

# Лабораторная работа № 7. Элементы криптографии.

## Однократное гаммирование

---

Лёшьян Стефани, НФИбд-02-19

## Цель выполнения лабораторной работы

---

## Цель выполнения лабораторной работы

Освоить на практике применение режима однократного гаммирования.

```
✓  
0 [1] import numpy as np  
сек. import pandas as pd  
import sys
```

Figure 1: Импорт библиотек

Написала функцию для определения вида шифротекста при известном ключе и известном открытом тексте.

```
✓  a=" С Новым годом, друзья"
def crypt(a):
    print("open text : ",a)
    text=[]
    for i in a :
        text.append(str(i).encode("cp1251").hex())
    print("open text in 16: ", *text)
    k=np.random.randint(0, 225, len(a))
    key=[hex(i)[2:] for i in k]
    newkey=[]
    for i in k:
        newkey.append(str(i).encode("cp1251").hex().upper())
    print("key in 16: ", *key)
    b=[]
    for i in range(len(text)):
        b.append("{:02x}".format(int(key[i],16)^int(text[i],16)))
    print("cypher text in 16:", *b)
    fintext=bytearray.fromhex("".join(b)).decode("cp1251")
    print("cypher text: ", fintext)
    return key,b,fintext
```

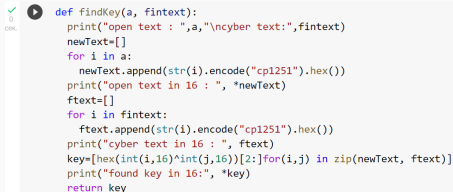
Вывод функции:

```
✓ [39] key, b, fintext=crypt(a)
0
>ENC

open text :   С Новым годом, друзья
open text in 16:  20 43 20 cd ee e2 fb ec 20 e3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff
key in 16:  d0 4d 95 a3 ba 80 d2 3d 79 44 13 62 84 b5 a0 3e ca 84 21 ac 13 19
cypher text in 16:  f0 0e b5 6e 54 62 29 d1 59 a7 fd 86 6a 59 8c 1e 2e 74 d2 4b ef e6
cypher text:  p0mTb)CŸ$ə†jŸŸ.тTKпк
```

Figure 2: Результат работы функции1

Написала функцию для определения ключа по открытому тексту и шифротексту.

A screenshot of a code editor showing a Python function named 'findKey'. The function takes two arguments: 'a' (open text) and 'fintext' (cipher text). It prints the input, then iterates over 'a' to create 'newText' by encoding each character in 'cp1251' and hex-encoding it. Then it iterates over 'fintext' to create 'ftext' by encoding each character in 'cp1251' and hex-encoding it. Finally, it calculates the key by XORing the hex values of 'newText' and 'ftext' in pairs and prints the result.

```
def findKey(a, fintext):  
    print("open text : ",a,"\ncyber text:",fintext)  
    newText=[]  
    for i in a:  
        newText.append(str(i).encode("cp1251").hex())  
    print("open text in 16 : ", *newText)  
    ftext=[]  
    for i in fintext:  
        ftext.append(str(i).encode("cp1251").hex())  
    print("cyber text in 16 : ", ftext)  
    key=[hex(int(i,16)^int(j,16))[2:]for(i,j) in zip(newText, ftext)]  
    print("found key in 16:", *key)  
    return key
```

Figure 3: Функция определения ключа

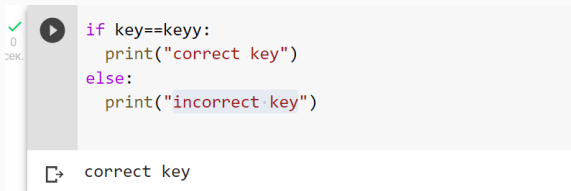
# Результат выполнения лабораторной работы

```
✓ 1 keyy=findKey(a,fintext)
000
❏ open text : С Новым годом, друзья
cyber text: рЉппТв)СУ99а'УМВ.тККк
open text in 16 : 20 43 20 cd ee a2 fb ec 20 e3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff
cyber text in 16 : ['f0', '0e', 'b5', '6e', '54', '62', '29', 'd1', '59', 'a7', 'fd', '86', '6a', '59', '8c', '1e',
found key in 16: d0 4d 95 a3 ba 80 d2 3d 79 44 13 62 84 b5 a0 3e ca 84 21 ac 13 19
```

Figure 4: Результат работы функции1



Проверяем если полученный ключ совпадает с тем, который мы получили на предыдущем шаге.



The screenshot shows a code editor with a green checkmark and a play button icon on the left. The code is a Python if-statement that checks if 'key' equals 'keyy'. If true, it prints 'correct key'. If false, it prints 'incorrect key'. Below the code, the output 'correct key' is displayed.

```
if key==keyy:  
    print("correct key")  
else:  
    print("incorrect key")
```

correct key

Figure 5: Изменили контекст файла на samba\_share\_t

## Выводы по лабораторной работе

---

Я освоила на практике применение режима однократного гаммирования.