# Лабораторная работа №8

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Буллет Т. А.

16 февраля 2025

Российский университет дружбы народов, Москва, Россия

# Информация

# Докладчик

- Буллер Татьяна Александровна
- студент направления Бизнес-информатика
- Российский университет дружбы народов

# Вводная часть

# Объект и предмет исследования

- Метод шифрования XOR
- Среда виртуализации VirtualBox

# Цели и задачи

- Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

# Материалы и методы

- Процессор pandoc для входного формата Markdown
- Среда виртуализации VirtualBox
- Язык программирования Python

# Выполнение лабораторной работы

# Генерация ключа

В отличие от предыдущей работы, закодированные сообщения пользователь может выбрать сам. Поэтому для упрощения реализации возьмем стандартную кодировку ASCII, известную без дополнительных включений всем языкам программирования. Для обоих шифротекстов необходимо сгенерировать ключ. Помним, что для обеспечения шифру адекватной криптостойкости, длина ключа должна быть не менее длины шифротекста. Напишем функцию, которая по переданной ей длине текста сгенерирует для него случайную строку ключа шифрования:

```python
import random
import string

# сгенерировать ключ ascii
def key_hex(n):
    key = ''
    for i in range(n):
        key += random.choice(string.ascii_letters + string.digits)
    return key

```

**Рис. 1:** Функция генерацияи ключа

# Функция операции XOR для двух строк

Далее необходимо было написать код, с помощью которого можно было бы зашифровать сообщение. Для этого напишем простую функцию, которая в цикле по каждому символу исходного текста и соответствующему символу ключа сгенерирует символ нового текста.

```
# зашифровать по ключу
def cipher(plain, key):
    new_text = ''
    for i in range(len(plain)):
        new_text += chr(ord(plain[i]) ^ ord(key[i % len(key)]))
    return new_text
```

**Рис. 2:** Функция шифрования

# Тело программы

В теле программы зададим две строки открытого текста одинаковой длины (plain1 и plain2), взяв за текст стандартную рыбу Lorem Ipsum. По одному из текстов (в представленном примере - по первому) сгенерируем ключ с помощью написанной ранее функции, затем продемонстрируем шифрование строки этим ключом и дешифрование полученных шифротекстов.

```
plain1 = 'Neque porro quisquam est qui dolorem' # открытый текст 1
key = key_hex(len(plain1)) # сгенерировать ключ для обоих текстов
cipher1 = cipher(plain1, key) # шифротекст 1
deciphered1 = cipher(cipher1, key) # расшифровать шифротекст 1 по ключу

plain2 = 'Fusce eu venenatis erat. Aliquam sed' # открытый текст 2
cipher2 = cipher(plain2, key) # шифротекст 2
deciphered2 = cipher(cipher2, key) # расшифровать шифротекст 2 по ключу

# человекочитаемый вывод

print('оригинальный текст 1: ', plain1, "\nсгенерированный ключ: ", key,
'\nшифротекст: ', cipher1, '\nрасшифровка: ', deciphered1,'\n')
print('оригинальный текст: ', plain2, "\nсгенерированный ключ: ", key,
'\nшифротекст: ', cipher2, '\nрасшифровка: ', deciphered2,'\n')
```

**Рис. 3:** Тело программы: открытый текст, генерация ключа, шифротекст и расшифровка по ключу

# Расшифровать один текст через второй

В случае, когда мы имеем два сообщения одинаковой длины и известно, что они были зашифрованы одним ключом, мы можем расшифровать одно сообщение через второе. Для этого проведем простую операцию: XOR для двух шифротекстов. Это позволит получить ключ шифрования, который мог бы быть использован, чтобы операцией XOR на одном из открытых текстов получить второй. Владея одним из открытых текстов и этим ключом, мы можем получить второй открытый текст.

```
xored = cipher(cipher2, cipher1) # проксорить второй через первый
print('второй по первому: ', cipher(plain1, xored))
print('первый по второму: ', cipher(plain2, xored))
```

**Рис. 4:** Расшифровка одного текста через второй

# Вывод программы

Человекочитаемый вывод программы состоит из трех частей: операции над первым текстом (сам открытый текст, сгенерированный ключ, шифротекст и открытый текст, полученный от шифротекста и ключа), аналогичные операции для второго текста и получение открытых сообщений по двум шифротекстам и одному открытому.

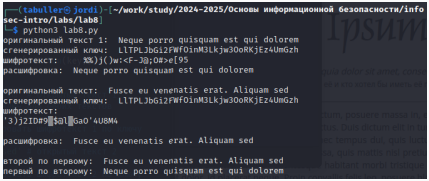


**Рис. 5:** Вывод программы

# Выводы

# Выводы

Было освоено на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом, написана программа, шифрующая две строки одним случайно сгенерированным ключом и дешифрующая шифротексты обратно по одному открытому тексту и двум шифрам.