

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 № 3
по дисциплине «Математические основы защиты информации»
ШИФР ВИЖЕНЕРА

Студент гр. БИБ 191

И.Г. Тюрин
" " _____ 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
2. Краткая теоретическая часть	4
2.1. Описание шифров	4
2.2. Методы криптоанализа шифров	5
3. Примеры шифрования	6
4. Программная реализация шифров	10
5. Примеры криптоанализа	12
6. Выводы о проделанной работе	14
7. Список использованных источников	15

1. Задание на практическую работу

Целью 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

Гаммирование заключается в наложении на открытый текст некоторой последовательности (гаммы), генерируемой на основе ключа шифрования. Под наложением гаммы на открытый текст обычно подразумевается сложение символов открытого текста с символами гаммы по модулю соответствующего алфавита. Однако в классических шифрах наложение гаммы может означать вычисление значений символов шифртекста на основе значений соответствующих символов открытого текста и гаммы по некоторому правилу.

Классическим представителем шифров гаммирования является шифр Виженера.

Символы алфавита A мощностью m представляются элементами кольца классов вычетов \mathbb{Z}_m .

Зашифрование заключается в сложении символов открытого текста с символами гаммы по модулю m .

Расшифрование заключается в сложении символов шифртекста с символами гаммы по модулю m .

В шифре Виженера в качестве ключа шифрования обычно использовалась короткая фраза, называемая лозунгом (паролем), которая циклически повторялась, формируя гамму.

Существует другой подход к формированию псевдослучайной ключевой последовательности — самоключ Виженера. Здесь в качестве начального ключа мы выбираем только один символ, к нему добавляем все символы открытого текста, за исключением последнего, и таким образом формируем гамму. Либо мы можем формировать гамму, добавляя к начальному символу поочередно символы шифртекста.

2.2. Методы криптоанализа шифров

Как повествует Википедия, шифр Виженера является шифром подстановки, то есть шифром, в котором каждая буква исходного текста заменяется буквой шифр-текста. Для вскрытия подобных шифров используется частотный криптоанализ (см. приложение).

Рассмотрим вариант использования бегущего ключа (running key, самоключ Виженера), который был когда-то был невзламываемым. Этот вариант заключается в использовании в качестве ключа блока текста, равного по длине исходному тексту. Впрочем, и этот вариант, как оказалось, успешно поддается взлому. Проблема с бегущим ключом шифра Виженера состоит в том, что криптоаналитик имеет статистическую информацию о ключе (учитывая, что блок текста написан на известном языке) и эта информация будет отражаться в зашифрованном тексте. Если ключ действительно случайный, его длина равна длине сообщения и он использовался единожды, то шифр Виженера теоретически будет невзламываемым, но такие системы уже относятся к классу систем одноразового кода, или одноразового шифр-блокнота (one-time pad). Они действительно не поддаются взлому, однако их практическое применение довольно затруднительно.

3. Примеры шифрования

1. Шифр Виженера

Открытый текст: ARTISTICALLY

Ключ: ABC

Преобразование ключа: ABCABCABCABC

Представляем открытый текст в виде номеров букв в алфавите при отсчёте с нуля:

[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]

Представляем ключ в виде номеров букв в алфавите при отсчёте с нуля:

[0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2]

Сложим поэлементно по модулю 26:

[0, 18, 21, 8, 19, 21, 8, 3, 2, 11, 12, 0]

Преобразовав числа к буквам, получим шифртекст ASVITVIDCLMA

Для расшифрования выполним вычитание ключа:

[0, 18, 21, 8, 19, 21, 8, 3, 2, 11, 12, 0] - [0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2] =

[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]

Преобразовав числа к буквам, получим текст ARTISTICALLY

2. Шифр Виженера (по открытому тексту)

Открытый текст: ARTISTICALLY

Ключ: ABC

Преобразование ключа: AARTISTICALL

Представляем открытый текст в виде номеров букв в алфавите при отсчёте с нуля:

[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]

Представляем ключ в виде номеров букв в алфавите при отсчёте с нуля:

[0, 0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11]

Сложим поэлементно по модулю 26:

[0, 17, 10, 1, 0, 11, 1, 10, 2, 11, 22, 9]

Преобразовав числа к буквам, получим шифртекст ARKBALBKCLWJ

Для расшифрования выполним вычитание ключа:

[0, 17, 10, 1, 0, 11, 1, 10, 2, 11, 22, 9] - [0, 0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11] =

[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]

Преобразовав числа к буквам, получим текст ARTISTICALLY

3. Шифр Виженера (по шифртексту)

Открытый текст: ARTISTICALLY

Ключ: ABC

Преобразование ключа: AARKSKDLNNYJ, где

Номер шага	Текущий символ	Получаемый ключ
1	Ключ(0) + Текст(0) = 0+0 = A	A + тек = AA
2	Ключ(1) + Текст(1) = 0+17 = R	AA + тек = AAR
3	Ключ(2) + Текст(2) = 17+19 = K	AAR + тек = AARK
4	Ключ(3) + Текст(3) = 10+8 = S	AARK + тек = AARKS
5	Ключ(4) + Текст(4) = 18+18 = K	AARKS + тек = AARKSK
6	Ключ(5) + Текст(5) = 10+19 = D	AARKSK + тек = AARKSKD
7	Ключ(6) + Текст(6) = 3+8 = L	AARKSKD + тек = AARKSKDL
9	Ключ(7) + Текст(7) = 11+2 = N	AARKSKDL + тек = AARKSKDLN
9	Ключ(8) + Текст(8) = 13+0 = N	AARKSKDLN + тек = AARKSKDLNN
10	Ключ(9) + Текст(9) = 13+11 = Y	AARKSKDLNN + тек = AARKSKDLNNY
11	Ключ(10) + Текст(10) = 24+11 = J	AARKSKDLNNY + тек = AARKSKDLNNYJ

Представляем открытый текст в виде номеров букв в алфавите при отсчёте с нуля:

[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]

Представляем ключ в виде номеров букв в алфавите при отсчёте с нуля:

[0, 0, 17, 10, 18, 10, 3, 11, 13, 13, 24, 9]

AARKSKDLNNYJ

Сложим поэлементно по модулю 26:

[0, 17, 10, 18, 10, 3, 11, 13, 13, 24, 9, 7]

Преобразовав числа к буквам, получим шифртекст ARKSKDLNNYJH

Для расшифрования выполним вычитание ключа:

$[0, 17, 10, 18, 10, 3, 11, 13, 13, 24, 9, 7] - [0, 0, 17, 10, 18, 10, 3, 11, 13, 13, 24, 9] =$

$[0, 17, 19, 8, 18, 19, 8, 2, 0, 11, 11, 24]$

Преобразовав числа к буквам, получим текст ARTISTICALLY

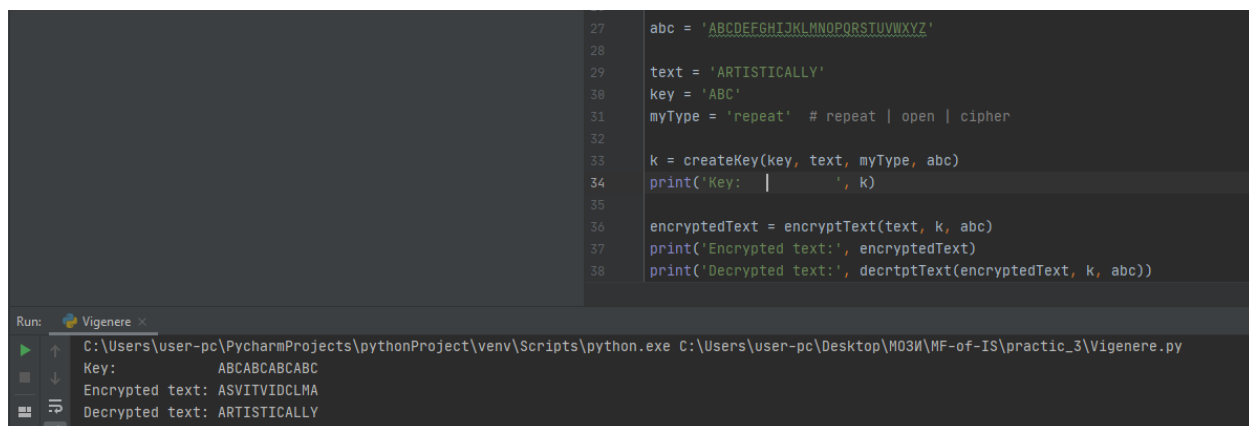
4. Программная реализация шифров

Результаты работы программы для вышеприведённых примеров шифрования:

1) Шифр Виженера

Открытый текст: ARTISTICALLY

Ключ: ABC



```
27 abc = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
28
29 text = 'ARTISTICALLY'
30 key = 'ABC'
31 myType = 'repeat' # repeat | open | cipher
32
33 k = createKey(key, text, myType, abc)
34 print('Key: ', k)
35
36 encryptedText = encryptText(text, k, abc)
37 print('Encrypted text:', encryptedText)
38 print('Decrypted text:', decryptText(encryptedText, k, abc))
```

Run: Vigenere x

C:\Users\user-pc\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\user-pc\Desktop\M03И\MF-of-IS\practic_3\Vigenere.py

Key: ABCABCABC

Encrypted text: ASVITVIDCLMA

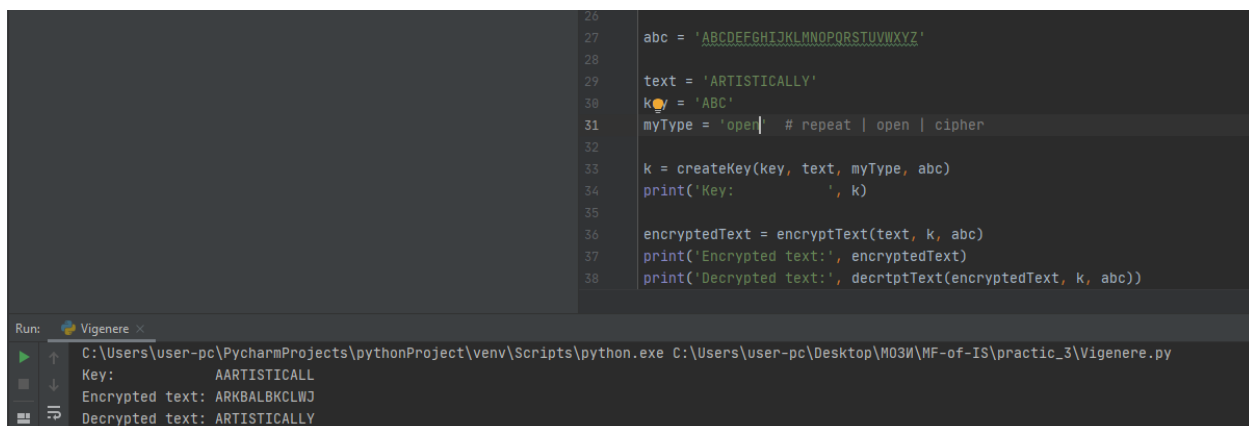
Decrypted text: ARTISTICALLY

Рисунок 4.1 – Результаты работы Алгоритма, реализующего шифр Виженера

2) Шифр Виженера (по открытому тексту)

Открытый текст: ARTISTICALLY

Ключ: ABC



```
27 abc = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
28
29 text = 'ARTISTICALLY'
30 key = 'ABC'
31 myType = 'open' # repeat | open | cipher
32
33 k = createKey(key, text, myType, abc)
34 print('Key: ', k)
35
36 encryptedText = encryptText(text, k, abc)
37 print('Encrypted text:', encryptedText)
38 print('Decrypted text:', decryptText(encryptedText, k, abc))
```

Run: Vigenere x

C:\Users\user-pc\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\user-pc\Desktop\M03И\MF-of-IS\practic_3\Vigenere.py

Key: AARTISTICALL

Encrypted text: ARKBALBKCLWJ

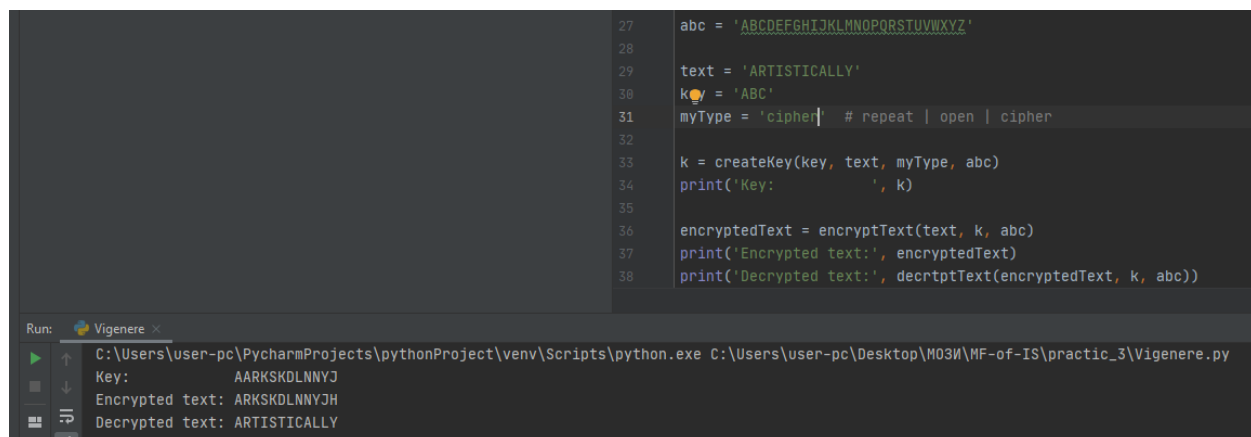
Decrypted text: ARTISTICALLY

Рисунок 4.2 – Алгоритм, реализующий шифр Виженера (по открытому тексту)

3) Шифр Виженера (по шифртексту)

Открытый текст: ARTISTICALLY

Ключ: ABC



```
27 abc = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
28
29 text = 'ARTISTICALLY'
30 key = 'ABC'
31 myType = 'cipher' # repeat | open | cipher
32
33 k = createKey(key, text, myType, abc)
34 print('Key: ', k)
35
36 encryptedText = encryptText(text, k, abc)
37 print('Encrypted text:', encryptedText)
38 print('Decrypted text:', decryptText(encryptedText, k, abc))
```

Run: Vigenere x

C:\Users\user-pc\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\user-pc\Desktop\М03И\МФ-of-IS\practic_3\Vigenere.py

Key: AARKSKDLNNYJ

Encrypted text: ARKSKDLNNYJH

Decrypted text: ARTISTICALLY

Рисунок 4.4 – Алгоритм, реализующий рекуррентный шифр Хилла

5. Криптоанализ шифра Виженера

1) Теоретическая часть

Шифр Виженера «размывает» характеристики частот появления символов в тексте, но некоторые особенности появления символов в тексте остаются. Главный недостаток шифра Виженера состоит в том, что его ключ повторяется. Поэтому простой криптоанализ шифра может быть построен в два этапа:

1. Поиск длины ключа. Можно анализировать распределение частот в зашифрованном тексте с различным прореживанием. То есть брать текст, включающий каждую 2-ю букву зашифрованного текста, потом каждую 3-ю и т. д. Как только распределение частот букв будет сильно отличаться от равномерного (например, по энтропии), то можно говорить о найденной длине ключа. Определение длины ключа возможно при помощи тестов Фридмана и Касиски
2. Криптоанализ. Совокупность / шифров Цезаря (где / — найденная длина ключа), которые по отдельности легко взламываются.

2) Практическая часть (программный метод)

Ссылка на алгоритм взлома:

<https://github.com/asweigart/codebreaker/blob/master/vigenereHacker.py>

3) Практическая часть (взлом без использования программы)

Рассмотрим данный алгоритм на примере (в данном случае определим длину методом Касиски).

Возьмём шифртекст:

wjvhujectatunpbzmgiojithitvsxktivhjugsgauryujooupjvtjpgjvswgnpoovukftotgnfwppptigwptlevhfdlpqddq
quofeekniksigaeacnejtjaofsxgrfvhfjaofspoeboaknheoofudvosrlbaioiamntigszopiqnjsphbmczjpgb
pdcwroknhvotiooidpynuueuctugrtcneehbtcpclswiouogjivtosa

Заметим, что сочетание ji повторяется через 93 и 123 символа. Отсюда можно сделать вывод, что 93 и 123 кратны размеру ключа => размер ключа 3 (так как делители 93: 3 и 31, а делители 123: 3 и 41).

Теперь, зная длину ключа можно легко перебрать все возможные варианты ключей. Однако при более длинных ключах этот процесс может затянуться. В таком случае, мы можем использовать дополнительную информацию о части открытого текста. Пусть нам известно, что в тексте содержится комбинация "withthe". Тогда мы можем воспользоваться вычитанием шифртекста из себя же, но сдвинутого на длину ключа.

Шифртекст:

wjvhujectatunpbzmgiojithitvsxktivhjugsgauryujooupjvtjpgjvswgnpoovukftotgnfwppptigwptlevhfdlpqddq
quofeekniksigaeacnejtjaofsxgrfvhfjaofspoeboaknheoofudvosrlbaioiamntigszopiqnjsphbmczjpgb
pdcwroknhvotiooidpynuueuctugrtcneehbtcpclswiouogjivtosa

Сдвинутый шифртекст:

hujectatunpbzmgiojithitvsxktivhjugsgauryujooupjvtjpgjvswgnpoovukftotgnfwppptigwptlevhfdlpqddquof
eekniksigaeacnejtjaofsxgrfvhfjaofspoeboaknheoofudvosrlbaioiamntigszopiqnjsphbmczjpgbpd
wroknhvotiooidpynuueuctugrtcneehbtcpclswiouogjivtosa____

Полученная разница:

ppmdsqejznetodvryxavcaamqwlzppmzbbrogypcaikgpzfwagnauonpfhszuueqbwmbokqyhwhjxtnllye
zswqnizajclqkvrwavaeseenatflaifeirzbslkamhreizyaatoaebentgtzulkgleddrrdsicvhehbjdsryczkvuzrdf
cdntiadztmomehpzfcgolfzkqvpjahlawcbeepygdflmrrxgdecuilgnoubvesa

Разница по кодовому слову: ppmd____ Данный участок совпадает с началом! Следовательно, мы нашли позицию ключа в шифртексте. Теперь вычитаем withthe из данной позиции:

получаем:

abca*****

*****. Так как длина ключа = 3, наш ключ abc.

Расшифруем данным ключом шифртекст. Получим осмысленный текст:

with_the_brass_nozzle_in_his_fists_with_this_great_python_spitting_its_venomous_kerosene_upon
_the_world_the_blood_pounded_in_his_head_and_his_hands_were_the_hands_of_some_amazing
_conductor_playing_all_the_symphonies_of_blazing_and_burning_to_bring_down_the_tatters_and_
charcoal_ruins_of_history (подчёркивания расставлены вручную вместо пробелов)

6. Выводы о проделанной работе

Шифр Виженера уязвим для атак типа «только шифртекст», однако при использовании других алгоритмов генерации ключа сложность взлома увеличивается. Несмотря на этот факт, в шифртексте (или ключах) сохраняются статистические зависимости. Это позволяет получить ключ шифрования используя комбинированные методы перебора и статистического анализа.

7. Список использованных источников

1. Традиционные шифры с симметричным ключом [электронный ресурс] – URL: <https://intuit.ru/studies/courses/552/408/lecture/9355?page=5>
2. Cryptanalysis of the Hill Cipher [электронный ресурс] – URL: <http://practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-hill-cipher/>
3. Хилл шифр – Hill cipher [электронный ресурс] – URL: https://ru.qaz.wiki/wiki/Hill_cipher

ПРИЛОЖЕНИЕ А.

Шифр Виженера

```
def encryptText(text, key, abc):
    return ''.join([abc[(abc.index(j) + abc.index(key[i])) % 26 +
abc.index('A')]] for i, j in enumerate(text)])

def decryptText(decodedText, key, abc):
    return ''.join([abc[(abc.index(j) - abc.index(key[i])) % 26 +
abc.index('A')]] for i, j in enumerate(decodedText)])

def createKey(key, text, myType, abc):
    l = len(text)
    k = len(key)

    if myType == 'repeat':
        res = key * int((l // k + 1))
    elif myType == 'open':
        res = key[0] + text[:-1]
    elif myType == 'cipher':
        key = key[:1]
        for j, i in enumerate(text):
            c = abc[(abc.index(i) + abc.index(key[j])) % len(abc)]
            key += c
        res = key

    return res

abc = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

text = 'SOMETEXT'
key = 'KEY'
myType = 'cipher' # repeat | open | cipher

k = createKey(key, text, myType, abc)

encryptedText = encryptText(text, k, abc)
print('Encrypted text:', encryptedText)
print('Decrypted text:', decryptText(encryptedText, k, abc))
```