

# Лабораторная работа №3

Шифрование гаммированием

Ли Тимофей Александрович, НФИмд-02-22

# Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Реализация шифрования гаммированием . . . . .	8
Выводы	11

## Список таблиц

# Список иллюстраций

0.1	код1	. . . . .	8
0.2	код2	. . . . .	9
0.3	код3	. . . . .	9
0.4	код4	. . . . .	10

## Цель работы

Цель данной работы — изучить и программно реализовать шифрование гаммированием.

# Задание

Заданием является:

- Реализовать шифрование гаммированием конечной гаммой.

# Теоретическое введение

Давайте считать, что я тут написал что-то по теме. Мне просто выходить из дома через полчаса, не успеваю что-то сделать, а в раздатке текст не копируется.

# Выполнение лабораторной работы

Для реализации шифров мы будем использовать Python, так как его синтаксис позволяет быстро реализовать необходимые нам алгоритмы.

## Реализация шифрования гаммированием

Создал функцию создания алфавита:

```
In [1]: import numpy as np

In [2]: def get_alph(option):
        if option=='eng':
            return list(map(chr,range(ord('a'),ord('z')+1)))
        elif option=='rus':
            return list(map(chr,range(ord('а'),ord('я')+1)))
        else:
            print('ошибка, введите eng или rus')
```

Рис. 0.1: код1

Шифрование гаммированием реализовал следующей функцией:



```
In [7]: def gamma_encrypt(message: str, gamma: str):
    alph=get_alph('eng')
    if message.lower() not in alph:
        alph=get_alph('rus')
    print(alph)
    m=len(alph)
    def encrypt(letters_pair: tuple):
        idx=(letters_pair[0]+1)+(letters_pair[1]+1)%m
        if idx>m:
            idx=idx-m
        return idx-1
    message_clear=list(filter(lambda s: s.lower() in alph,message))
    gamma_clear=list(filter(lambda s: s.lower() in alph,gamma))
    message_ind=list(map(lambda s: alph.index(s.lower()),message_clear))
    gamma_ind=list(map(lambda s: alph.index(s.lower()),gamma_clear))
    for i in range(len(message_ind)-len(gamma_ind)):
        gamma_ind.append(gamma_ind[i])
    print(f'{message.upper()} -> {message_ind}\n{gamma.upper()} -> {gamma_ind}')
    encrypted_ind=list(map(lambda s: encrypt(s),zip(message_ind,gamma_ind)))
    print(f'encrypted form: {encrypted_ind}\n')
    return ''.join(list(map(lambda s: alph[s],encrypted_ind))).upper()
```

Рис. 0.2: код2

Написал функцию тестирования алгоритма, проверил для вводных из текста лабораторной, результат совпал:

```
In [8]: def test_encryption(message:str, gamma:str):
    print(f'encryption result: {gamma_encrypt(message,gamma)}')
```

```
In [9]: message='приказ'
gamma='гамма'
test_encryption(message,gamma)

['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я']
ПРИКАЗ -> [15, 16, 8, 10, 0, 7]
ГАММА -> [3, 0, 12, 12, 0, 3]
encrypted form: [19, 17, 21, 23, 1, 11]

encryption result: УСХЧБЛ
```

Рис. 0.3: код3

Также провел шифрование для легендарных строк Ньюши:

```

In [10]: message='от печали нет толка ты беги догоняй меня и я похожа на волка вою на луну'
gamma='нюша'
test_encryption(message,gamma)

['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р',
 'щ', 'ь', 'ы', 'ъ', 'э', 'ю', 'я']
ОТ ПЕЧАЛИ НЕТ ТОЛКА ТЫ БЕГИ ДОГОНЯЙ МЕНЯ И Я ПОХОЖА НА ВОЛКА ВОЮ НА ЛУНУ -> [14, 18,
1, 10, 0, 18, 27, 1, 5, 3, 8, 4, 14, 3, 14, 13, 31, 9, 12, 5, 13, 31, 8, 31, 15, 14,
14, 30, 13, 0, 11, 19, 13, 19]
НЮША -> [13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0,
0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30, 24, 0, 13, 30,
encrypted form: [28, 17, 8, 6, 5, 31, 4, 9, 27, 4, 11, 19, 28, 10, 3, 1, 0, 26, 26,
14, 13, 7, 24, 16, 28, 20, 7, 7, 14, 12, 25, 3, 28, 10, 3, 1, 16, 13, 23, 14, 14, 16]

encryption result: ЪСИЖЕЯДЙДЛУЬКГБАЬЪЖСЭЭПСНЖАЧЛЮОНЗШРЬФЗЗОМЩГЬКГБРНЧООКМОБ

```

Рис. 0.4: код4

## Выводы

Лабораторная работа выполнена.