Атака на основе связанных ключей** – неизвестны ключи, но известно соотношение или связь между ними.

**Атака с выбором ключа** – часть ключа задаётся при взломе.

1. **Сравнить криптостойкость шифра Цезаря и шифра Виженера**

Шифр Цезаря имеет меньшую криптостойкость, так как его легко взломать брутфорсом (прямом перебором значений) или с помощью частотного анализа, если имеется большой образец зашифрованной инфоомации.

1. **Охарактеризовать основные методы взлома подстановочных шифров**

**Брутфорс** – прямой перебор всех возможных вариантов.

**Частотный анализ** – установка однозначного соответствия между символами шифра и алфавита на основе схожего числа вероятности встречи символа в тексте.

**Анализ встречаемости триграмм** – лучше всего подходит для зашифрованного английского текста. Так как триграммы THE, AND, ING и другие встречаются в английском языке достаточно часто, необходимо установить соответствие между буквами триграмм и буквами шифра (как правило на основе частотного анализа). Соответствие между оставшимися символами устанавливается на основе уже установленных.

**Поиск восхождением к вершине** – берётся часть шифротекста (называемая ключом), и оставшаяся часть расшифровывается с его помощью. Затем вычисляется коэффициент вероятности принадлежности расшифрованного текста к естественному языку. В ключе производятся какие-либо изменения (например, перестановка первых двух букв местами) и снова расшифровывается текст и вычисляется коэффициент. Так повторяется, пока коэффициент не перестанет изменяться.

**Вывод:** в данной лабораторной работе были закреплены теоретические знания по подстановочным шифрам. Также было разработано приложение для работы с таблицей Трисемуса и шифром Цезаря.