Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №4**

**«Исследование криптографических шифров на основе подстановки (замены) символов»**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

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

Задачи:

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости подстановочных шифров.

2. Ознакомиться с особенностями реализации и свойствами

различных подстановочных шифров на основе готового программного средства (L\_LUX).

3. Разработать приложение для реализации указанных преподавателем методов подстановочного зашифрования/расшифрования.

4. Выполнить исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях.

5. Оценить скорость зашифрования/расшифрования реализованных способов шифров.

6. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**2. Теоретические сведения**

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 замены производится раздельно над каждым одиночным символом сообщения Мi.

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

Для математического описания криптографического преобразования предполагаем, что зашифрованная буква ay (ay ∈ Сi), соответствующая символу aх (aх ∈ Мi), находится на позиции

y ≡ x + k mod N,

где x, y – индекс (порядковый номер, начиная с 0) символа в используемом алфавите; k – ключ.

Для расшифрования сообщения Сi необходимо произвести расчеты, обратные выражению, т. е.

х ≡ у – k mod N.

Одним из существенных недостатков моноалфавитных шифров является их низкая криптостойкость. Зачастую метод криптоанализа базируется на частоте встречаемости букв исходного текста.

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). Ключевое слово пишется в начале алфавита подстановки.

Применяя одновременно операции сложения и умножения по модулю n над элементами множества (индексами букв алфавита), можно получить систему подстановок, которую называют аффинной системой подстановок Цезаря. Определим процедуру зашифрования в такой системе:

y ≡ ax + b mod N,

где a и b – целые числа. При этом взаимно однозначные соответствия между открытым текстом и шифртекстом будут иметь место только при выполнении следующих условий: 0 ≤ a, b < N, наибольший общий делитель (НОД) чисел a, N равен 1, т. е. эти числа являются взаимно простыми.

Полиалфавитные (или многоалфавитные) шифры состоят из нескольких шифров однозначной замены. Выбор варианта алфавита для зашифрования одного символа зависит от особенностей метода шифрования.

Таблица Трисемуса. 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. При шифровании с ключевым словом буква открытого текста заменяется буквой, расположенной ниже нее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.

Шифр Виженера. В этом шифре мы имеем дело с последовательностью сдвигов, циклически повторяющейся. Основная идея заключается в следующем. Создается таблица (таблица Виженера) размером N×N (N – число знаков в используемом алфавите). Эти знаки могут включать не только буквы, но и, например, пробел или иные знаки. В первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущего циклического сдвига последней на 1 символ влево.

Криптоанализ – это раздел криптологии, занимающийся методами взлома шифров или методами организации криптографических атак на шифры.

Основные криптоатаки:

* Атака с известным шифртекстом;
* Атака с выбором шифртекста;
* Адаптивная атака с выбором шифртекста;
* Атака с известным открытым текстом;
* Атака с выбором открытого текста;
* Адаптивная атака с выбором открытого текста;
* Атака на основе связанных ключей;
* Атака с выбором ключа.

**3. Практическая часть**

**Практическое задание:**

1. Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции:

• выполнять зашифрование/расшифрование текстовых документов (объемом не менее 5 тысяч знаков), созданных на основе алфавита языка в соответствии с нижеследующей таблицей вариантов задания; при этом следует использовать шифры подстановки из третьего столбца данной таблицы

Алфавит: Английский.

Шифры:

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

2. Таблица Трисемуса, ключевое слово – собственное имя

• сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений;

• оценить время выполнения операций зашифрования/расшифрования (напоминание: во многих языках программирования есть встроенные методы для замеров времени; при отсутствии такового в используемом языке можно воспользоваться разностью двух дат (например, в миллисекундах: время после выполнения программы – время до начала выполнения преобразования))

2. Результаты оформить в виде отчета по установленным правилам.

**Ход работы**

Для реализации шифра Цезаря было выбрано ключевое слово «GURINA». Для создания подстановочного алфавита для шифра Цезаря была разработана функция createSubstitutionAlphabet, представленная на рисунке 3.1.

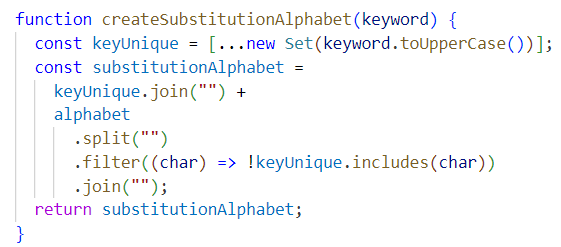


Рисунок 3.1 – Код функции createSubstitutionAlphabet

Функция createSubstitutionAlphabet(keyword) создает новый алфавит на основе ключевого слова, удаляя дубликаты и добавляя оставшиеся буквы из исходного алфавита.

Для шифрования текста с помощью шифра Цезаря была разработана функция encrypt, представленная на рисунке 3.2.

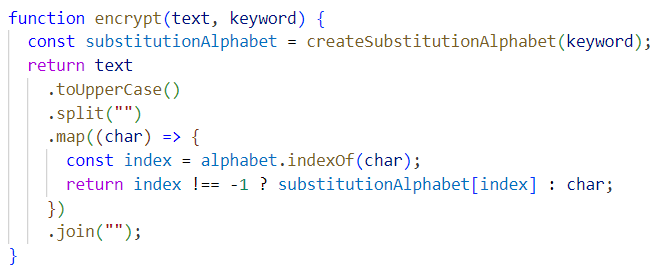


Рисунок 3.2 – Код функции encrypt

Функция encrypt(text, keyword) шифрует текст, заменяя каждую букву на соответствующую букву из подстановочного алфавита.

Для расшифровки была реализована функция decrypt. Функция decrypt(text, keyword) выполняет обратное преобразование, заменяя каждую букву на соответствующую букву из исходного алфавита. Код функции можно увидеть на рисунке 3.3.

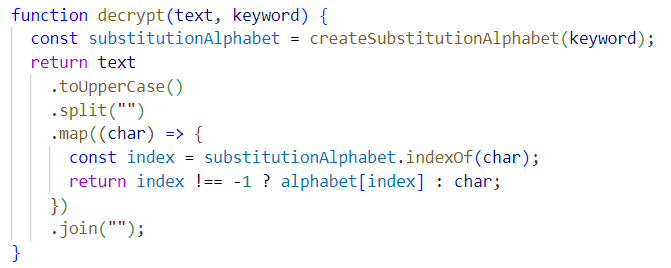


Рисунок 3.3 – Код функции decrypt

Функция encryptFile(inputFile, outputFile, keyword) считывает текст из исходного файла с помощью функции fs.readFileSync, шифрует его и сохраняет результат в новый файл с помощью функции fs.writeFileSync. Таже эта функция рассчитывает вероятности появления каждого символа в зашифрованном тексте и выводит их в консоль. Результат работы функции представлен на рисунке 3.4.

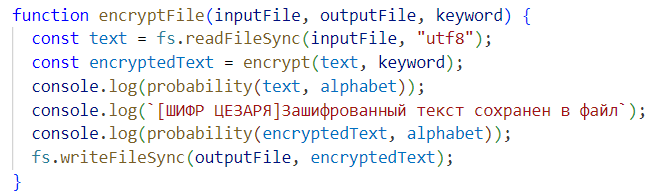
****

Рисунок 3.4 – Код функции encryptFile

Фрагмент, закодированного с помощью функции encryptFile, файла представлен на рисунке 3.5.

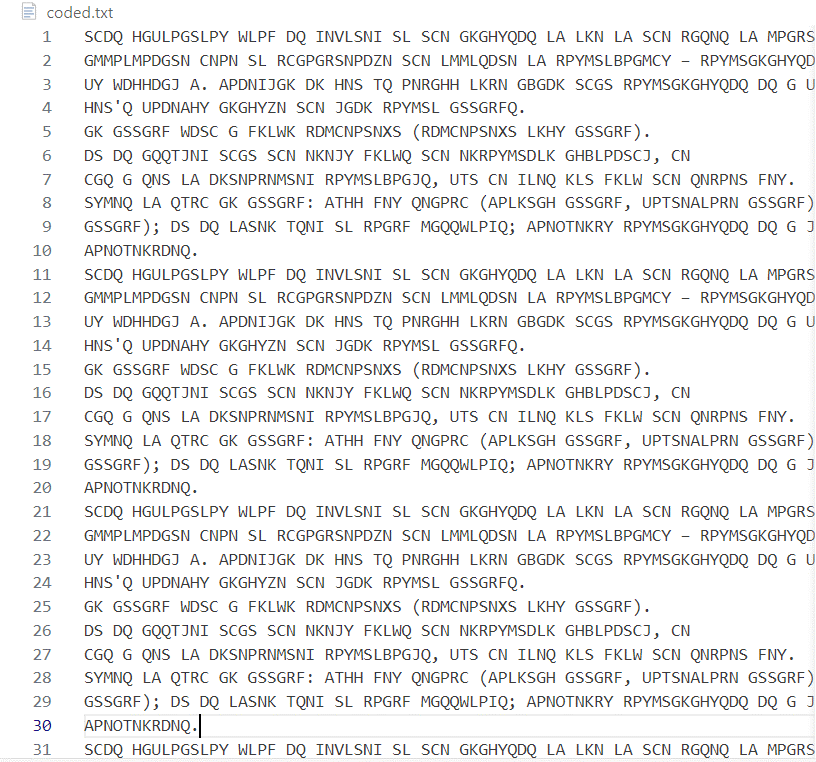


Рисунок 3.5 – Фрагмент закодированного файла

Функция decryptFile(inputFile, outputFile, keyword) считывает шифрованный текст, дешифрует его и сохраняет результат в файл. Код функции можно увидеть на рисунке 3.6.

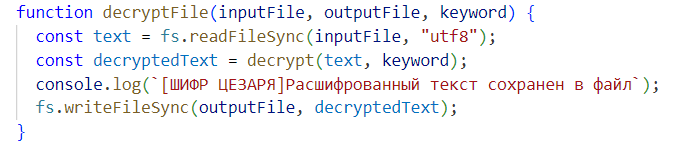


Рисунок 3.6 – Код функции decryptFile

Фрагмент, раскодированного с помощью функции decryptFile, файла представлен на рисунке 3.7.

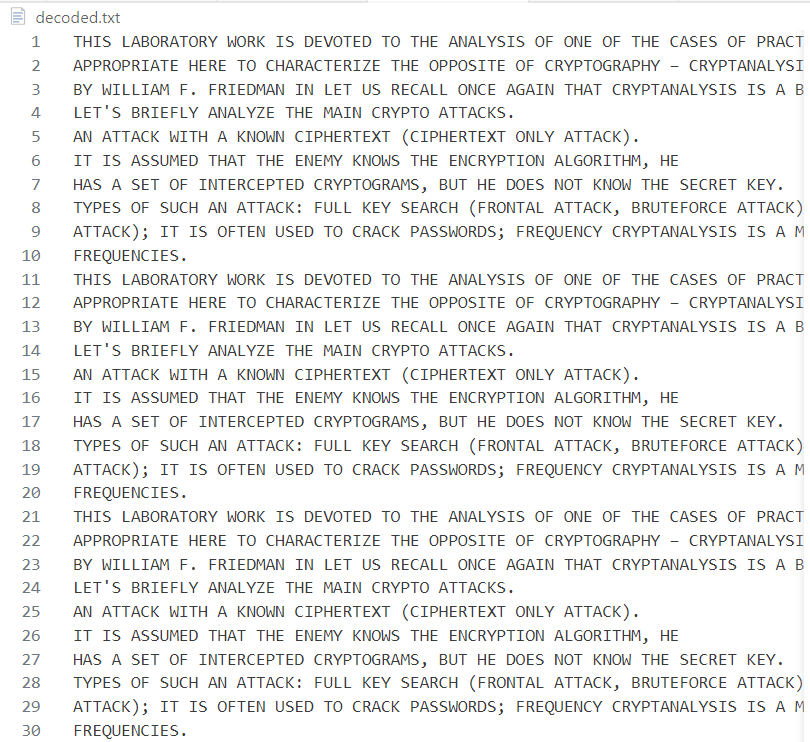


Рисунок 3.7 – Фрагмент раскодированного файла

Время выполнения шифрования и расшифрования с использованием шифра Цезаря с ключом представлено на рисунке 3.8

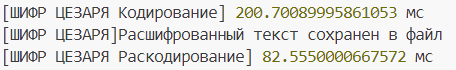


Рисунок 3.8 – Время работы алгоритма

Далее, нужно было сформировать гистограммы частот появления символов для исходного и зашифрованного сообщений. Гистограмма частот появления символов исходного сообщения приведена на рисунке 3.9.

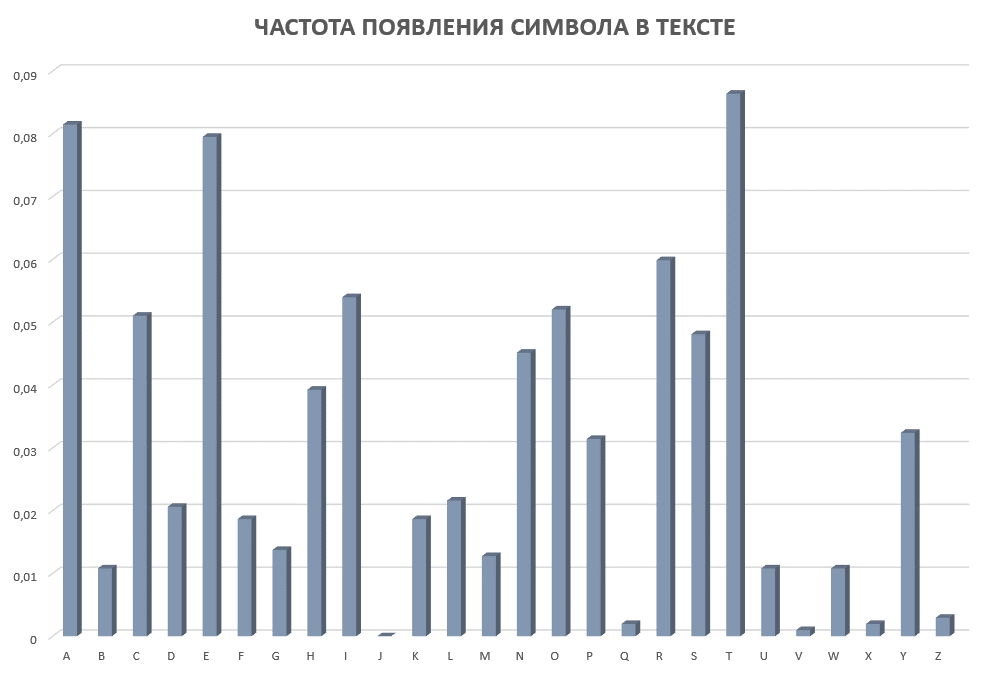


Рисунок 3.9 – Гистограмма частот появления символов в исходном сообщении

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 3.10.

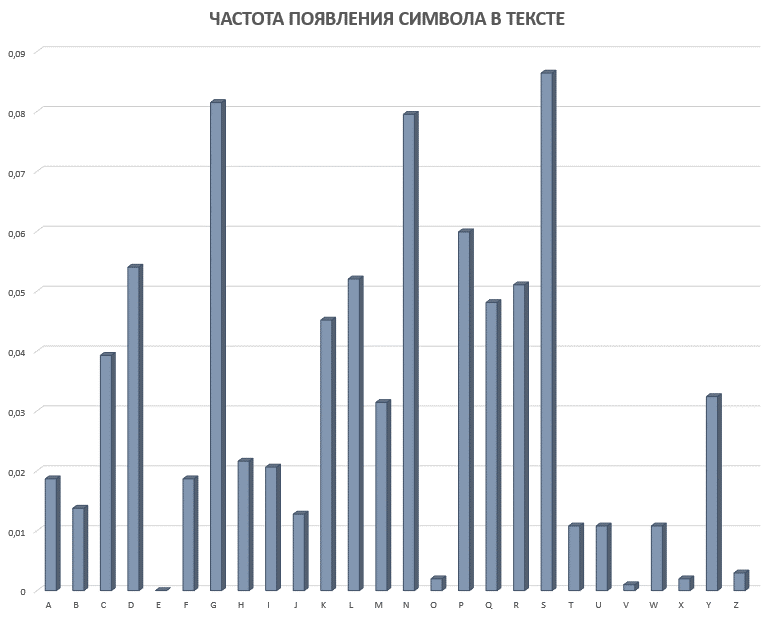


Рисунок 3.10 – Гистограмма частот появления символов в зашифрованном сообщении

Сопоставив гистограммы частот появления символов исходного и зашифрованного сообщений можно определить алфавит подстановки.

Затем были разработаны несколько функций, выполняющих шифрование и дешифрование текста с использованием таблицы Трисемуса, где ключевое слово – собственное имя.

Таблица Трисемуса создается вручную и представляет собой матрицу символов, в которой ключевое слово (в данном случае, собственное имя) используется для заполнения таблицы, а оставшиеся буквы алфавита заполняют остальные ячейки. Таблица Трисему представлена на рисунке 3.11.

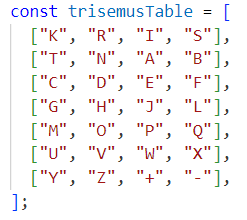


Рисунок 3.11 – Таблица Трисемуса

Была разработана функция encodeDecodeText(text, isEncoding) выполняет шифрование и дешифрование текста в зависимости от значения параметра isEncoding. Её код представлен на рисунке 3.12.

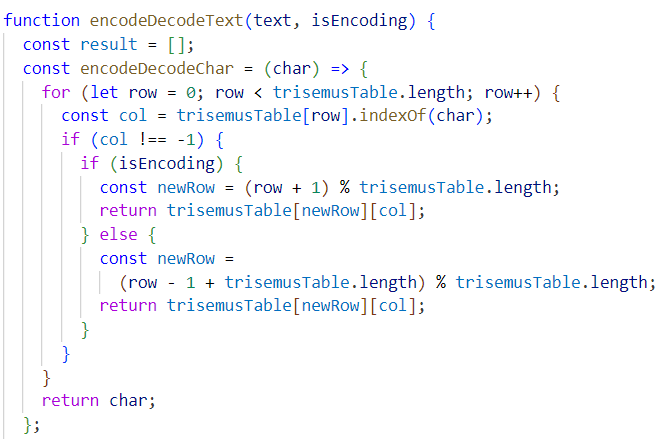


Рисунок 3.12 – Код функции encodeDecodeText

Для каждого символа входного текста находится его позиция в таблице Трисемуса, и он заменяется символом из строки ниже (при шифровании) или строки выше (при дешифровании). Если символ находится в последней строке, он заменяется символом из первой строки (и наоборот при дешифровании).

Аналогичные функциями чтения/записи в файл были созданы и для этого алгоритма. Функция encryptFile2(inputFile, outputFile) выполняет шифрование текста, прочитанного из файла inputFile, и сохраняет зашифрованный текст в файл outputFile. Чтение текста из файла осуществляется с помощью функции fs.readFileSync, после чего текст шифруется с использованием функции encodeDecodeText с параметром true, что указывает на шифрование. Фрагмент закодированного файла представлен на рисунке 3.13

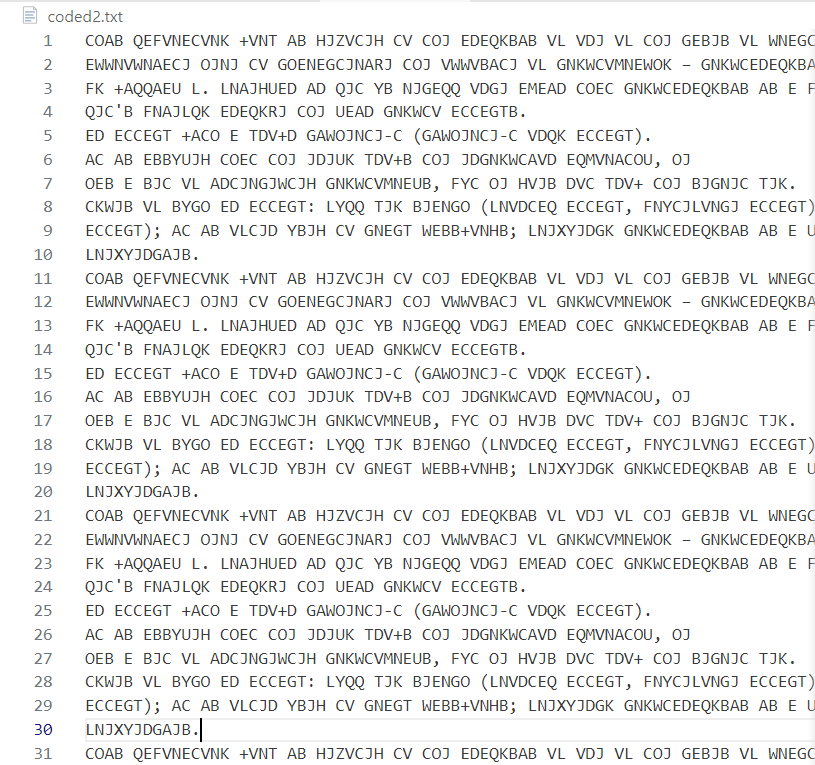


Рисунок 3.13 – Фрагмент закодированного файла

Функция decryptFile2(inputFile, outputFile) выполняет дешифрование текста, прочитанного из файла inputFile, и сохраняет расшифрованный текст в файл outputFile. Чтение текста из файла осуществляется с помощью функции fs.readFileSync, после чего текст дешифруется с использованием функции encodeDecodeText с параметром false, что указывает на дешифрование. Фрагмент раскодированного файла представлен на рисунке 3.14.

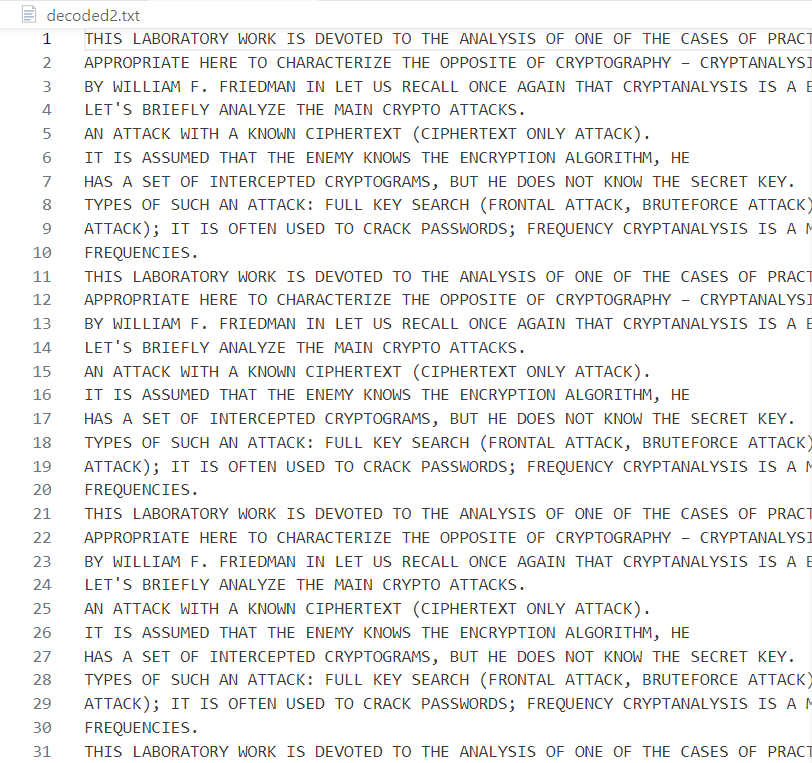


Рисунок 3.14 – Фрагмент раскодированного файла

Для измерения времени выполнения операций шифрования и дешифрования используются функции performance.now(), которые фиксируют время до и после выполнения операций, что позволяет вычислить затраченное время.

Гистограмма частот появления символов исходного сообщения приведена на рисунке 3.15.

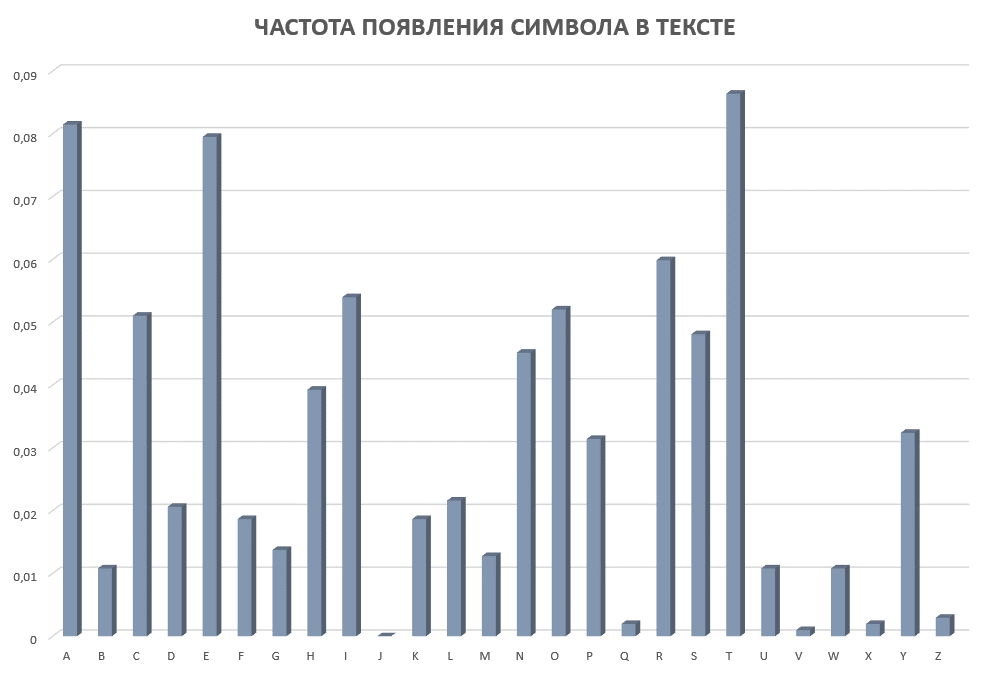


Рисунок 3.15 – Гистограмма частот появления символов в исходном сообщении

Гистограмма частот появления символов зашифрованного сообщения приведена на рисунке 3.16.

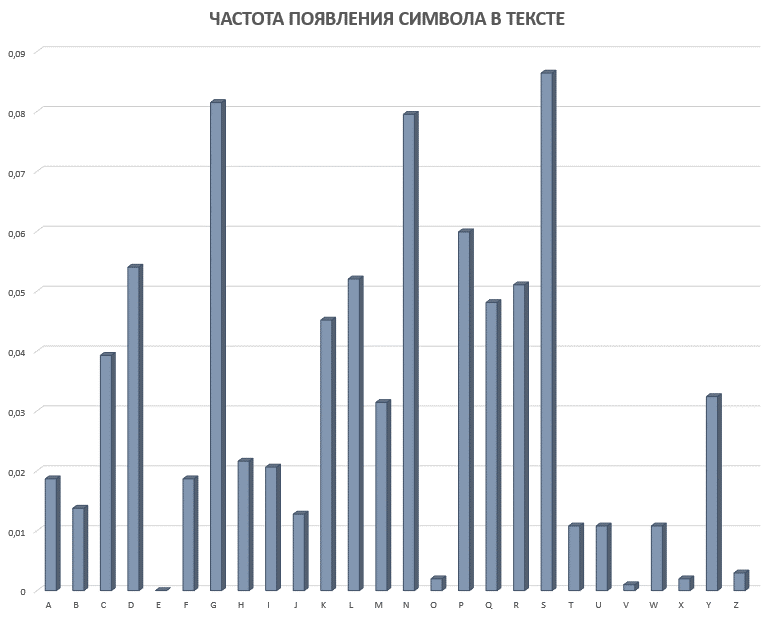


Рисунок 3.16 – Гистограмма частот появления символов в зашифрованном сообщении

Сопоставив гистограммы частот появления символов исходного и зашифрованного сообщений также можно определить алфавит подстановки.

**Вывод**

В ходе выполнения лабораторной работы были изучены основные принципы работы подстановочных шифров, методы их зашифрования и расшифрования. Также было разработано приложение для реализации указанных методов подстановочного зашифрования и расшифрования. Было выполнено исследование криптостойкости шифров на основе статистических данных о частотах появления символов в исходном и зашифрованном сообщениях. Также работы были реализованы и протестированы два криптографических метода подстановки: шифр Цезаря с ключевым словом и таблица Трисемуса. Для шифрования и дешифрования текста использовались собственные фамилия и имя соответственно. Шифр Цезаря показал простоту реализации и использования, однако он также имеет слабую криптостойкость из-за ограниченного набора возможных ключей. Таблица Трисемуса, будучи более сложным методом подстановки, предоставляет лучшую защиту за счет использования матрицы символов и разнообразных подстановок. Была проведена оценка скорости зашифрования и расшифрования реализованных способов шифров. Это позволило сравнить эффективность различных алгоритмов.