

# Отчет по лабораторной работе №7

---

Жиронкин Павел Владимирович НПИбд-01-18<sup>1</sup>

Информационная Безопасность–2021, 7 декабря, 2021, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цели и задачи работы

---

# Цель лабораторной работы

Освоить на практике применение режима однократного гаммирования.

## Задание к лабораторной работе

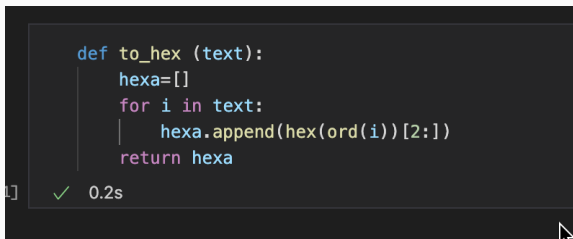
Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

# **Процесс выполнения лабораторной работы**

---

1. Написана функция *to\_hex*, трансформирующая текст в шестнадцатиричное представление (рис. 1).

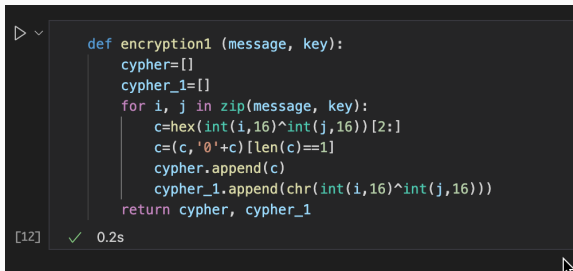


```
def to_hex (text):  
    hexa=[]  
    for i in text:  
        hexa.append(hex(ord(i))[2:])  
    return hexa
```

1] ✓ 0.2s

**Рис. 1:** Код функции *to\_hex*

2. Написана функция *encryption*, которая с помощью однократного гаммирования из сообщения и ключа получает шифротекст (рис. 2).



```
def encryption1 (message, key):  
    cypher=[]  
    cypher_1=[]  
    for i, j in zip(message, key):  
        c=hex(int(i,16)^int(j,16))[2:]  
        c=(c,'0')[len(c)==1]  
        cypher.append(c)  
        cypher_1.append(chr(int(i,16)^int(j,16)))  
    return cypher, cypher_1
```

[12] ✓ 0.2s

Рис. 2: Код функции *encryption*

3. Написана функция *gen\_key*, генерирующая случайный ключ (рис. 3).

```
from random import randrange

def gen_key (length):
    key=[]
    for _ in range(length):
        temp=randrange(256)
        temp=hex(temp)[2:]
        key.append((temp,'0'+temp)[len(temp)==1])
    return ' '.join(key)
#print(gen_key(22))
```

[13] ✓ 0.2s

Рис. 3: Код функции *gen\_key*



4. Определяю вид шифротекста при известном ключе и известном открытом тексте. Применяю к шифротексту ключ снова, чтобы получить исходное сообщение (рис. 4).

```
message='Лабораторная работа №7, Жиронкин Павел'
#key='01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01'
key=gen_key(len(message))
#key = ' '.join(isa_hex('Колесников Кирилл Владимирович'))

print('Применение ключа к исходному сообщению:\nСообщение:\t\t\t\t %s \nКлюч:\t\t\t\t %s' % (message, key))
key_m=key.split()
message_hex = to_hex(message)

cipher_hex, cypher=encryption(message_hex, key_m)
cypher=''.join(cypher)
cipher_hex=' '.join(cypher_hex)
#print('Зашифрованное сообщение:\t %s' % cypher)
print('Зашифрованное сообщение:\t %s' % cipher_hex)

print('\nПрименение ключа к зашифрованному сообщению:\nЗашифрованное сообщение:\t %s \nКлюч:\t\t\t\t %s' % (cypher_hex, key))
mess_hex, mess=encryption(cypher_hex.split(), key_m)
mess=''.join(mess)
print('Расшифрованное сообщение:\t %s' % mess)
```

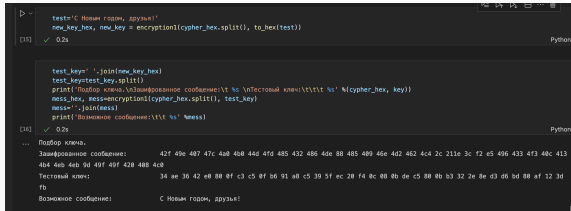
[14] ✓ 0.2s Python

... Применение ключа к исходному сообщению.  
Сообщение: Лабораторная работа №7, Жиронкин Павел  
Ключ: 34 ac 36 42 e8 88 8f c3 c5 8f b6 91 a8 c5 39 5f ec 20 f4 8c 08 0b de c5 88 0b b3 32 2e 8e d3 d6 bd 80 af 12 3d fb  
Зашифрованное сообщение: 42f 49e 487 47c 4a8 4b0 44d 4fd 485 432 486 4de 88 485 489 46e 4d2 462 4c4 2c 211e 3c f2 e5 496 433 4f3 48c 413 4b4 4eb 4eb 9d 49f 49f 428 488 4c8

Применение ключа к зашифрованному сообщению.  
Зашифрованное сообщение: 42f 49e 487 47c 4a8 4b0 44d 4fd 485 432 486 4de 88 485 489 46e 4d2 462 4c4 2c 211e 3c f2 e5 496 433 4f3 48c 413 4b4 4eb 4eb 9d 49f 49f 428 488 4c8  
Ключ: 34 ac 36 42 e8 88 8f c3 c5 8f b6 91 a8 c5 39 5f ec 20 f4 8c 08 0b de c5 88 0b b3 32 2e 8e d3 d6 bd 80 af 12 3d fb  
Расшифрованное сообщение: Лабораторная работа №7, Жиронкин Павел

Рис. 4: Получение шифротекста

5. Определяю ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!»)(рис. 5).



```
> test='С Новым годом, друзья!'
new_key_hex, new_key = encryption(cypher_hex.split(), to_hex(test))
[15] ✓ 0.2s Python

test_keys=''.join(new_key_hex)
test_key=test_key.split()
print('Подбор ключа:\nЗашифрованное сообщение:\t %s \nТестовый ключ:\t\t\t %s' % (cypher_hex, key))
mess_hex, mess=encryption(cypher_hex.split(), test_key)
mess=''.join(mess)
print('Возможное сообщение:\t\t %s' %mess)
[16] ✓ 0.2s Python

... Подбор ключа.
Зашифрованное сообщение:  42f 49e 487 47c 4a0 4b8 44d 4fd 485 432 486 4de 88 485 489 46e 4d2 462 4c4 2c 211e 3c f2 c5 496 433 4f3 48c 413
4b4 4eb 4eb 9d 49f 49f 42b 48b 4c8
Тестовый ключ:          34 ae 36 42 e8 8f c3 c5 ef b6 91 a8 c5 39 5f ec 28 f4 8c 88 0b de c5 88 0b b3 32 2e 8e d3 d6 bd 88 af 12 3d
fb
Возможное сообщение:    С Новым годом, друзья!
```

Рис. 5: Один из вариантов прочтения шифротекста

## **Выводы по проделанной работе**

---

На основе проделанной работы освоил на практике применение режима однократного гаммирования.

# **Контрольные вопросы**

---

1. Поясните смысл однократного гаммирования.
2. Перечислите недостатки однократного гаммирования.
3. Перечислите преимущества однократного гаммирования.
4. Почему длина открытого текста должна совпадать с длиной ключа?

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?
6. Как по открытому тексту и ключу получить шифротекст?
7. Как по открытому тексту и шифротексту получить ключ?
8. В чём заключаются необходимые и достаточные условия абсолютной стойкости шифра?