# Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Информационная безопасность (Криптография)
Лабораторная работа 1.1

Вариант 2

Работу выполнил студент группы Р34111: Кривоносов Егор Дмитриевич

Преподаватель:

Маркина Татьяна Анатольевна

2022 г.

г. Санкт-Петербург

## Оглавление

Цель работы	3
Задание	3
Листинг разработанной программы	3
Результаты работы программы	7
Шифровка	7
Пример работы:	7
Скриншоты вывода:	7
Шифрование	7
Частотный анализ	8
Расшифровка обычным частотным анализом	9
Расшифровка	9
Пример работы:	9
Скриншоты вывода:	9
Вывод	9
Полезные ссылки	11

## Цель работы

Изучение основных принципов шифрования информации, знакомство с широко известными алгоритмами шифрования, приобретение навыков их программной реализации.

## Задание

Реализовать шифрование и дешифрацию файла по методу Виженера. Ключевая фраза вводится. Реализовать в программе частотный криптоанализ зашифрованного текста.

## Листинг разработанной программы

```
from collections import Counter
from tabulate import tabulate
# Считывает текст из файла
def read file():
    while True:
        path = input ("Введите путь к файлу: ").strip()
        try:
            with open (path, encoding='utf-8') as f:
                data = f.readlines()
                return data[0]
        except FileNotFoundError:
            print("Файл не найден!", '\n')
        except ValueError:
            print("Неправильный формат файла!", '\n')
# Проверяет какой язык используется
def check lang(plaintext, lang 1, lang 2):
   spec character = ",.:;'\"-?!/1234567890[]= ..."-«»"*&^(){}"
   upper text = ""
    answer = []
    for letter in plaintext:
        upper text += letter.upper()
        if spec character.find(letter) != -1:
            continue
       num = lang 1.find(letter.upper())
    if num !=-1:
        answer.append("EN")
```

```
for letter in plaintext:
        if spec character.find(letter) != -1:
            continue
        num = lang 2.find(letter.upper())
    if num != -1:
        answer.append("RU")
    answer.append(upper text)
    return answer
def encrypt(plaintext, key, ALPHABET):
    result text = ""
   key_ind = 0
    for letter in plaintext:
        num = ALPHABET.find(letter.upper())
        if num != -1:
           num = (ALPHABET.find(letter.upper()) + ALPHABET.find(key[key ind % len(key)])) %
len (ALPHABET)
            key ind += 1
            if letter.islower():
                result text += ALPHABET[num].lower()
            else:
                result text += ALPHABET[num]
        else:
            result text += letter
    return result text
def decrypt(ciphertext, key, ALPHABET):
    result text = ""
    key ind = 0
    for letter in ciphertext:
        num = ALPHABET.find(letter.upper())
        if num != -1:
            num = (ALPHABET.find(letter.upper()) - ALPHABET.find(key[key ind % len(key)])) %
len (ALPHABET)
            key ind += 1
            if letter.islower():
                result text += ALPHABET[num].lower()
            else:
               result text += ALPHABET[num]
        else:
            result text += letter
    return result text
# Формируем словарь сопоставляющий по частоте буквы в тексте и в языке
def mapping dict(counter, DICT):
    decrypt arr = []
    # Упорядочиваем по частоте появление букв в тексте
```

```
for i in range(len(counter)):
        decrypt arr.append(counter.most common(len(counter))[i][0])
    return dict(zip(decrypt arr, DICT))
# Простой частотный анализ
def freq analysis(ciphertext, ALPHABET, DICT):
    spec_character = ",.:;'\"-?!/1234567890[]= ..."-\(\*\)" \\ \\^() \{\}"
    modified ciphertext = "".join([(char if char not in spec character else '') for char in
ciphertext])
   modified ciphertext = modified ciphertext.replace('\n', '')
    chars counter = Counter(modified ciphertext.upper())
    amount = 0
    for i in range(len(ALPHABET)):
        amount += chars counter[ALPHABET[i]]
   table = []
    for i in range(len(ALPHABET)):
        table.append([ALPHABET[i], chars counter[ALPHABET[i]], round(chars counter[ALPHABET[i]]
/ amount * 100, 3)])
   print("\n Частотный анализ:")
   print(tabulate(table, headers=["Буква", "Количество", "Частота"], tablefmt="grid",
floatfmt="2.5f"), "\n")
    new DICT = mapping dict(chars counter, DICT)
    print(new DICT)
    # Если в исходном зашифрованном тексте встречается символ, то сохраняется, иначе буква
сопоставляется по словарю
    new result = ""
    j = 0
    for i in ciphertext:
        num = ALPHABET.find(i.upper())
        if num != -1:
            if ciphertext[j].islower():
                new result += (new DICT.get(modified ciphertext[j].upper())).lower()
                new result += new DICT.get(modified ciphertext[j].upper())
            j += 1
        else:
            new result += i
    return new result
# Запускает основную программу
if name == ' main ':
    # Алфавит для сдвига
   ALPHABET EN = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
   ALPHABET RU = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЦЬЫЬЭЮЯ"
    # Упорядоченные по частоте появления в языке буквы
    DICT EN = [*'ETAOINSHRDLCUMWFGYPBVKJXQZ']
```

```
DICT RU = [*'ОЕЁАИТНСРВЛКМДПУЯЫГЗБЧЙХЪЖЬЮЩЦЩЭФ']
   mode = ""
   key = ""
   key_lang = ""
   ALPHABET = ALPHABET RU
   DICT = DICT RU
    # Считываем текст с файла
    text = read file()
    text lang = check lang(text, ALPHABET EN, ALPHABET RU)[0]
   print(f"Исходный текст: {text} \n")
    # Выбираем режим Шифрование / Расшифрование
    while True:
       mode = input("Введите 'enc' для шифрования или 'dec' для расшифрования: ")
        if mode == "enc" or mode == "dec":
       print("Выбран неправильный режим!", '\n')
    key = input("Введите ключ: ")
    key lang, new key = check lang(key, ALPHABET EN, ALPHABET RU)
    # Проверяем языки ключа и исходного текста
    if key lang != text lang:
       print("Язык ключа и текста не совпадают, попробуйте снова!")
        print(f"key lang: {key lang} != {text lang} :text lang")
        exit(-1)
    if text lang == "EN":
       ALPHABET = ALPHABET EN
       DICT = DICT EN
    # Шифрование и расшифрование исходного текста
    result = encrypt(text, new key, ALPHABET) if mode == "enc" else decrypt(text, new key,
ALPHABET)
   print(f"Результат: {result}")
    # Если у нас было шифрование, тогда производим частотный анализ
    if mode == "enc":
       print(f"Частотный анализ результат: {freq analysis(result, ALPHABET, DICT)}")
```

## Результаты работы программы

#### Шифровка

#### Пример работы:

**Исходный текст:** Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Ключ: redgry

**Результат:** Csukd Ggwxs zq jmpvcw uypsp rvbw uw ryi sxzlkmqm rlu xbvvqvxwoee zrgajric. Ouicd Msylk yev hvce xkk zluyvziw'j wwgebrvg jlkdc wkor vzhx jgegh zyc 1500j, akke ye yqqemnr sxzlkiu zfmb e jgcjvc rl kwgi dtu qtvdssjvh lz km denk r rpth ygctmpke zfsn. Ok frw vaitzzhj emk sqrp dzzh ivlkyuovq, syw gcqf xkk ccrt ltkm vphikpfrli kwgivkkrzrj, xvkrmqoee vwvkerzeorp segkgeevh. Lz nyj trvljrvlyvb zr wnv 1960q nmwn kfv vhrvyji rl Cckvdyvr jlhkkq tsqzrgemqm Cmiip Ogqlq sgjqrkhy, rlu qrxv pvghtkjp alzy bvwnzfn gyerzqymqm jmwxzgic cmnk Rjuyv VrevQdqvp zrfrlbzrj bvpjmrtj mw Prxvk Ztvad.

#### Скриншоты вывода:

#### Шифрование

```
Becaute nyts k qaany: tests/en
Mcxoquebb rexcr: Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

It was popularised in the 1900s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Becaute 'enc' gna mudpopaemum unu 'dec' gna pacumuppomanum: enc

Becaute knew: redgry

Pesynetar: Caukd Ggaxs zq jmpvcm uypsp rvbw um ryi sxzlkmqm rlu xbvvqxxmoee zrgajric.

Ouicd Msylk yew hvce xkk zluyvzlw'j wwgebrvg jlddc wkor vzhx jgegh zyc 1500j, akke ye yqqemnr sxzlkiu zfmb e jgcjvc rl kwgi dtu qtvdssjvh lz km denk r rpth ygctmpke zfsn.

Ok frm valtzchj enk sprg dezb ivlkyvoug, syw gocf xkk cort lttm ynhipfril kmgikkrarj, xwrmqoee vmvkerzeorp segkgeevh.

Lz nyj trvljrvlyvb zr wnv 1900g nmmn kfv vhrvyji rl Cckvdyvr jlhkkq tsqzrgemqm Cmiip Ogqlq sgjqrkhy, rlu qrvv pvghtkjp alzy bvmzfn gyerzqymqm jmmxzgic cmnk Rjuyv Vrevqdqvp zrfrlbzrj bvpjmrtj mw Prxvk Ztvad.
```

#### Частотный анализ

Частотный а	анализ:	
Буква   +======+	Количество	Частота
A	5	1.07100
B	8	1.71300
C	18	3.85400
D	10	2.14100
E	23	4.92500
F	8	1.71300
G	20	4.28300
H	13	2.78400
I	15	3.21200
J	22	4.71100
K	35	7.49500
	20	4.28300
M	21	4.49700
N	10	2.14100
10 1	8	1.71300
P	15	3.21200
1 Q	22	4.71100
R	38	8.13700
S	14	2.99800
T	12	2.57000
U	11	2.35500
l v l	41	8.77900
W	19	4.06900
1 x 1	12	2.57000
Y	20	4.28300
Z	27	5.78200
+		++

#### Расшифровка обычным частотным анализом

```
MacrothwW ahanwa pesymbrat: Ufbav RRcyf oN shMeuc bdmFm tejc Bc tdw FyolahnH tlb yjeenEyoxIi otrzstwu.
xbMuv hfdla dIE Geui yAA olbdEowc's ceriJter Slavu caxt EoBy sring odu 1508s, zaai Di dmnInkt ffolamb Oqhj I SRUSeU Tl acrW vpB npevffsEg lo ah viKa t tmPg DruphmaI oqFk.
xa qTc ezWpoogs iha fMtm Wood welaDbXen, fdc runq yaa uUtp lpAH EmBowamQtlw acrweAatoTs, yeaThnxiI ecesitoixTM firAriieG.
lo KDs pteistelDej ot cke 1980m khCk age egtedsw Tl uuweevdet sigaan pfnotriHnh uhwmm xrNLN fRantAgd, tlb ntye meRgpAsm zIDD JEcKoqk rDitoNdhnh shcYorWU uhka tsbDe etienYnem otqtljoTS jemsHTps hc Mtyea opezV.
```

#### Расшифровка

#### Пример работы:

**Исходный текст:** Csukd Ggwxs zq jmpvcw uypsp rvbw uw ryi sxzlkmqm rlu xbvvqvxwoee zrgajric. Ouicd Msylk yev hvce xkk zluyvziw'j wwgebrvg jlkdc wkor vzhx jgegh zyc 1500j, akke ye yqqemnr sxzlkiu zfmb e jgcjvc rl kwgi dtu qtvdssjvh lz km denk r rpth ygctmpke zfsn. Ok frw vaitzzhj emk sqrp dzzh ivlkyuovq, syw gcqf xkk ccrt ltkm vphikpfrli kwgivkkrzrj, xvkrmqoee vwvkerzeorp segkgeevh. Lz nyj trvljrvlyvb zr wnv 1960q nmwn kfv vhrvyji rl Cckvdyvr jlhkkq tsqzrgemqm Cmiip Ogqlq sgjqrkhy, rlu qrxv pvghtkjp alzy bvwnzfn gyerzqymqm jmwxzgic cmnk Rjuyv VrevQdqvp zrfrlbzrj bvpjmrtj mw Prxvk Ztvad.

Ключ: redgry

**Результат:** Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

#### Скриншоты вывода:

```
Begarre nyto k $aäny: tests/en_enc
Mcxoganwä Text: Gsudd Ggwxs zq japrow wypsp rvbw uw ryi sxzlkmqm rlu xbvvqvxwoee zrgajric.
Ouicid Msytk yew hove kak zluyvzim'j wwgebrvg jlkdo skor vzhx jageh zyc 1500j, akke ye yqqemnr sxzlkiu zfmb e jgcjvc rl kmgi dtu qtvdssjvh lz km denk r rpth ygctmpke zfsn.
Ok frw vaitzzhj emk sqrp dzzh ivlkyuovq, syw godf xkk cort ltkw vphikpfrli kwgivkkrzrj, xvkrmqoee www.kerzeorp segkgeevh.
Lz nyj trvljrvlyvb z mwn 1960q nmwn kfv vhrvyji rl Cokvdyvr jlhkkq tsqzrgemqm Cmiip Ogqlq sgjqrkhy, rlu qrxv pvghtkjp alzy bwmzfn gyerzqymqm jmwzzgic cmnk Rjuyv VrevQdqvp zrfrlbzrj bvpjartj mw Prxvk Ztvad.

Begarre 'enc' для шw6pomanwa wnu 'dec' для расшw6pomanwa: dec
Begarre know: redgry
Pezymbrari: Lorem Ijsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

It was popularised in the 1960s with the release of letraset sheets containing lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
```

## Вывод

В результате выполнения лабораторной работы были получены навыки программной реализации алгоритма Виженера на языке Python и проведения его частотного анализа простым способом. Обычный частотный анализ очевидно не даёт внятного результата, в силу того, что шифр Виженера по сути состоит из нескольких частей текста, каждая из

которых имеет свой сдвиг, а значит и анализ надо проводить для каждого из блоков текста по отдельности.

## Полезные ссылки

https://habr.com/ru/post/517410/ - частотный анализ
https://gist.github.com/dssstr/aedbb5e9f2185f366c6d6b50fad3e4a4 - реализации метода
Виженера

Как работает метод Виженера

https://planetcalc.ru/2468/ - шифр Виженера онлайн