

Отчёт по лабораторной работе №3

Шифр гаммирования

Шевляков Илья Николаевич НФИмд-01-21

Содержание

Цель работы	4
Теоретические сведения	5
Шифр гаммирования	5
Выполнение работы	7
Реализация шифратора и дешифратора Python	7
Контрольный пример	11
Выводы	12
Список литературы	13

Список иллюстраций

0.1	Работа алгоритма гаммирования	11
-----	---	----

Цель работы

Изучение алгоритма шифрования гаммированием

Теоретические сведения

Шифр гаммирования

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (например, используя операцию сложения по модулю 2). Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Метод гаммирования становится бессильным, если известен фрагмент исходного текста и соответствующая ему шифрограмма. В этом случае простым вычитанием по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Метод гаммирования с обратной связью заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных. Например, если рассматривать гамму шифра как объединение непересекающихся множеств $H(j)$, то процесс шифрования можно представить следующими шагами:

1. Генерация сегмента гаммы $H(1)$ и наложение его на соответствующий участок шифруемых данных.
2. Подсчет контрольной суммы участка, соответствующего сегменту гаммы $H(1)$.
3. Генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гамм $H(2)$.
4. Подсчет контрольной суммы участка данных, соответствующего сегменту данных $H(2)$ и т.д.

Выполнение работы

Реализация шифратора и дешифратора Python

```
def gamma():
    dict = {"a": 1, "б": 2, "в": 3, "г": 4, "д": 5, "е": 6, "ё": 7, "ж": 8, "з": 9, "и": 10,
            "м": 14, "н": 15, "о": 16, "п": 17,
            "р": 18, "с": 19, "т": 20, "у": 21, "ф": 22, "х": 23, "ц": 24, "ч": 25, "ш": 26,
            "ы": 29, "ь": 30, "э": 31, "ю": 32, "я": 32}
    dict2 = {v: k for k, v in dict.items()}
    gamma_ = input('Введите гамму: \n').lower()
    text = input('Введите текст для зашифровки: \n').lower()
    listofdigitsoftext = list()
    listofdigitsofgamma = list()
    for i in text:
        listofdigitsoftext.append(dict[i])
    print('Числа текста: \n', listofdigitsoftext)
    for i in gamma_:
        listofdigitsofgamma.append(dict[i])
    print('Числа гамма: \n', listofdigitsofgamma)
    listofdigetsresult = list()
    tmp = 0
    for i in text:
        try:
```

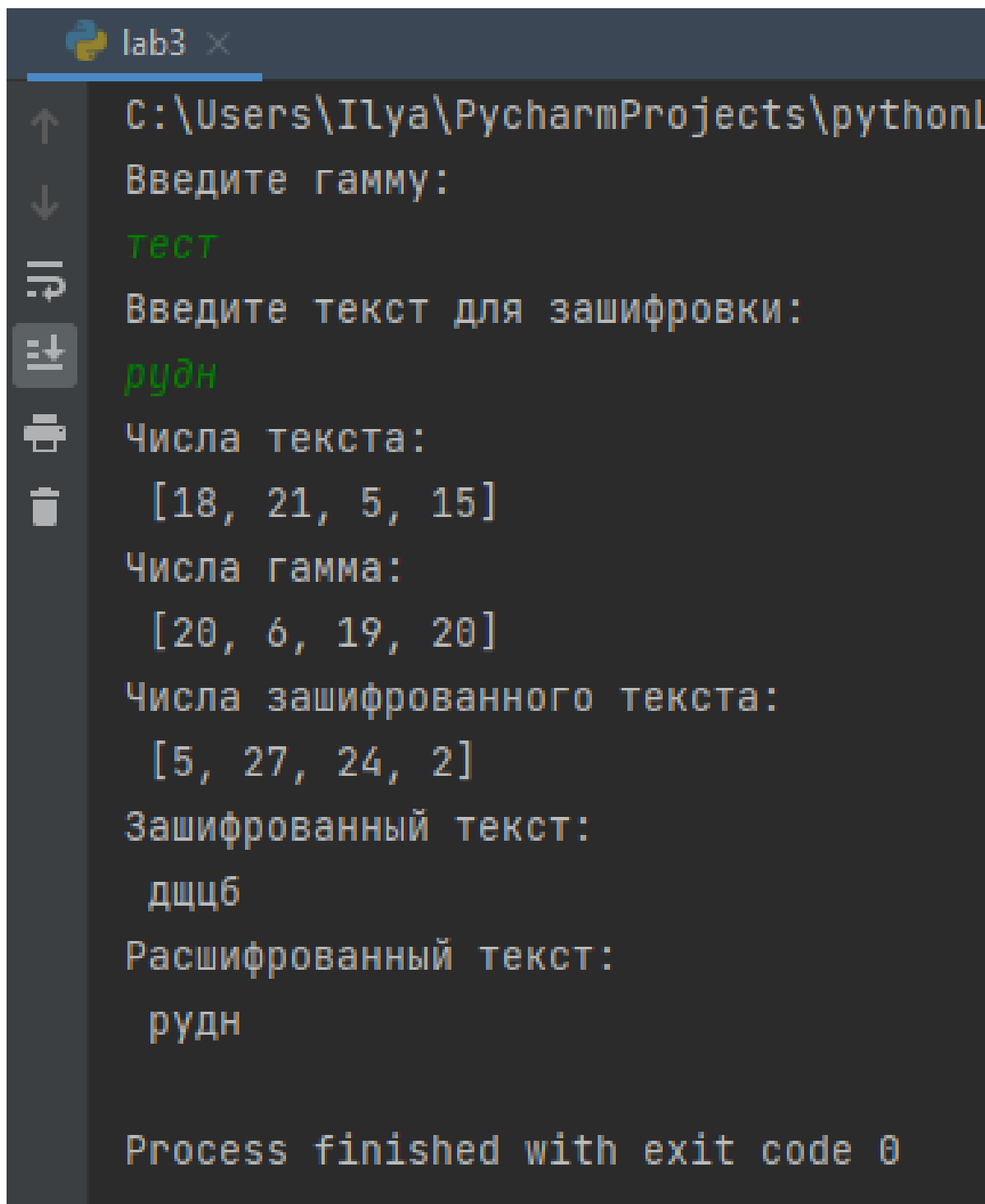
```

        a = dict[i] + listofdigitsofgamma[tmp]
except:
    tmp = 0
    a = dict[i] + listofdigitsofgamma[tmp]
if a >= 33:
    a = a % 33
tmp += 1
listofdigetsresult.append(a)
print('Числа зашифрованного текста: \n', listofdigetsresult)
txtencryp = ''
for i in listofdigetsresult:
    txtencryp += dict2[i]
print('Зашифрованный текст: \n', txtencryp)
listofdigets = list()
for i in txtencryp:
    listofdigets.append(dict[i])
tmp = 0
listofdigets_ = list()
for i in listofdigets:
    a = i - listofdigitsofgamma[tmp]
    if a < 1:
        a = 33 + a
    listofdigets_.append(a)
    tmp += 1
txtdecryp = ''
for i in listofdigets_:
    txtdecryp += dict2[i]
print('Расшифрованный текст: \n', txtdecryp)

```


`gamma()`

Контрольный пример



```
lab3 x
C:\Users\Ilya\PycharmProjects\pythonL
Введите гамму:
тест
Введите текст для зашифровки:
рудн
Числа текста:
[18, 21, 5, 15]
Числа гамма:
[20, 6, 19, 20]
Числа зашифрованного текста:
[5, 27, 24, 2]
Зашифрованный текст:
дщцб
Расшифрованный текст:
рудн
Process finished with exit code 0
```

Рис. 0.1: Работа алгоритма гаммирования

Выводы

Изучили алгоритмы шифрования на основе гаммирования

Список литературы

1. Шифрование методом гаммирования
2. Режим гаммирования в блочном алгоритме шифрования