

# **Отчет по лабораторной работе №8**

**Элементы криптографии. Шифрование (кодирование) различных  
исходных текстов одним ключом**

Асеинова Елизавета

2022 Oct 5th

# Содержание

1. Цель работы	5
2. Выполнение лабораторной работы	6
3. Контрольные вопросы	8
4. Выводы	10
5. Список литературы	11

## **Список таблиц**

# Список иллюстраций

2.1. Функции . . . . .	6
2.2. Создание ключа . . . . .	6
2.3. Шифрование . . . . .	7

# 1. Цель работы

Целью данной работы является освоение на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом. [1]

## 2. Выполнение лабораторной работы

1. Импортировала необходимые библиотеки, задала функцию для генерации ключа, преобразованию ключа в шестнадцатеричное представление, и для шифрования текста.

```
[7] import string
    import random

[8] def shest(message):
    return ' '.join(hex(ord(i))[2:] for i in message)

    def rand_key(s):
    return ' '.join(random.choice(string.ascii_letters + string.digits) for _ in range(s))

    def code(message1, message2):
    mess1 = [ord(i) for i in message1]
    mess2 = [ord(i) for i in message2]
    return ' '.join(chr(a^b) for a,b in zip(mess1, mess2))
```

Рис. 2.1.: Функции

2. Задала 2 текста, создала ключ, преобразовала его в шестнадцатеричное представление.

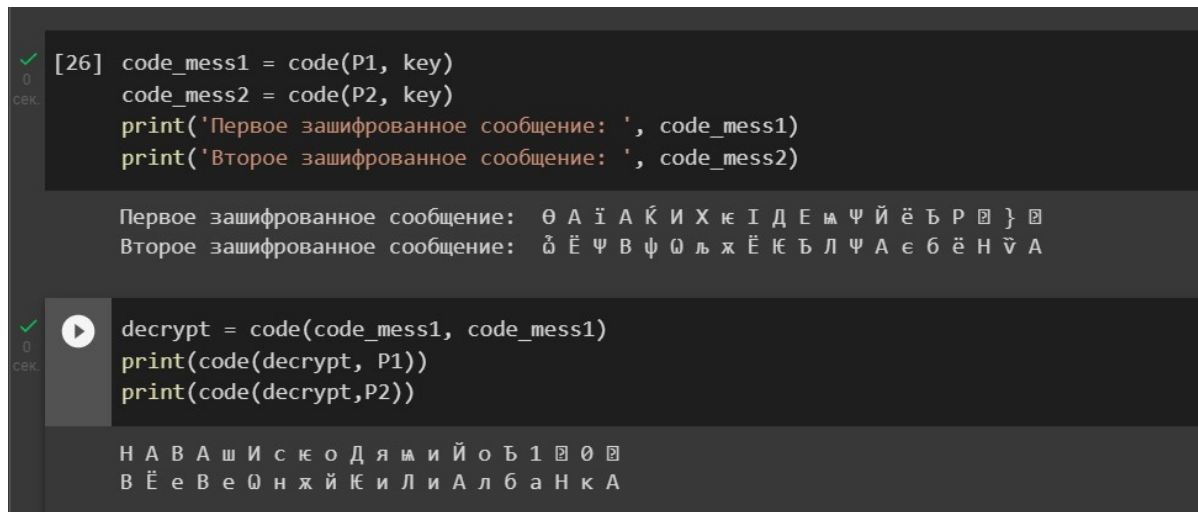
```
[9] P1 = 'НаВашисходящийот1204'
    P2 = 'ВСеверныйфилиалБанка'

[19] key = rand_key(len(P1))
    print('Ключ для шифрования сообщений: ', key)
    hex_key = shest(key)
    print('Ключ в шестнадцатеричной форме: ', hex_key)

Ключ для шифрования сообщений:  o E D d 8 Z H o a M h 5 i m 7 f 1 Y 3 M
Ключ в шестнадцатеричной форме:  6f 20 45 20 44 20 64 20 38 20 5a 20 48 20 6f 20 61 20 4d 20 68 20 35 20 69 20 6d 20 37 20 66 20 31 20 59 20 33 20 4d
```

Рис. 2.2.: Создание ключа

3. Закодировала оба сообщения с помощью ключа. Создала декриптор, использующий оба сообщения. Раскодировала сообщения при помощи него.



```
[26] code_mess1 = code(P1, key)
code_mess2 = code(P2, key)
print('Первое зашифрованное сообщение: ', code_mess1)
print('Второе зашифрованное сообщение: ', code_mess2)
```

Первое зашифрованное сообщение: 0 A ĭ A Ķ И X Ĳ Д Е ъ П Й ё Ъ Р Ѣ } Ѧ  
Второе зашифрованное сообщение: Ѣ Ё П В п Ѧ ѡ ѡ ж Ё Ĳ Ъ Л П А є б ё Н Ѣ А

```
decrypt = code(code_mess1, code_mess1)
print(code(decrypt, P1))
print(code(decrypt, P2))
```

Н А В А ш И с Ĳ о Д я ѡ и Й о Ъ 1 Ѣ 0 Ѣ  
В Ё е В е Ѧ н ж й Ĳ и Л и А л б а Н к А

Рис. 2.3.: Шифрование

### 3. Контрольные вопросы

1. Чтобы определить один из текстов, зная другой, необходимо воспользоваться следующей формулой:  $C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2$ , где  $C_1$  и  $C_2$  - шифротексты. Ключ в данной формуле не используется.
2. При повторном использовании ключа при шифровании текста получим исходное сообщение.
3. Режим шифрования однократного гаммирования одним ключом двух открытых текстов реализуется по следующей формуле:

$$C_1 = P_1 \oplus K$$

$$C_2 = P_2 \oplus K,$$

где  $C_i$  - шифротексты,  $P_i$  - открытые тексты,  $K$  - единый ключ шифровки

4. Недостатки шифрования одним ключом двух открытых текстов: Во-первых, имея на руках одно из сообщений в открытом виде и оба шифротекста, злоумышленник способен расшифровать каждое сообщение, не зная ключа. Во-вторых, зная шаблон сообщений, злоумышленник получает возможность определить те символы сообщения  $P_2$ , которые находятся на позициях известного шаблона сообщения  $P_1$ .
5. Преимущества шифрования одним ключом двух открытых текстов: Такой подход помогает упростить процесс шифрования и дешифровки. Также,



при отправке сообщений между 2-я компьютерами, удобнее пользоваться одним общим ключом для передаваемых данных

## 4. Выводы

В ходе работы мы освоили на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## **5. Список литературы**

1. Методические материалы курса