# Лабораторная работа № 7

**Элементы криптографии. Однократное гаммирование**

Тарусов Артём Сергеевич

# Содержание

# Список иллюстраций

# Цель работы

Освоить на практике применение режима однократного гаммирования.

# Задание

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.

# Теоретическое введение

- Шифрование – это технология кодирования и раскодирования данных.Зашифрованные данные -это результат применения алгоритма для кодирования данных с целью сделать их недоступными для чтения. Данные могут быть раскодированы в исходную форму только путем применения специальный ключа. [1].
- Гаммирование — это наложение (или снятие при расшифровке сообщений) на открытое (или зашифрованное) сообщение так называемой криптографической гаммы. Криптографическая гамма — это последовательность элементов данных, которая вырабатывается с помощью определенного алгоритма. [2].

# Выполнение лабораторной работы

1. Импортируем необходимые модули (fig. 1).

```
import string
import random
```

Рис. 1: Импорт

2. Создадим функцию для преобразования данных в шестнадцатеричный формат (fig. 2).

```
def toHex(text):
    return " ".join(hex(ord(i))[2:] for i in text)
```

Рис. 2: Функция toHex

3. Напишем функцию, генерирующую ключ (fig. 3).

```
def gen_key(size):
    key = "".join(random.choice(string.ascii_letters + string.digits) for _ in range(size))
    return key
```

Рис. 3: Функция gen_key

4. Реализуем функцию для кодирования и декодирования данных (fig. 4).

```
def encoder(text, key):
    return "".join(chr(a^b) for a, b in zip(text, key))
```

Рис. 4: Функция encoder

5. Закодируем и декодируем строку “С Новым годом, друзья!” (fig. 5).

```
msg = "С Новым годом, друзья!"
key = gen_key(len(msg))
hex_key = toHex(key)
print("Ключ: ", hex_key)
enc_text = encoder([ord(i) for i in msg], [ord(i) for i in key])
hex_text = toHex(enc_text)
print("Зашифрованное сообщение: ", hex_text)
decr_text = encoder([ord(i) for i in enc_text], [ord(i) for i in key])
print("Расшифрованный текст: ", decr_text)
```

```
Ключ:  37 75 70 41 76 41 5a 50 4c 73 4a 52 4c 44 33 71 72 41 72 79 62 77
Зашифрованное сообщение:  416 55 46d 47f 444 40a 466 70 47f 44d 47e 46c 470 68 13 445 432 402 445 435
42d 56
Расшифрованный текст:  С Новым годом, друзья!
```

Рис. 5: Кодирование и декодирование строки

6. Получим ключ, с помощью которого получим сообщение “С Новым годом, коллега”, вместо “С Новым годом, друзья!” при декодировании. Воспользуемся симметричностью кодирования(fig. 6).

```
new_msg = "С Новым годом, коллега"

key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_msg])
print("Ключ: ", toHex(key))
```

```
Ключ:  37 75 70 41 76 41 5a 50 4c 73 4a 52 4c 44 33 7f c 39 7e 0 1e 466
```

Рис. 6: Получение ключа для другого прочтения открытого текста

# Выводы

В рамках данной лабораторной работы было освоено на практике применение режима однократного гаммирования.

# Список литературы

[1] https://www.kaspersky.ru/resource-center/definitions/encryption

[2] https://xakep.ru/2019/07/18/crypto-xor/