

# Лабораторная работа №7. Элементы криптографии.

## Однократное гаммирование

---

Выполнила: Лебедева Ольга Андреевна

Преподаватель Кулябов Дмитрий Сергеевич д.ф.-м.н., профессор кафедры прикладной информатики и кибербезопасности

2024

Российский университет дружбы народов, Москва, Россия

Освоить на практике применение режима однократного гаммирования.

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком  $\oplus$ ) между элементами гаммы и элементами подлежащего сокрытию текста[1].

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.

Код лабораторной работы:

```
import random
```

```
def generate_random_key(text):  
    possible_symbol = list(range(32, 127)) + list(range(1  
    key_str = ''.join(chr(random.choice(possible_symbol))  
    return key_str
```

```
def xor(text, key):  
    return [ord(s1) ^ ord(s2) for s1, s2 in zip(text, key
```

```
def encrypt(text, key):  
    chiphr = xor(text, key)  
    chiphrotext = ''.join(chr(i) for i in chiphr)  
    return chiphrotext  
  
def decrypt(chiphro, key):  
    decrypted = xor(chiphro, key)  
    opentext = ''.join(chr(i) for i in decrypted)  
    return opentext
```

```
def find_key(chiphrotext, text_fragment):  
    chipher_fragment = chiphrotext[:len(chiphrotext)]  
    key_f = xor(text_fragment, chipher_fragment)  
    found_key = ''.join(chr(i) for i in key_f)  
    return found_key  
  
text = "С Новым Годом, друзья!"  
text_fragment = "С Новым"  
key = generate_random_key(text)  
print("Созданный ключ: ", key)  
chiphrotext = encrypt(text, key)  
print('Открытый текст: ', text)
```

```
print('Зашифрованный текст: ', chiphrotext)
opentext = decrypt(chiphrotext, key)
print('Расшифрованный текст: ', opentext)
found_key = find_key(chiphrotext, text_fragment)
open_fragtext = decrypt(chiphrotext[:len(text_fragment)],
print('Один из возможных вариантов прочтения текста по фр
```



Эта программа реализует шифрование и дешифрование текста с использованием симметричного шифра на основе операции XOR. Сначала генерируется случайный ключ той же длины, что и исходный текст, выбирая случайные символы из определенного диапазона (включая латинские и кириллические символы). Затем текст шифруется с использованием операции XOR между символами исходного текста и ключа, что приводит к зашифрованному тексту. Дешифрование осуществляется аналогично: выполняется обратная операция XOR между зашифрованным текстом и ключом, что позволяет восстановить исходный текст. Дополнительно программа включает функцию для нахождения ключа по фрагменту исходного текста и соответствующему фрагменту зашифрованного текста, что демонстрирует возможность частичной дешифровки текста.

Результат работы кода: См. рис. 1

```
Созданный ключ: чИГГ ажbM$9щ2\MuYщRVdc
Открытый текст: С Новым Годом, друзья!
Зашифрованный текст: фи0-ВЪ
В0КйиЎрмссјѐКЫѡ
Расшифрованный текст: С Новым Годом, друзья!
Один из возможных вариантов прочтения текста по фрагменту С НовымВ0КйиЎрмссјѐКЫѡ
```

**Рис. 1:** Результат работы кода

Освоили на практике применение режима однократного гаммирования.

[1] Гаммирование: [https://www.researchgate.net/profile/Dmitry-Kulyabov/publication/339290917\\_Informacionnaa\\_bezopasnost\\_komputernyh\\_setej\\_laboratornyh\\_bezopasnost-komputernyh-setej-laboratornye-raboty.pdf](https://www.researchgate.net/profile/Dmitry-Kulyabov/publication/339290917_Informacionnaa_bezopasnost_komputernyh_setej_laboratornyh_bezopasnost-komputernyh-setej-laboratornye-raboty.pdf)