

Лабораторная работа №7

Основы информационной безопасности

Мурашов И. В., НКАбд-03-23

16 мая 2025

Российский университет дружбы народов, Москва, Россия

- Мурашов Иван Вячеславович
- Студент, 2 курс, группа НКАбд-03-23
- Российский университет дружбы народов
- 1132236018@rudn.ru
- <https://neve7mind.github.io>

Освоить на практике применение режима однократного гаммирования

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

Я выполнил лабораторную работу на языке программирования Python, листинг программы и результаты выполнения приведены в отчете.

Выполнение лабораторной работы

Требуется разработать программу, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Начнем с создания функции для генерации случайного ключа

```
import random
import string

def generate_key_hex(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits) #генерация цифры для каждого символа в тексте
    return key
```

Рис. 1: Функция генерации ключа

Необходимо определить вид шифротекста при известном ключе и известном открытом тексте. Так как операция исключающего или отменяет сама себя, делаю одну функцию и для шифрования и для дешифрования текста

```
#для шифрования и дешифрования  
def en_de_crypt(text, key):  
    new_text = ''  
    for i in range(len(text)): #проход по каждому символу в тексте  
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))  
    return new_text
```

Рис. 2: Функция для шифрования текста

Выполнение лабораторной работы

Нужно определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Для этого создаю функцию для нахождения возможных ключей для фрагмента текста.

```
def find_possible_key(text, fragment):  
    possible_keys = []  
    for i in range(len(text) - len(fragment) + 1):  
        possible_key = ""  
        for j in range(len(fragment)):  
            possible_key += chr(ord(text[i + j]) ^ ord(fragment[j]))  
        possible_keys.append(possible_key)  
    return possible_keys
```

Рис. 3: Подбор возможных ключей для фрагмента

Выполнение лабораторной работы

Проверка работы всех функций. Шифрование и дешифрование происходит верно, как и нахождение ключей, с помощью которых можно расшифровать верно только кусок текста

```
t = 'С Новым Годом, друзья!'
key = generate_key_hex(t)
en_t = en_de_crypt(t, key)
de_t = en_de_crypt(en_t, key)
keys_t_f = find_possible_key(en_t, 'С Новым')
fragment = "С Новым"
print('Открытый текст: ', t, "\nКлюч: ", key, "\nШифротекст: ", en_t, "\nИсходный текст: ", de_t,)

print('Возможные ключи: ', keys_t_f)
print('Расшифрованный фрагмент: ', en_de_crypt(en_t, keys_t_f[0]))
```

Открытый текст: С Новым Годом, друзья!
Ключ: ApV5aBX756a1R3P1xoVYII
Шифротекст: 6Pyh/MEDU2sU3fрйивЕIh
Исходный текст: С Новым Годом, друзья!
Возможные ключи: ['ApV5aBX', 'фж\x16m;И', 'jбИ7VК\x1a', 'ф\x14Zxm4', 'rщyЦ\x14C1', '{ф\x18:\x1e3', 'E7;6gDR', 'жI\x15k-%', '\x07И1\\3ь', 'v\x12Pел
а', 'tЯsjt\x16\x04', '.юооos\x10', 'OFжс\ngj', 'чР@\x06\x1e*)', 'ёёё\x125^:', '[И1_Мe"]
Расшифрованный фрагмент: С НовымVi9Q3b>193ЙEi9)

Рис. 4: Результат работы программы

В ходе выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования.