Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образoвания

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра Информатики и защиты информации

Курсовая работа

по дисциплине «МСКЗИ»

**Выполнил:**

Студент группы ИСБ-119

Журавлев Д. А.

**Принял:**

Александров А.В.

**Владимир 2021**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc90921855)

[1.1. Криптология 3](#_Toc90921856)

[1.2. Принцип Керкгоффса 4](#_Toc90921857)

[1.3. Модели криптографии 5](#_Toc90921858)

[АЛГОРИТМИЧЕСКАЯ ТЕОРИЯ СЛОЖНОСТИ 8](#_Toc90921859)

[2.1. Complexity 8](#_Toc90921860)

[2.2. Быстрое возведение в степень 9](#_Toc90921861)

[2.3. Факторизация числа 10](#_Toc90921862)

[КРИПТОАНАЛИЗ 12](#_Toc90921863)

[3.1. Взлом подстановки методом частотного анализа 12](#_Toc90921864)

[3.2. Атака Фридмана на гаммирование 16](#_Toc90921865)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 20](#_Toc90921866)

[ПРИЛОЖЕНИЕ 1 21](#_Toc90921867)

[ПРИЛОЖЕНИЕ 2 23](#_Toc90921868)

[ПРИЛОЖЕНИЕ 3 25](#_Toc90921869)

[ПРИЛОЖЕНИЕ 4 29](#_Toc90921870)

# ВВЕДЕНИЕ

## Криптология

С зарождением человеческой цивилизации возникла необходимость передачи информации одним людям так, чтобы она не становилась известной другим. С возникновением письменности задача обеспечения секретности и подлинности передаваемых сообщений стала особенно актуальной. Поэтому именно после возникновения письменности появилось искусство тайнописи, искусство "тайно писать" - набор методов, предназначенных для секретной передачи записанных сообщений от одного человека другому.  
Бурное раз­ви­тие крип­то­гра­фи­че­ские сис­те­мы по­лу­чи­ли в го­ды пер­вой и вто­рой ми­ро­вых войн. Начиная с послевоенного времени и по нынешний день появление вычислительных средств ускорило разработку и совершенствование криптографических методов.

По­че­му про­бле­ма ис­поль­зо­ва­ния крип­то­гра­фи­че­ских ме­то­дов в информационных системах (ИС) ста­ла в на­стоя­щий мо­мент осо­бо ак­ту­аль­на?

С од­ной сто­ро­ны, рас­ши­ри­лось ис­поль­зо­ва­ние ком­пь­ю­тер­ных се­тей, в частности глобальной сети Интернет, по ко­то­рым пе­ре­да­ют­ся боль­шие объ­е­мы ин­фор­ма­ции го­су­дар­ствен­но­го, во­ен­но­го, ком­мер­че­ско­го и ча­ст­но­го ха­рак­те­ра, не до­пус­каю­ще­го воз­мож­ность дос­ту­па к ней по­сто­рон­них лиц.

С дру­гой сто­ро­ны, по­яв­ле­ние но­вых мощ­ных ком­пь­ю­те­ров, тех­но­ло­гий се­те­вых и ней­рон­ных вы­чис­ле­ний сде­ла­ло воз­мож­ным дис­кре­ди­та­цию криптографических сис­тем еще не­дав­но счи­тав­ших­ся прак­ти­че­ски не раскрываемыми.

Про­бле­мой защиты информации путем ее преобразования за­ни­ма­ет­ся *крип­то­ло­гия* (*kr*y*p*to*s* - тай­ный, lo*g*o*s* - нау­ка). Криптология раз­де­ля­ет­ся на два на­прав­ле­ния - *крип­то­гра­фию* и *крип­тоа­на­лиз*. Це­ли этих на­прав­ле­ний прямо про­ти­во­по­лож­ны.

Если криптография занимается построением надёжных криптосистем, то цель же криптоанализа заключается во взломе существующих алгоритмы.

«Криптография — научная и практическая деятельность, связанная с разработкой криптографических средств защиты информации, а также анализом и обоснованием их криптографической стойкости. В отличие от организационных и других способов защиты информации, под криптографическими понимаются такие, которые используют математические методы преобразования информации. Криптоанализ — научная и практическая деятельность по исследованию криптографических алгоритмов с целью получения обоснованных оценок их криптографической стойкости. Криптология — понятие, объединяющее криптографию и криптоанализ»[3 -с. 47]

## Принцип Керкгоффса

Требования к криптосистеме впервые изложены в книге Керкгоффса «Военная криптография» (издана в 1883 году). Шесть основных требований к криптосистеме:

1. Система должна быть физически, если не математически, не вскрываемой;
2. Нужно, чтобы не требовалось сохранение системы в тайне; попадание системы в руки врага не должно причинять неудобств;
3. Хранение и передача ключа должны быть осуществимы без помощи бумажных записей; корреспонденты должны располагать возможностью менять ключ по своему усмотрению;
4. Система должна быть пригодной для сообщения через телеграф;
5. Система должна быть легко переносимой, работа с ней не должна требовать участия нескольких лиц одновременно;
6. Наконец, от системы требуется, учитывая возможные обстоятельства её применения, чтобы она была проста в использовании, не требовала значительного умственного напряжения или соблюдения большого количества правил.

## Модели криптографии

Криптографическая система представляет собой семейство T преобразований открытого текста. Составные этого семейства индексируются, или обозначаются символом k; параметр k является ключом. Пространство ключей K – это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита.

Криптографические системы делят на:

* симметричные
* ассиметричные

Симметричные криптосистемы (также симметричное шифрование, симметричные шифры) — способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Алгоритм шифрования выбирается сторонами до начала обмена сообщениями.

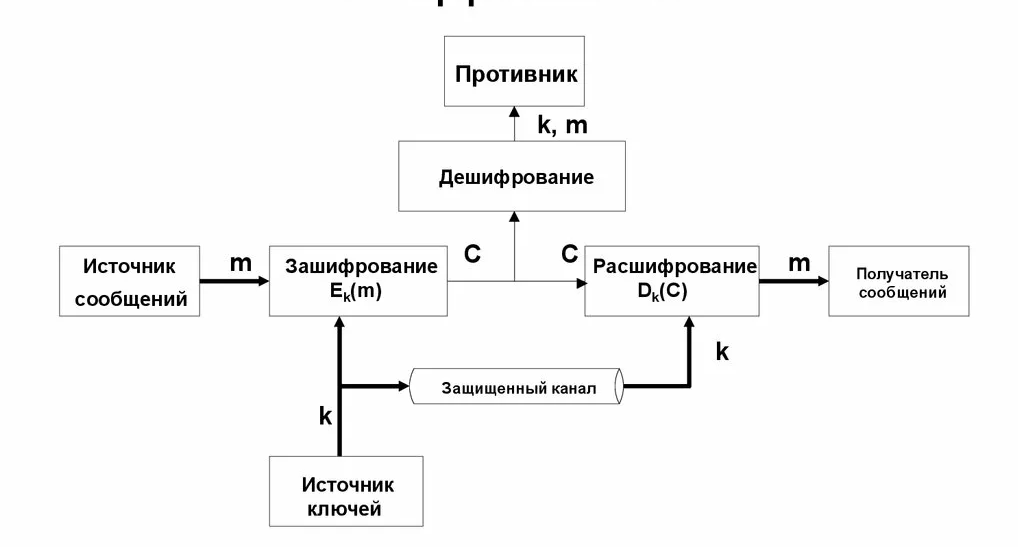


Рисунок 1 – Схема симметричного шифрования

В зависимости от объёма данных, обрабатываемых за одну операцию шифрования, симметричные шифры делятся на блочные и потоковые.

* Блочные шифры – обрабатывают информацию блоками определённой длины (обычно 64, 128 бит), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми раундами. Результатом повторения раундов является лавинный эффект — нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных.
* Поточные шифры – шифры, в которых шифрование проводится над каждым битом либо байтом исходного (открытого) текста.

**Асимметричное шифрование**— это метод шифрования данных, предполагающий использование двух ключей — открытого и закрытого. Открытый (публичный) ключ применяется для шифрования информации и может передаваться по незащищенным каналам. Закрытый (приватный) ключ применяется для расшифровки данных, зашифрованных открытым ключом. Открытый и закрытый ключи — это очень большие числа, связанные друг с другом определенной функцией, но так, что, зная одно, крайне сложно вычислить второе.

Схема передачи данных между двумя субъектами (А и Б) с использованием открытого ключа выглядит следующим образом:

* Субъект А генерирует пару ключей, открытый и закрытый (публичный и приватный).
* Субъект А передает открытый ключ субъекту Б. Передача может осуществляться по незащищенным каналам.
* Субъект Б шифрует пакет данных при помощи полученного открытого ключа и передает его А. Передача может осуществляться по незащищенным каналам.
* Субъект А расшифровывает полученную от Б информацию при помощи секретного, закрытого ключа.

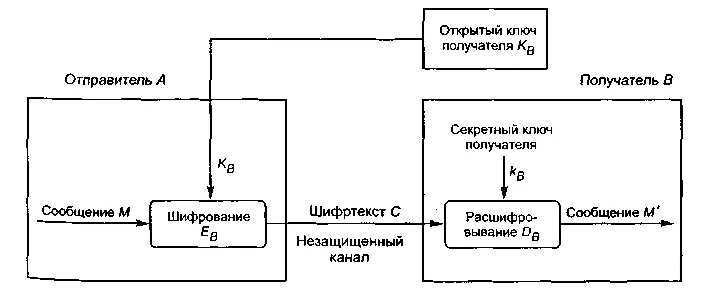


Рисунок 2 – Схема асимметричного шифрования

Так же стоит сказать про **плюсы** и **минусы** моделей:

* Симметричные алгоритмы требуют меньше ресурсов и демонстрируют большую скорость шифрования, чем асимметричные алгоритмы. Большинство симметричных шифров предположительно устойчиво к атакам с помощью квантовых компьютеров, которые в теории представляют угрозу для асимметричных алгоритмов.
* Слабое место симметричного шифрования — обмен ключом. Поскольку для работы алгоритма ключ должен быть и у отправителя, и у получателя сообщения, его необходимо передать; однако при передаче по незащищенным каналам его могут перехватить и использовать посторонние. На практике во многих системах эта проблема решается шифрованием ключа с помощью асимметричного алгоритма.

# АЛГОРИТМИЧЕСКАЯ ТЕОРИЯ СЛОЖНОСТИ

## 2.1. Complexity

Сложность алгоритма помогает оценить затраты на его реализацию и определяется вычислительными мощностями, необходимыми для его выполнения. Сложность алгоритмов оценивают по времени выполнения или по используемой памяти. В обоих случаях сложность зависит от размеров входных данных

Обычно алгоритмы классифицируются в соответствии с их временной сложностью. Можно выделить следующие типы:

* Постоянный –
* Логарифмический –
* Линейный –
* Квадратный –
* Кубический, полиноминальный – ,
* Экспоненциальный – , где t – постоянная, p(n) - функция
* Факториальный –

Существует теория сложности, которая классифицирует не только сложность самих алгоритмов, но и сложность самих задач. Выделяют несколько классов сложности алгоритмов:

Класс P. Задачи, относящиеся к классу сложности P, могут быть решены за полиномиальное время. Класс P является одним из самых узких классов сложности.

Примерами алгоритмов класса P являются стандартные алгоритмы целочисленного сложения, умножения, деления, взятия остатка от деления, перемножения матриц, выяснение связности графов и некоторые другие.

Класс NP. Это класс недетминированно-полиномиальных алгоритмов. Для них невозможно (или пока невозможно) создать реализацию, выполняющуюся за полиномиальное время, но возможно получить некоторое решение недетерминированным способом, а затем проверить его допустимость за полиномиальное время.

В данной работе будут рассмотрены следующие алгоритмы: быстрое возведение в степень; факторизация числа.

## 2.2. Быстрое возведение в степень

**Алгоритм быстрого возведения в степень –** алгоритм, предназначенные для возведения числа x в натуральную степень n за наименьшее число умножений, чем это требуется в определении степени.

В данной работе мною использовался бинарный алгоритм:

Бинарное (двоичное) возведение в степень — это приём, позволяющий возводить любое число в n-ую степень за  умножений (вместо n умножений при обычном подходе).

Для тестирования алгоритма быстрого возведения числа была написана программа PowerAlgorithm.py. Код данной программы можно посмотреть в приложении 1

Результаты выполнения программы:

Для определения корректности вычисления проводятся два испытания (рисунок 3)



Рисунок 3 – Тестирование программы на корректность

Из рисунков видно, что вычисления верные, программа справилась с задачей.

Далее программа с замеров выполнения вычисляет , где a = 1234567; n = [10000, 11000 … , 30000]. По итогу выводится график (рисунок 4):

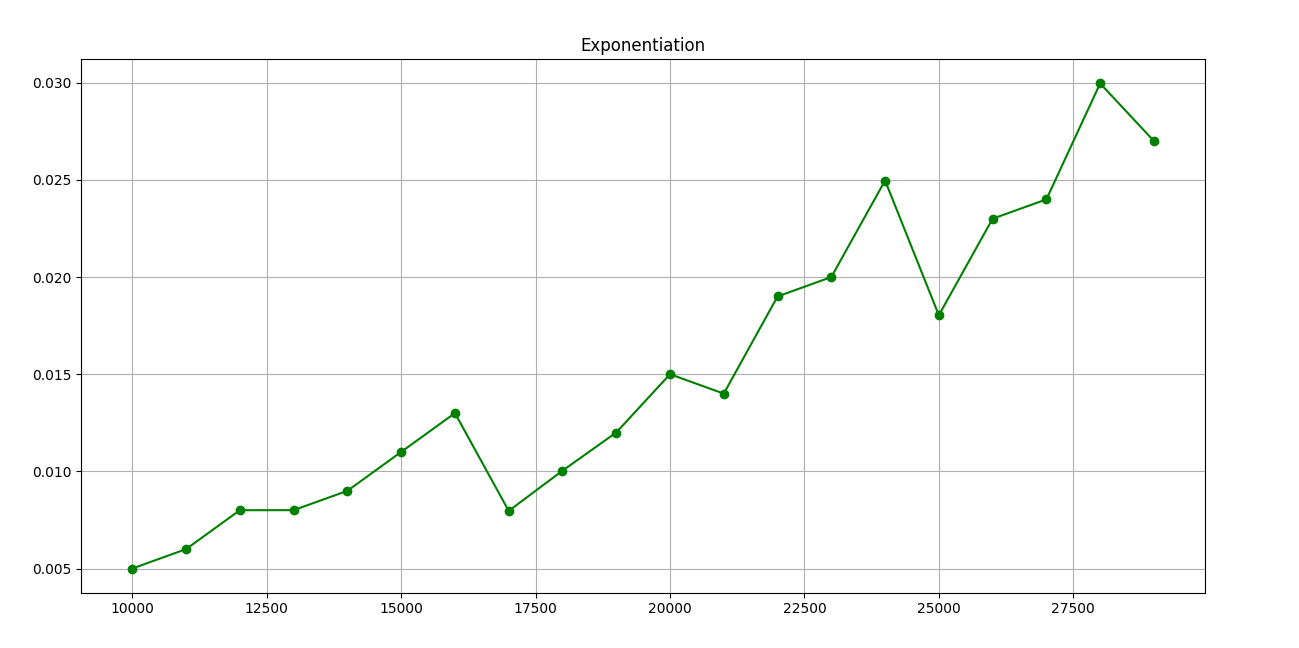


Рисунок 4 – Результат выполнения программы

## 2.3. Факторизация числа

*Факторизацией* называется разложение числа на простые множители. Алгоритм факторизации основывается на тех же идеях, что и алгоритм проверки на простоту, а именно: если у числа существует простой делитель, отличный от него самого, то он не превышает корня из числа. Для факторизации числа нужно перебрать все числа в промежутке , и попытаться разделить  на каждое из них по очереди.

Так же стоит сказать, что если при факторизации числа x был найден делитель y, то останется факторизовать число x/y. Поэтому можно смело разделить число x на y и продолжить работу алгоритма. Можно не начинать проверку сначала, так как число x/y гарантированно не имеет делителей меньше y, иначе они были бы найдены при факторизации x.

Также очевидно, что все множители, найденные алгоритмом, будут простыми. Можно заметить, что каждый раз алгоритм находит минимальный из всех делителей числа, и делит на него само число. Минимальный возможный делитель числа всегда будет простым**.**

Для факторизации числа была написана программа Factorization.py. Полный код программы можно найти в приложении 2.

Данное ПО находит все простые делители нескольких чисел. Данные числа получаются путём добавления в конец предыдущего числа цифры от 0 до 9.

На рисунке 5 представлен вывод программы:

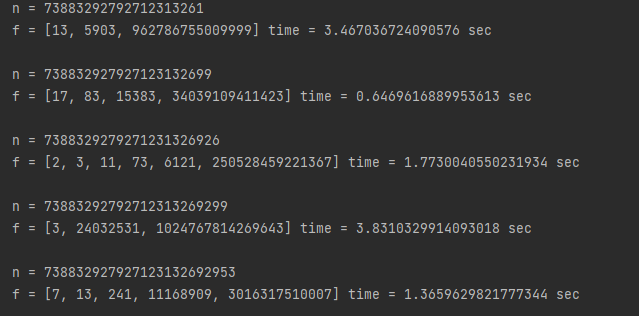


Рисунок 5 – Результат выполнения программы

На рисунке 6 представлен график:

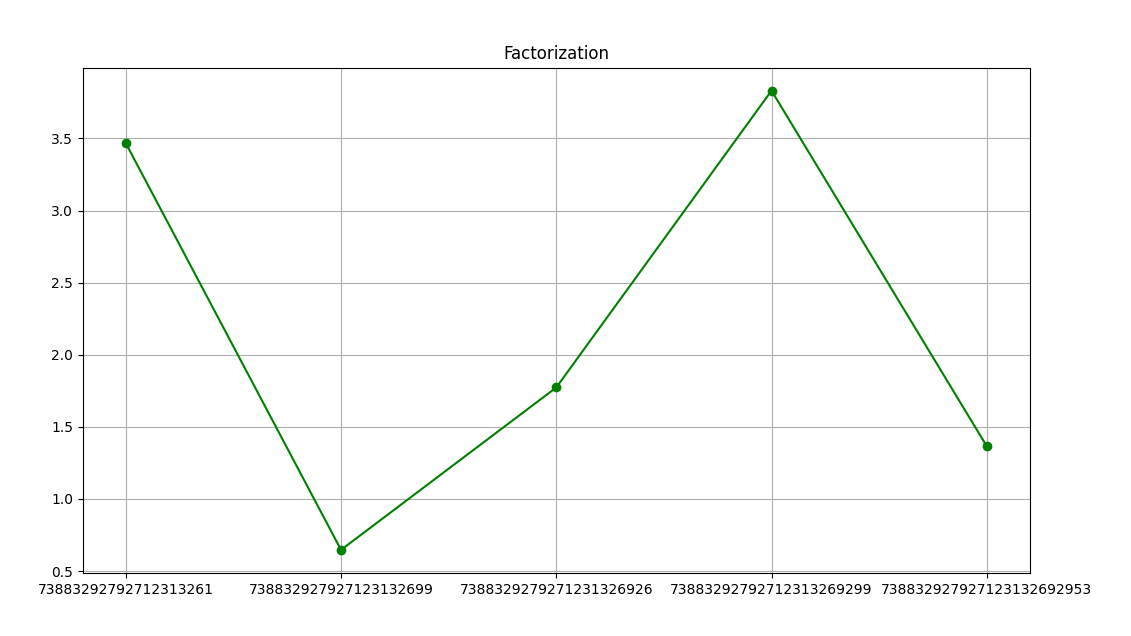


Рисунок 6 – График с результатами

# КРИПТОАНАЛИЗ

## 3.1. Взлом подстановки методом частотного анализа

**Исходный текст 4 вариант:**

ЩДГЕФШСЫЙЯСКХЖЩЯСЛДЕГЬЮЬЯЬКСЛДЕГЬЫЬЕЬАСФСЬЫЯТАТЯС ЬСКЯЬЮТХСКЬЮЕДХДГГФВЬЮСЬЫТУЭДЯОГЬПС

ЮГДИГЬКЭОРСГТЕУАСДЩЬСЫЙЯСГТСЕДАВЬКЭОСЫЬЩТЭСТЭЯТКГЙПСМДЯДГЙПСВТЦЭТГСКСЫДЯЙХФС ЕЬЕДМУХФСЫДЕДЭСКСЫД

ЯЙХС ДЕЬХСЫЯФКЭТЮИФПСБДХЛЖЩЬХСФСТЯХТМТХФСИТЩТСЕТ

ЫЬЭЙСЫДГЮДГЖЭЬ

Для начала с помощью программы из лабораторной работы №1 по КМСЗИ (рисунок 7). Данное ПО необходимо для автоматизации процесса подсчёта частоты.

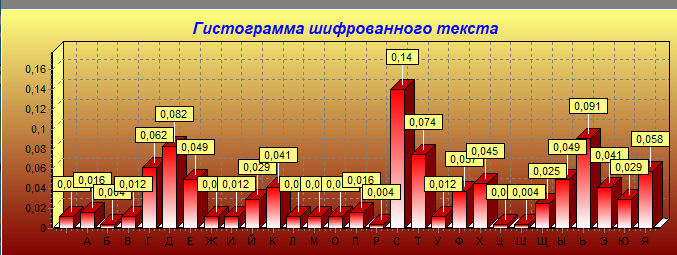


Рисунок 7 – Гистограмма зашифрованного текста

Предполагаем что «С» пробел, так как это самый частый символ. Тогда:

ЩДГЕФШ

ЫЙЯ

КХЖЩЯ

ЛДЕГЬЮЬЯЬК

ЛДЕГЬЫЬЕЬА

Ф

ЬЫЯТАТЯ

Ь

КЯЬЮТХ

КЬЮЕДХДГГФВЬЮ

ЬЫТУЭДЯОГЬП

ЮГДИГЬКЭОР

ГТЕУА

ДЩЬ

ЫЙЯ

ГТ

ЕДАВЬКЭО

ЫЬЩТЭ

ТЭЯТКГЙП

МДЯДГЙП

ВТЦЭТГ

К

ЫДЯЙХФ

ЕЬЕДМУХФ

ЫДЕДЭ

К

ЫДЯЙХ

ДЕЬХ

ЫЯФКЭТЮИФП

БДХЛЖЩЬХ

Ф

ТЯХТМТХФ

ИТЩТ

ЕТЫЬЭЙ

ЫДГЮДГЖЭЬ

Можно заметить 2 подряд стоящие ГГ. Самая часто употребляющая биграмма в середине слова – НН.

«Г» = «Н»

Предполагаем что «Ь» -> «О», так как это следующий самый употребляемый символ.

Заметим эти два слова (прописные символы = зашифрованные символы):

ЛДЕноЮоЯоК

ЛДЕноЫоЕоА

Они однокоренные. А значит близкие по смыслу. Отличаются только концовкой. А значит это два сложных слова с одинаковым началом. Так же стоит предположить, что эти слова являются описанием чего-то(кого-то). После этих слов идёт «Ф» и скорее всего «Ф» -> «И».

Поработаем ещё над этими словами:

ЛДЕноЮоЯоК

ЛДЕноЫоЕоА

Символ «Е» до шифрования являлся согласным. Так же данный символ часто употребляется. Опытным путём был найден исходный символ «Е» -> «Р».

И данные слова:

ЧЕРНОВОЛОС

ЧЕРНОБОРОД

Далее с помощью сервиса по помощи в сканвордах и уже известным буквам были найдены все остальные слова.

ГЕНРИХ

БЫЛ

СМУГЛ

ЧЕРНОВОЛОС

ЧЕРНОБОРОД

И

ОБЛАДАЛ

ПО

СЛОВАМ

СОВРЕМЕННИКОВ

ОБАЯТЕЛЬНОЙ

ВНЕШНОСТЬЮ

НАРЯД

ЕГО

БЫЛ

НА

РЕДКОСТЬ

БОГАТ

АТЛАСНЫЙ

ЗЕЛЕНЫЙ

КАФТАН

С

БЕЛЫМИ

ПРОРЕЗЯМИ

БЕРЕТ

С

БЕЛЫМ

ПЕРОМ

БЛИСТАВШИЙ

ЖЕМЧУГОМ

И

АЛМАЗАМИ

ШПАГА

РАБОТЫ

БЕНВЕНУТО

Все замены букв представлены в таблице 1.2 и 1.2 (продолжение):

Таблица 1.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Нормативный  алфавит | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| Числовые  эквиваленты | 1  Д | 2  Ж | 3  К | 4  Н | 5  Е | 6  Р | 7  У | 8 | 9  Ш | 10  Ы | 11  С | 12  Ч | 13  З | 14 | 15  Ь | 16  Й |

Таблица 1.2 (продолжение).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Нормативный  алфавит | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | “\_“ |
| Числовые  эквиваленты | 17  Ю | 18 | 19  А | 20  Я | 21  И | 22  М | 23  Ы | 24 | 25  Х | 26  Г | 27 | 28  Б | 29  О | 30  Т | 31  В | 32  Л | 33  П |

В итоге был получен данный текст:

ГЕНРИХ БЫЛ СМУГЛ ЧЕРНОВОЛОС ЧЕРНОБОРОД И ОБЛАДАЛ ПО СЛОВАМ СОВРЕМЕННИКОВ ОБАЯТЕЛЬНОЙ ВНЕШНОСТЬЮ НАРЯД ЕГО БЫЛ НА РЕДКОСТЬ БОГАТ АТЛАСНЫЙ ЗЕЛЕНЫЙ КАФТАН С БЕЛЫМИ ПРОРЕЗЯМИ БЕРЕТ С БЕЛЫМ ПЕРОМ БЛИСТАВШИЙ ЖЕМЧУГОМ И АЛМАЗАМИ ШПАГА РАБОТЫ БЕНВЕНУТО

С помощью интернета был найден текст, откуда был взят данный фрагмент: **Две Дианы.** Исторический роман Александра Дюма

## 3.2. Атака Фридмана на гаммирование

**Исходный текст 4 вариант:**

тфщгщпьзершгждяыэйбэшояфазпжсяэнтйрабьпъпухзхлотйббкмгрффйфиъпжпббпфпушъзтмпщзисыэшэрюьзафюахгспрбмспэщраоьъумпьзисфофпртымтвюощпфюохоспючйббптэююоцпфынлжьрофбрьэфунупжйрхфжуцььбжрыбюйббймачшсбжруъзиспазлряяхтщоъпаырщжвгфахаяэошлсчптацщоюуябэжоцюрблююсмоююоьпаюжмжрээжпюпцфбьпещпрхтхафврба\_\_эпиюх\_тйрычяэряяхтвюфжрашсмуббспжряэщпэпспеппещпрффйфиъпжлсъообуюяхзцэьз\_раыхуашажоспчопт\_пнжюшфжтсыпубюфябаюрочфырсмацуэжугцбчлщпчжоспуптып\_жхцэчстяьоцсщъэтпвкшжлруяъещпэфашртхгя\_чт

**Для начала найдём длину ключа.**

Для этого найдём схожие символы в тексте, в особенности биграммы. Это:

бб – отличия между позициями данной биграммы: [15;75;45;135]

фф – отличия между позициями данной биграммы: [285]

юю – отличия между позициями данной биграммы: [105;5]

Общий делитель для всех этих чисел = 5.

Значит длина ключа = 5.

Разделим символы по фрагментам по 5 символов. Для автоматизации процесса была использована собственная программа, о которой в конце.

Первые фрагменты по 5 символов и последние:

тфщгщ

пьзер

шгждя

ыэйбэ

шояфа

…

…

пэфаш

ртхгя

\_чт

Частота (рисунок 8):

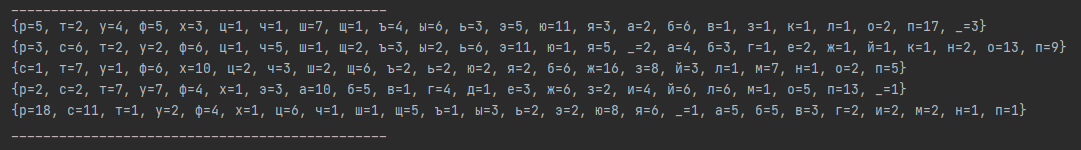


Рисунок 8 – Частота символов

**Группа символов** – совокупность символов, у которых один и тот же ключ при шифровании.

1 группа символов (1 строчка на рисунке 8): Делаем предположение что «п» = «\_», тогда ключ для данной группы= «п» => «ю» = «о» «ш» = «и»…это популярные буквы, проверим на практике.

Тфщгщ = В????

Пьзер = \_?????

Шгждя = И????

Ыэйбэ = Л????

2 группа символов (2 строчка на рисунке 8): Делаем предположение что «о» = «\_», тогда ключ для данной группы= «о» => «э» = «о» «п» = «а»…это популярные буквы, проверим на практике.

Тфщгщ = ВЕ???

Пьзер = \_Н???

Шгждя = ИХ???

Ыэйбэ = ЛО???

3 группа символов (3 строчка на рисунке 8): Делаем предположение что «ж» = «\_», тогда ключ для данной группы= «ж» => «х» = «о» «з» = «а» «т» = «л» «м» = «е» …это популярные буквы, проверим на практике.

Тфщгщ = ВЕТ??

Пьзер = \_НА??

Шгждя = ИХ\_??

Ыэйбэ = ЛОВ??

4 группа символов (4 строчка на рисунке 8): Делаем предположение что «п» = «\_», тогда ключ для данной группы= «п» => «а» = «с» «т» = «в» «у» = «г» «ж» = «е», проверим на практике.

К сожалению, второе слово \_НАЦ??ИХ\_ странное, предположим, что «а» = «\_» => ключ для данной группы= «а».

Тфщгщ = ВЕТВ?

Пьзер = \_НАД?

Шгждя = ИХ\_Г?

Ыэйбэ = ЛОВА?

5 группа символов (5 строчка на рисунке 8): Делаем предположение что «р» = «\_», тогда ключ для данной группы= «р» => «с» = «а» «щ» = «и» «у» = «г» «ж» = «е» …это популярные буквы, проверим на практике.

Тфщгщ = ВЕТВИ

Пьзер = \_НАД\_

Шгждя = ИХ\_ГО

Ыэйбэ = ЛОВАМ

**КЛЮЧ = ПОЖАР**

Расшифруем последний фрагмент:

Пэфаш

ртхгя

\_чт

Расшифровываем \_он\_заговорил.

**Так как известно произведение и автор оригинала, то найдём начало и конец расшифрованного текста и получим:**

ветви над их головами шурша роняли сухой дождь листьев девушка остановилась казалось она готова была отпрянуть назад но вместо того она пристально поглядела на монтэга и ее темные лучистые живые глаза так просияли как будто он сказал ей чтото необыкновенно хорошее но он знал что его губы произнесли лишь простое приветствие потом видя что девушка как завороженная смотрит на изображение саламандры на рукаве его тужурки и на диск с фениксом приколотый к груди он заговорил

**Оригинал же текста со знаками препинания выглядит так:**

Ветви над их головами, шурша, роняли сухой дождь листьев. Девушка остановилась. Казалось, она готова была отпрянуть назад, но вместо того она пристально поглядела на Монтэга, и ее темные, лучистые, живые глаза так просияли, как будто он сказал ей что-то необыкновенно хорошее. Но он знал, что его губы произнесли лишь простое приветствие. Потом, видя, что девушка как завороженная смотрит на изображение саламандры на рукаве его тужурки и на диск с фениксом, приколотый к груди, он заговорил.

**В процессе работы** для автоматизации процесса я пользовался своей программой, код которой представлен в приложение 3. Данная программа получает на вход строку, разбивает её на фрагменты по 5 символов, и считает частоту символов для всей строки и для символов каждой группы. Полный вывод программы представлен в приложение 4.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. В. В. Ященко Введение в криптографию – МЦНМО, 2012 -с. 348 URL: https://www.rulit.me/data/programs/resources/pdf/Vvedenie-v-kriptografiyu\_RuLit\_Me\_661516.pdf (дата обращения: 11.12.2021).
2. Владимиров С. М. Криптографические методы защиты информации – 2019 г.– URL: https://www.rulit.me/data/programs/resources/pdf/Vladimirov\_Kriptograficheskie-metody-zashchity-informacii\_RuLit\_Me\_670857.pdf (дата обращения 11.12.2021)
3. Н. Н. Токарева Симметричная криптография. Краткий курс. Учебное пособие. – М.: Новосибирский государственный университет, 2012 -с. 228 — URL: http://www.math.nsc.ru/~tokareva/lib/crypto-online.pdf (дата обращения: 11.12.2021).
4. kodek16 Разложение числа на простые множители (факторизация). // brestprog [Электронный ресурс] – URL: https://brestprog.by/topics/factorization/ (дата обращения 11.12.2021)

# ПРИЛОЖЕНИЕ 1

import time as t

import matplotlib.pyplot as plt

def power(a, n):

return (1 if n == 0

else power(a \* a, n // 2) if n % 2 == 0

else a \* power(a, n - 1))

arg\_n = []

n = 10000

for i in range(20):

arg\_n.append(n)

n+=1000

print("2^10 = ", power(2, 10))

print("2^32 = ", power(2, 32))

c\_x = []

c\_y = []

for i in range(len(arg\_n)):

a = 1234567

n = arg\_n[i]

c\_x.append(n)

start\_time = t.time()

power(a, n)

res = t.time() - start\_time

c\_y.append(res)

plt.plot(arg\_n, c\_y, 'g-o')

plt.title('Exponentiation')

plt.grid()

plt.show()

# ПРИЛОЖЕНИЕ 2

import random

import time as t

import matplotlib.pyplot as plt

def factorize(n):

Ans = []

d = 2

while d \* d <= n:

if n % d == 0:

Ans.append(d)

n //= d

else:

d += 1

if n > 1:

Ans.append(n)

return Ans

c\_x = []

c\_y = []

test\_n = []

n = 738832927927123132

for i in range(5):

n = int(str(n) + str(random.randint(0, 9)))

test\_n.append(n)

for i in range(5):

n = int(str(test\_n[i])+str(random.randint(0, 9)))

print(f'n = {n}')

c\_x.append(str(n))

start = t.time()

fact = factorize(n)

end = t.time() - start

c\_y.append(end)

print(f'f = {fact} time = {end} sec\n')

plt.plot(c\_x, c\_y, 'g-o')

plt.title('Factorization')

plt.grid()

plt.show()

# ПРИЛОЖЕНИЕ 3

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Locale;

public class Program {

public static void main(String[] args){

int n = 0;

String string = StdIn.readLine();

/\* Если необходимо удалить знаки препинания

String Rstring = string.replaceAll("[^A-Za-zА-Яа-я0-9 ]", "").toLowerCase(Locale.ROOT);

System.out.println(Rstring);

//System.exit(-1);

\*/

List<String> list = new ArrayList<>();

char[] charArray = string.toCharArray();

System.out.println("Dlina "+(charArray.length-1));

String newStr = "";

System.out.println("Kajday 5 stroka ");

for(int i = 1; i <= charArray.length; i++){

newStr+=charArray[i];

if((i%5 == 0 || i == charArray.length-1) && i != 0){

list.add(newStr);

newStr = "";

}

}

for(String s : list){

System.out.println(s);

}

System.out.println("Частота всего");

HashMap<Character, Integer> chastota = new HashMap<>();

for(int i = 1; i <= charArray.length;i++){

if(chastota.get(charArray[i]) == null){

chastota.put(charArray[i], 1);

}

else chastota.put(charArray[i], chastota.get(charArray[i])+1);

}

System.out.println(chastota.toString());

HashMap<Character, Integer> chastota1 = new HashMap<>();

HashMap<Character, Integer> chastota2 = new HashMap<>();

HashMap<Character, Integer> chastota3 = new HashMap<>();

HashMap<Character, Integer> chastota4 = new HashMap<>();

HashMap<Character, Integer> chastota5 = new HashMap<>();

for(String s : list) {

char[] charArray\_ = s.toCharArray();

for (int i = 0; i < charArray\_.length; i++) {

switch (i){

case 0:

if (chastota1.get(charArray\_[i]) == null) {

chastota1.put(charArray\_[i], 1);

} else chastota1.put(charArray\_[i], chastota1.get(charArray\_[i]) + 1);

break;

case 1:

if (chastota2.get(charArray\_[i]) == null) {

chastota2.put(charArray\_[i], 1);

} else chastota2.put(charArray\_[i], chastota2.get(charArray\_[i]) + 1);

break;

case 2:

if (chastota3.get(charArray\_[i]) == null) {

chastota3.put(charArray\_[i], 1);

} else chastota3.put(charArray\_[i], chastota3.get(charArray\_[i]) + 1);

break;

case 3:

if (chastota4.get(charArray\_[i]) == null) {

chastota4.put(charArray\_[i], 1);

} else chastota4.put(charArray\_[i], chastota4.get(charArray\_[i]) + 1);

break;

case 4:

if (chastota5.get(charArray\_[i]) == null) {

chastota5.put(charArray\_[i], 1);

} else chastota5.put(charArray\_[i], chastota5.get(charArray\_[i]) + 1);

break;

}

}

}

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println(chastota1.toString());

System.out.println(chastota2.toString());

System.out.println(chastota3.toString());

System.out.println(chastota4.toString());

System.out.println(chastota5.toString());

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

}

# ПРИЛОЖЕНИЕ 4

Входные данные:

тфщгщпьзершгждяыэйбэшояфазпжсяэнтйрабьпъпухзхлотйббкмгрффйфиъпжпббпфпушъзтмпщзисыэшэрюьзафюахгспрбмспэщраоьъумпьзисфофпртымтвюощпфюохоспючйббптэююоцпфынлжьрофбрьэфунупжйрхфжуцььбжрыбюйббймачшсбжруъзиспазлряяхтщоъпаырщжвгфахаяэошлсчптацщоюуябэжоцюрблююсмоююоьпаюжмжрээжпюпцфбьпещпрхтхафврба\_\_эпиюх\_тйрычяэряяхтвюфжрашсмуббспжряэщпэпспеппещпрффйфиъпжлсъообуюяхзцэьз\_раыхуашажоспчопт\_пнжюшфжтсыпубюфябаюрочфырсмацуэжугцбчлщпчжоспуптып\_жхцэчстяьоцсщъэтпвкшжлруяъещпэфашртхгя\_чт

Выходные данные:

Dlina 473

Kajday 5 stroka

тфщгщ

пьзер

шгждя

ыэйбэ

шояфа

зпжся

энтйр

абьпъ

пухзх

лотйб

бкмгр

ффйфи

ъпжпб

бпфпу

шъзтм

пщзис

ыэшэр

юьзаф

юахгс

прбмс

пэщра

оьъум

пьзис

фофпр

тымтв

юощпф

юохос

пючйб

бптэю

юоцпф

ынлжь

рофбр

ьэфун

упжйр

хфжуц

ььбжр

ыбюйб

бймач

шсбжр

уъзис

пазлр

яяхтщ

оъпаы

рщжвг

фахая

эошлс

чптац

щоюуя

бэжоц

юрблю

юсмою

юоьпа

южмжр

ээжпю

пцфбь

пещпр

хтхаф

врба\_

\_эпию

х\_тйр

ычяэр

яяхтв

юфжра

шсмуб

бспжр

яэщпэ

пспеп

пещпр

ффйфи

ъпжлс

ъообу

юяхзц

эьз\_р

аыхуа

шажос

пчопт

\_пнжю

шфжтс

ыпубю

фябаю

рочфы

рсмац

уэжуг

цбчлщ

пчжос

пупты

п\_жхц

эчстя

ьоцсщ

ъэтпв

кшжлр

уяъещ

пэфаш

ртхгя

\_чт

Частота всего

{р=28, с=20, т=19, у=16, ф=25, х=15, ц=10, ч=10, ш=11, щ=14, ъ=10, ы=11, ь=13, э=21, ю=22, я=16, \_=7, а=21, б=25, в=5, г=7, д=1, е=5, ж=23, з=11, и=6, й=10, к=2, л=8, м=10, н=4, о=22, п=45}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{р=5, т=2, у=4, ф=5, х=3, ц=1, ч=1, ш=7, щ=1, ъ=4, ы=6, ь=3, э=5, ю=11, я=3, а=2, б=6, в=1, з=1, к=1, л=1, о=2, п=17, \_=3}

{р=3, с=6, т=2, у=2, ф=6, ц=1, ч=5, ш=1, щ=2, ъ=3, ы=2, ь=6, э=11, ю=1, я=5, \_=2, а=4, б=3, г=1, е=2, ж=1, й=1, к=1, н=2, о=13, п=9}

{с=1, т=7, у=1, ф=6, х=10, ц=2, ч=3, ш=2, щ=6, ъ=2, ь=2, ю=2, я=2, б=6, ж=16, з=8, й=3, л=1, м=7, н=1, о=2, п=5}

{р=2, с=2, т=7, у=7, ф=4, х=1, э=3, а=10, б=5, в=1, г=4, д=1, е=3, ж=6, з=2, и=4, й=6, л=6, м=1, о=5, п=13, \_=1}

{р=18, с=11, т=1, у=2, ф=4, х=1, ц=6, ч=1, ш=1, щ=5, ъ=1, ы=3, ь=2, э=2, ю=8, я=6, \_=1, а=5, б=5, в=3, г=2, и=2, м=2, н=1, п=1}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_