Правительство Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н. Тихонова

ОТЧЕТ

О ПРАКТИЧЕСКОЙ РАБОТЕ №1

по дисциплине «Проектный семинар по информационной безопасности»

ШИФРЫ ГАММИРОВАНИЯ

|  |
| --- |
| Студент гр. БИБ211  \_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Семененя  «21» ноября 2021 г. |
| Руководитель  Заведующий кафедрой информационной безопасности киберфизических систем  канд. техн. наук, доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_О.О. Евсютин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва 2021

**СОДЕРЖАНИЕ**

[1 Задание на практическую работу 3](#_Toc88425895)

[2 Краткая теоретическая часть 4](#_Toc88425896)

[2.1 Описание шифров 4](#_Toc88425897)

[2.2 Методы криптоанализа шифров 4](#_Toc88425898)

[3 Примеры шифрования 5](#_Toc88425899)

[4 Программная реализация шифров 9](#_Toc88425900)

[5 Примеры криптоанализа 12](#_Toc88425901)

6 Выводы о проделанной работе

[7 Список использованных источников 19](#_Toc88425902)

# 1 Задание на практическую работу

Целью данной работы является приобретение навыков программной реализации и криптоанализа применительно к шифрам гаммирования.

В рамках практической работы необходимо выполнить следующее:

1) написать программную реализацию следующих шифра Виженера с тремя

способами выработки гаммы на основе секретного ключа шифрования:

- повторение короткого лозунга;

- самоключ Виженера по открытому тексту;

- самоключ Виженера по шифртексту.

2) изучить методы криптоанализа шифров гаммирования с использованием

дополнительных источников;

3) провести криптоанализ данных шифров;

4) подготовить отчет о выполнении работы.

# 2 Краткая теоретическая часть

## 2.1 Описание шифров

В работе используется Шифр Виженера – метод полиалфавитного шифрования буквенного текста. Он основан на многоалфавитной замене и использует ключевое слово. Несмотря на простоту в реализации, три века он не поддавался криптоанализу, чем заслужил звание «неразгаданного».[1]

­Найти ключевое слово можно тремя способами:

1. взять короткий лозунг, повторяющийся на протяжении всего текста;
   1. Вводится ключ длиной меньше, чем открытый текст;
   2. Он циклично повторяется на протяжении всего текста.
2. использовать самоключ по открытому тексту:
   1. Ключ случайным образом выбирается из алфавита;
   2. К нему добавляется гамма: весь открытый текст кроме последней буквы.
3. использовать самоключ по шифртексту:
   1. Ключ случайным образом выбирается из алфавита;
   2. Складывается первая пара номеров букв «ключ-открытый текст»;
   3. Результат сложения дописывают в гамму, и номер следующей буквы открытого текста складывают уже с ним.

Далее устанавливается соответствие между буквами открытого текста и гаммы так, что для текста из n символов создаётся n пар «буква текст­а – буква гаммы». Далее для шифрования каждой букве присваивают их номера в алфавите, и по каждой паре происходит сложение этих номеров. Полученные суммы делят на количество букв в алфавите и берут остаток, чтобы не получилось некорректного значения. Затем числа переводят обратно в буквы, опираясь на их номера в алфавите. [2]

Для расшифрования описанная выше процедура претерпевает небольшое изменение – вместо сложения номер буквы ключевого слова отнимают от номера буквы шифртекста.

## 2.2 Методы криптоанализа шифров

Для криптоанализа шифра Виженера зачастую используется метод Касиски. Он основывается на предположении, что повторяющиеся биграммы и триграммы шифртекста указывают на одинаковые части открытого текста (при этом ищутся лишь группы из трёх и большего количества букв).[3] Расстояние между ними может быть кратным длине ключевого слова, обозначаемую n (предполагается, что это будет НОД всех расстояний). Как только криптоаналитик её находит, он может записать шифртекст в n колонках, и каждую из них подвергнуть частотному анализу, как зашифрованный моноалфавитным шифром текст.[4]

# 3 Примеры шифрования

В примерах будем использовать заглавные буквы латинского алфавита.

В качестве открытого текста возьмём фразу «**FLOWERS FOR ALGERNON**», а ключа – повторяющийся лозунг «**KEY**».  
 Для начала нам нужно подставить к каждому символу исходного текста соответствующую букву из ключа:

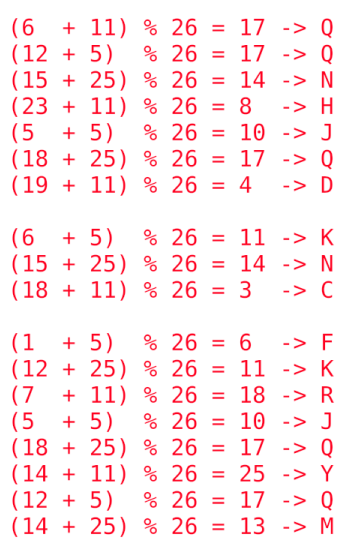


Рисунок

Затем каждую букву как из открытого послания, так и из ключа, нужно представить как её номер в алфавите. 

Рисунок

Теперь мы можем сложить номера находящихся на одной позиции букв послания и ключа: F с K, L с E и так далее, чтобы затем снова найти буквы по ним:



Рисунок

Мы получили результат «**QQNHJQD KNC FKRJQYQM**».

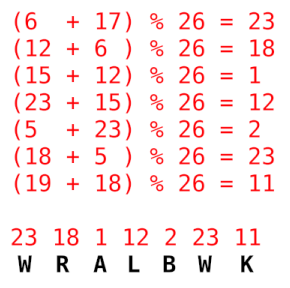
Теперь попробуем зашифровать текст «**FLOWERS**» по самоключу Виженера.

В качестве ключа возьмём букву **Q**. Добавим к ней остальные символы гаммы – то есть все буквы открытого текста, кроме последней:



Рисунок

Производим сложение, переводим цифры в буквы.



Рисунок

Получаем результат **«WRALBWK»**.

Рассмотрим способ шифрования с помощью самоключа по шифртексту. В качестве открытого текста возьмём слово **«FLOWERS»**, а первую букву выберем случайно – пусть, как и в предыдущем примере, это будет **Q**.



Рисунок

Произведём сложение первой пары номеров букв, а именно 6 (F) и 17 (Q). Получаем 23, под этим номером стоит буква W:



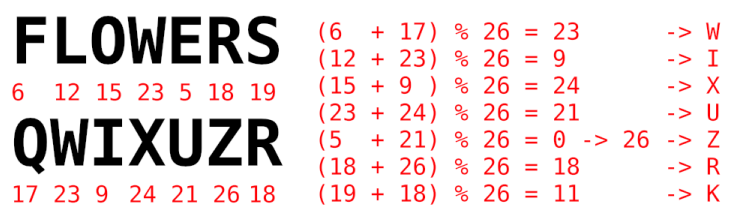
Рисунок

Пополняем гамму полученным символом:



Рисунок

Складываем на этот раз 12 (L) и 23 (W), результат так же дописываем в гамму. После повторения этих действий до получения результата шифрования мы получаем слово **WIXUZRK**.



Рисунок

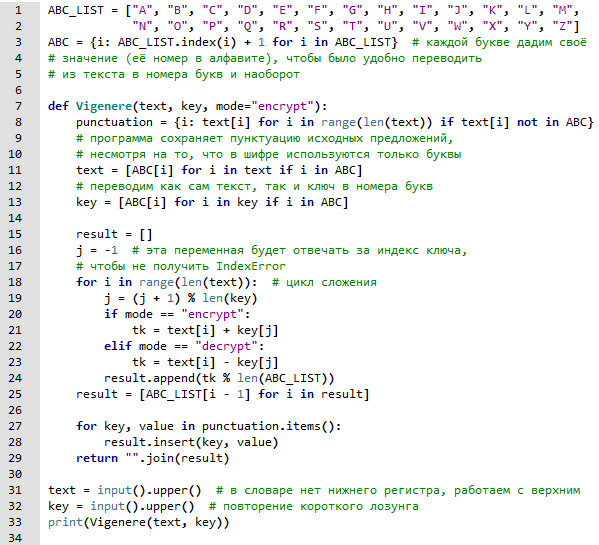
# 4 Программная реализация шифров

Программная реализация шифра Виженера написана на языке программирования Python 3.

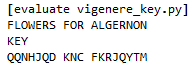
Для того, чтобы избежать повторяющегося кода, шифрование и расшифрование находятся в одной функции, и указать, что именно требуется от программы, можно с помощью параметров функции. По умолчанию стоит шифрование.

Код и примеры работы программы:

1. Для короткого лозунга:

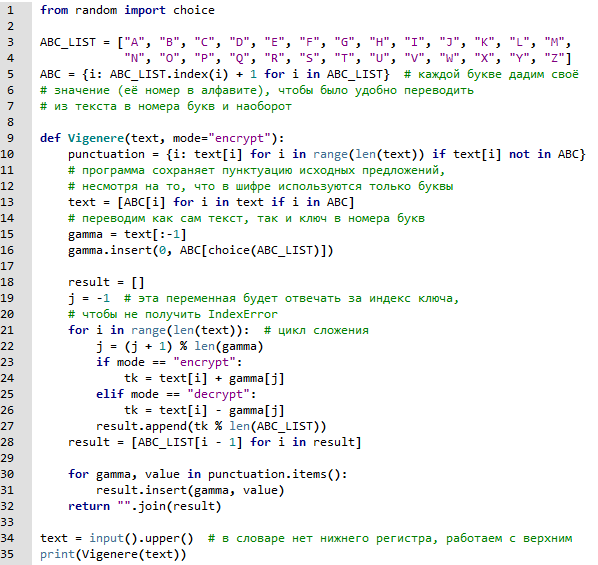


Рисунок



Рисунок

1. Для самоключа по открытому тексту:

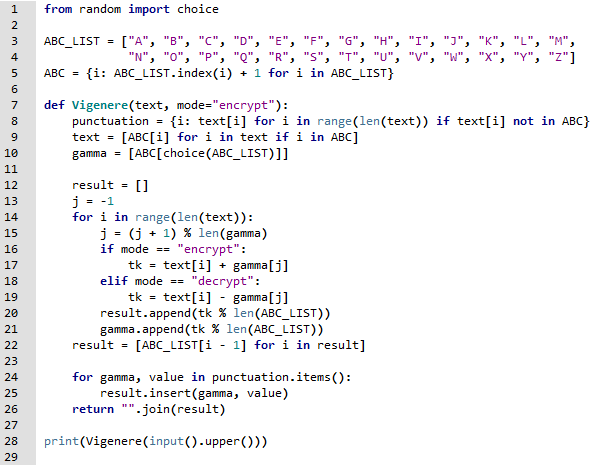


Рисунок



Рисунок

1. Для самоключа по шифртексту:



Рисунок



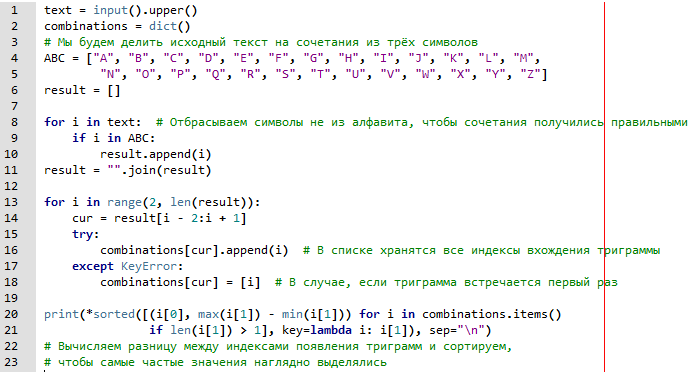
Рисунок

Можно сравнить результаты с полученными от шифрования «вручную» и убедиться, что они совпадают.

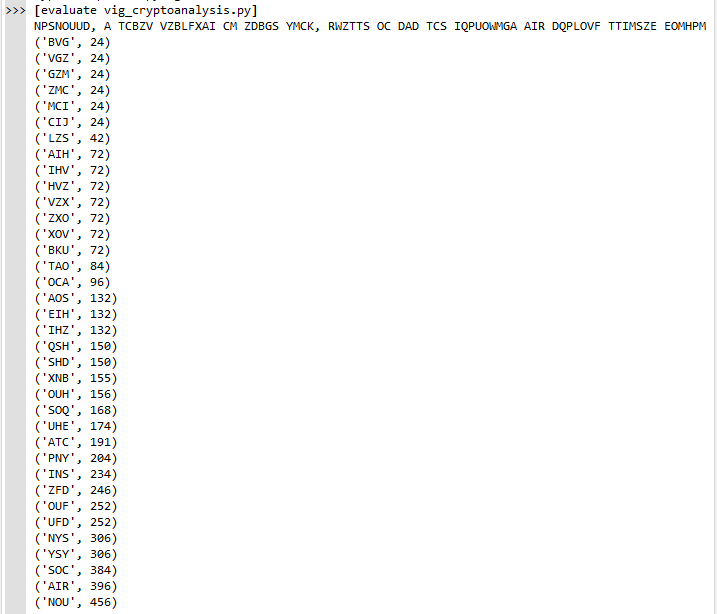
# 5 Примеры криптоанализа

В качестве шифртекста возьмём «**NPSNOUUD, A TCBZV VZBLFXAI CM ZDBGS YMCK, RWZTTS OC DAD TCS IQPUOWMGA AIR DQPLOVF TTIMSZE EOMHPM DF WSSYDNO. VHHXNB GXGPNYSYQS HDG LEIAOS, OQ CEZRZ 3,000 PJCVHZ FD SPPZUSINS OUH ESDLZSIOIYQH AN O ZGXTJF. IMHSVBPA PPKFVMRHZG OUH FMWLZS AIHVZXO, V KLMATCM TQGCCOUF DF QSUURE, RVV TPS KFLHXOPGSK PNY FLBTAOSKXN BVWSQS HDA VGI. AIHVZXO VUYQTS, WIA EXNXS OQ XS XOZT-EOJF – OUH SCWWE PNY ALDRHVBKUHE VFL NJST OA ETA OC ADXPJZPE, IHZ WUPXEN, ALJXCJ OUP TNBZHZS – HZ DYABINSZ FD CJJLD P BJBK UU BVGZMCIJ QHZ UIIR H XTNYSY, ED BVGZMCIJ HBDCS OC ATT JZKPEW MJBLKAEIRLD HHTZVOZ AIR UMBEN OUFDNDC HE IHZ ZVMC'S BIHDPNOCY**».

Для успешного дешифрования по методу Касиски сначала стоит найти в нём повторяющиеся триграммы (сочетания из трёх повторяющихся символов). Чтобы не пропустить никакой из них, напишем программу на Python 3:



Рисунок



Рисунок

Мы видим, что чаще других результатов с заметным отрывом встречаются расстояния 72 и 24: семь и шесть раз, соответственно. 72 нацело делится на 24, значит, можно предположить, что эти расстояния кратны длине ключа.

Все делители числа **24**, не считая 1: 2, 3, 4, 6, 8, 12, 24  
Все делители числа **72**, не считая 1: 2, 3, 4, 6, 8, 9, 12, 18, 24, 36, 72

Зелёным цветом отмечены общие для обоих чисел. Одно из них, возможно, будет длиной ключевого слова. Так как значений слишком много, чтобы пробовать подставлять каждое из них, посмотрим на другие комбинации.

Три раза встречается **132**, взглянем на его делители (без учёта 1): 2, 3, 4, 6, 11, 12, 22, 33, 44, 66, 132.

Дважды видим **150**, его делители (без учёта 1): 2, 3, 5, 6, 10, 15, 25, 30, 50, 75, 150.

Также дважды видим **252**, делители этого числа: 2, 3, 4, 6, 7, 9, 12, 14, 18, 21, 28, 36, 42, 63, 84, 126, 252.

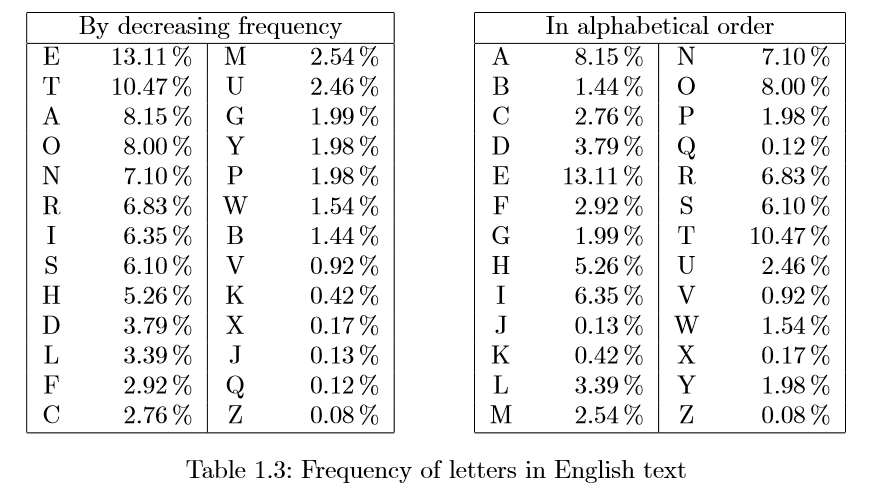
Для **306**, встречающегося дважды: 2, 3, 6, 9, 17, 18, 34, 51, 102, 153, 306.

Предполагаемая длина ключевого слова равна НОД всех расстояний. Для 132, 150, 252 и 306 это будет 6. Предположим, что это так, и разобьём текст на колонки по 6 символов каждая:

**NPSNOU UDATCB ZVVZBL FXAICM ZDBGSY MCKRWZ TTSOCD ADTCSI QPUOWM GAAIRD QPLOVF TTIMSZ EEOMHP MDFWSS YDNOVH HXNBGX GPNYSY QSHDGL EIAOSO QCEZRZ PJCVHZ FDSPPZ USINSO UHESDL ZSIOIY QHANOZ GXTJFI MHSVBP APPKFV MRHZGO UHFMWL ZSAIHV ZXOVKL MATCMT QGCCOU FDFQSU URERVV TPSKFL HXOPGS KPNYFL BTAOSK XNBVWS QSHDAV GIAIHV ZXOVUY QTSWIA EXNXSO QXSXOZ TEOJFO UHSCWW EPNYAL DRHVBK UHEVFL NJSTOA ETAOCA DXPJZP EIHZWU PXENAL JXCJOU PTNBZH ZSHZDY ABINSZ FDCJJL DPBJBK UUBVGZ MCIJQH ZUIIRH XTNYSY EDBVGZ MCIJHB DCSOCA TTJZKP EWMJBL KAEIRL DHHTZV OZAIRU MBENOU FDNDCH EIHZZV MCSBIH DPNOCY**

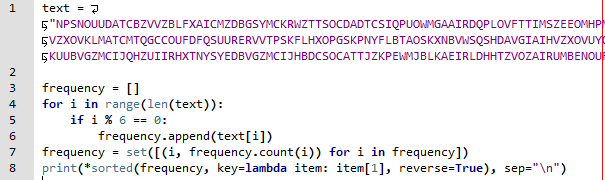
Так как по предположению длина ключа равна шести, все первые (вторые, третьи, …) буквы в колонках имеют одинаковый сдвиг. Таким образом с помощью частотного анализа мы можем их проанализировать “N U Z F Z M T A Q G Q T E M Y H G Q E Q P F U U Z Q G M A M U Z Z M Q F U T H K B X Q G Z Q E Q T U E D U N E D E P J P Z A F D U M Z X E M D T E K D O M F E M D” и понять, какими буквами они были изначально.

Воспользуемся этой таблицей (рисунок 18)[5]­:



Рисунок

На этом шаге удобно снова использовать программу для подсчёта частоты букв:



Рисунок



Рисунок

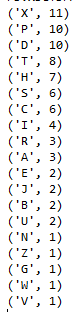
Мы видим, что хоть E, самая часто используемая в английском языке буква, на первом месте по популярности и в шифртексте, но она делит его с Q, по статистике (рис.[]) стоящей на предпоследнем месте, и M. Для удобства сравнения введём индекс частоты – сумма частотности трёх самых часто встречающихся предполагаемых букв открытого текста в процентах. Для этой комбинации он будет равен 13,11 + 0,12 + 2,54 = 15,77.

Предположим, что E превращается в Q, то есть сдвиг – **12**[6]. В таком случае в тексте также должно быть много S (-> E) и A (-> M). Для этого предположения индекс частоты равен 13,11 + 6,10 + 8,15 = **27,36**.

Если же посчитать, что E становится M, то есть сдвиг – 8, в тексте также должно быть много вхождений W (-> E) и I (-> Q), однако эти буквы встречаются реже: их индекс частоты равен 13,11 + 1,54 + 6,35 = 21,0. Индекс выше всего у второго предположения, где E -> Q.

Так как теперь мы знаем, что сдвиг всех первых букв равен 12, мы можем найти первую букву ключа – **L**.

Теперь проанализируем все вторые буквы из колонок: «PDVXDCTDPAPTEDDXPSICJDSHSHXHPRHSXAGDRPXPTNSIXTXXEHPRHJTXIXXTSBDPUCUTDCCTWAHZBDICP». Обратимся к программе:



Рисунок

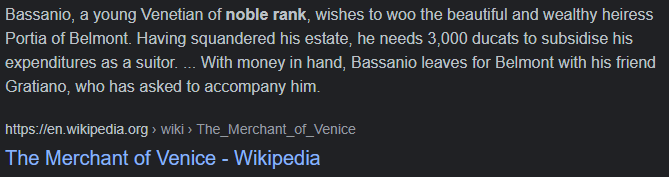
В этот раз можно однозначно выделить самую частую букву: X. Предположим, что E -> X, то есть сдвиг равен 19. Тогда W -> P и K -> D. Однако индекс слишком низкий – 13,11 + 1,54 + 0,42 = 15,07, хотя у прошлой буквы был 27.

Попробуем присвоить E второй по популярности букве – P. В таком случае ROT11. M -> X, S -> D. Индекс равен 2,54 + 13,11 + 6,10 = 21,75.

Если E -> D, то сдвиг равен 25. W -> X, O -> P, индекс = 13,11 + 1,54 + 8,00 = 22,65.

Однако если подставить E -> T со сдвигом **15**, I -> X, A -> P, то индекс сильно возрастает: 13,11 + 6,35 + 8,15 = **27,61**. Это позволяет нам понять, что вторая буква ключа – **O**.

Таким же образом я нахожу оставшиеся четыре буквы. Полный ключ выглядит так: «**LOZUNG**». С помощью программы, описанной в пункте 4, расшифровываем текст и получаем “BASSANIOAYOUNGVENETIANOFNOBLERANKWISHESTOWOOTHEBEAUTIFULANDWEALTHYHEIRESSPORTIAOFBELMONTHAVINGSQUANDEREDHISESTATEHENEEDSDUCATSTOSUBSIDISEHISEXPENDITURESASASUITORBASSANIOAPPROACHESHISFRIENDANTONIOAWEALTHYMERCHANTOFVENICEWHOHASPREVIOUSLYANDREPEATEDLYBAILEDHIMOUTANTONIOAGREESBUTSINCEHEISCASHPOORHISSHIPSANDMERCHANDISEAREBUSYATSEATOTRIPOLISTHEINDIESMEXICOANDENGLANDHEPROMISESTOCOVERABONDIFBASSANIOCANFINDALENDERSOBASSANIOTURNSTOTHEJEWISHMONEYLENDERSHYLOCKANDNAMESANTONIOASTHELOANSGUARANTORBASSANIO”. Это оказывается описание пьесы Шекспира:



Рисунок

**6 Выводы о проделанной работе**

У шифра Виженера много достоинств, таких как простота и понятность его работы. Тем не менее, даже при них его тяжело взломать. Зашифрованный им текст устойчив к частотному анализу.

Главный недостаток шифра Виженера в том, что длина используемого ключа будет меньше, чем длина текста, что приведёт к тому, что ключ будет повторяться, из чего можно найти его длину и впоследствии сам ключ.

Ограничения выбранных методов криптоанализа: так как метод Касиски полагается на частотный анализ, с его помощью будет сложно либо невозможно дешифровать короткие тексты.

Наиболее эффективные сценарии криптоанализа: найти длину ключа, разделить шифртекст на колонки по n символов, где n – длина ключа, провести частотный анализ.

# 7 Список использованных источников

1. Википедия «Шифр Виженера» – URL: <https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0>

2. Vigenere Cipher – URL: <https://www.javatpoint.com/vigenere-cipher>

3. Википедия «Метод Касиски» » – URL: https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\_%D0%9A%D0%B0%D1%81%D0%B8%D1%81%D0%BA%D0%B8

4. Jeff Suzuki: The Random Professor "Kasiski Attack" – URL: <https://www.youtube.com/watch?v=TxClRjnRNJw&ab_channel=JeffSuzuki%3ATheRandomProfessor>

5. Изображение «Entropy of English Language» – URL: <https://www.chegg.com/homework-help/questions-and-answers/based-following-table-number-births-month-us-estimate-entropy-random-variable-taking-value-q27896728>

6. Онлайн-калькулятор «Шифр Цезаря» – URL: https://planetcalc.ru/1434/

7. Портал знаний «Анализ текстов» – URL: <http://statistica.ru/local-portals/data-mining/analiz-tekstov/>

8. KMB CTF «Шифр Виженера» – URL: <https://kmb.cybber.ru/crypto/vigener/main.html>