

Лабораторная работа №7

Основы информационной безопасности

Павлюченков С.В.

07 сентября 25

Российский университет дружбы народов, Москва, Россия

- Павлюченков Сергей Витальевич
- Студент ФФМиЕН
- Российский университет дружбы народов
- 1132237372@pfur.ru
- <https://serapshi.github.io/svpavliuchenkov.github.io/>



Освоить на практике применение режима однократного гаммирования¹

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.

Выполнение лабораторной работы

Функции на Python

```
ды + Код + Текст ▶ Выполнить все ▼

import string
import random as rand

def get_key(text_len):
    return ''.join(rand.choice(string.ascii_letters + string.digits) for _ in range(text_len))

def xor_crypt(text, key):
    res = []
    len_key = len(key)
    for i, ch in enumerate(text):
        res.append(chr(ord(ch) ^ ord(key[i%len_key])))
    return ''.join(res)

def find_key(message, open):
    poss_key = []
    max_depth = len(message) - len(open) + 1
    for i in range(max_depth):
        var = []
        for j in range(len(open)):
            var.append(chr(ord(message[i+j]) ^ ord(open[j])))
        poss_key.append(''.join(var))
    return poss_key

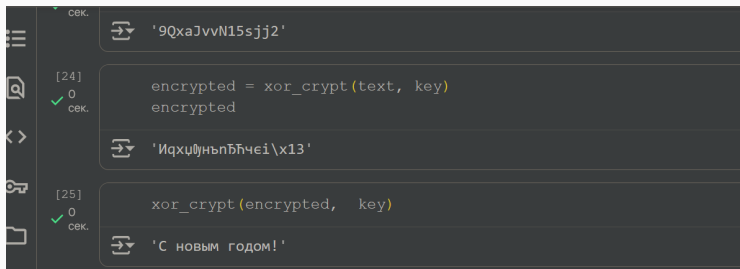
text = 'С новым годом!'
key = get_key(len(text))

'6F35kHyv896uu8'
```

переменные Терминал

```
Python import string import random as rand def get_key(text_len): return  
"".join(rand.choice(string.ascii_letters + string.digits) for _ in range(text_len))  
  
def xor_crypt(text, key): res = [] len_key = len(key) for i, ch in enumerate(text):  
res.append(chr(ord(ch) ^ ord(key[i%len_key]))) return "".join(res)  
  
def find_key(message, open): poss_key = [] max_depth = len(message) - len(open) + 1 for i in  
range(max_depth): var = [] for j in range(len(open)):  
var.append(chr(ord(message[i+j])^ord(open[j]))) poss_key.append("".join(var)) return poss_key
```

Тут показано возможность и работоспособность обратного дешифрования при известном ключе (рис. (fig:002?)).



The screenshot shows a code editor with a dark theme. On the left, there is a sidebar with icons for a menu, a document, navigation arrows, a key, and a folder. The main editor area contains three rows of code, each with a swap icon (two arrows) to its left. The first row has a green checkmark and 'сек.' in the left margin, followed by the string `'9QxaJvvN15sjj2'`. The second row has a green checkmark, '[24]', '0', and 'сек.' in the left margin, followed by the code `encrypted = xor_crypt(text, key)` and `encrypted`. The third row has a green checkmark, '[25]', '0', and 'сек.' in the left margin, followed by the code `xor_crypt(encrypted, key)` and the string `'С новым годом!'`.

```
'9QxaJvvN15sjj2'
```

```
[24] 0 сек.  
encrypted = xor_crypt(text, key)  
encrypted
```

```
[25] 0 сек.  
xor_crypt(encrypted, key)  
'С новым годом!'
```

Рис. 2: Пример применения

В этой работе я освоил на практике применение режима однократного гаммирования.