

# Прагматика выполнения лабораторной работы

Студенты должны разбираться в методах шифрования и познакомиться способом шифрования текста гаммированием. Все это необходимо для общего понимания методов шифрования и повышения безопасности в системах.

## Цель выполнения лабораторной работы

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

## Задачи выполнения лабораторной работы

1. Разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования.
2. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе
3. Определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить.

## Результаты выполнения лабораторной работы

1. Разработала приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования.

(рис. -@fig:001)

### Однократное гаммирование

```
Звод [37]: import random as rnd
import string as strg

p1 = "НаВашисходящийот1204"
p2 = "ВЮхныйфилиалБанка"
k = "".join(rnd.choice(strg.ascii_letters.join(strg.digits)) for i in range(len(p1)))

Звод [38]: #функция сложение по модулю 2
def xor_string(data, key):
    return "".join(chr(ord(x)^ord(y)) for x, y in zip(data, key))

#функция преобразования в байты
def butes_print(x, k):
    c = xor_string(x, k)
    print(bytes(c, "UTF-8").hex())
    return c

Звод [39]: c1 = butes_print(p1, k)
c2 = butes_print(p2, k)

d19bd193d185d1a9d09fd19cd0a9d092d19cd1b8d0a1d09fd1b3d18ed1abd0b45348045b
d194d18dd1a1d1a4d09cd19dd195d09cd19bd088d196d1add1b3d187d1aed1a7d192d187d08ed19f

Звод [40]: c1_c2 = butes_print(c1, c2)

0f1e240d03017c0e0770777200090553d081d08fd08ad084

Звод [41]: found = xor_string(c1_c2, p1)
print(found)

ВЮхныйфилиалБанка
```

Приложение написано на python 3. Я запускала его через jupyter Notebook. В данном коде имеется 2 основные функции. 1 - сложение по модулю 2, 2 - представление в байтовом виде.

## Результаты выполнения лабораторной работы

2. Определим вид шифро-текстов C1 и C2 обоих текстов P1 и P2 при известном ключе. Для того запустим данную часть кода в нашем ноутбуке.

(рис. -@fig:002)

```
Ввод [39]: c1 = bytes_print(p1, k)
           c2 = bytes_print(p2, k)

           d19bd193d185d1a9d09fd19cd0a9d092d19cd1b8d0a1d09fd1b3d18ed1abd0b45348045b
           d194d18dd1a1d1a4d09cd19dd195d09cd19bd088d196d1add1b3d187d1aed1a7d192d187d08ed19f
```

С помощью функции `byte_print()` шифруем оба текста `p1` и `p2` с помощью одного заранее созданного ключа `k`. `byte_print()` вызывает `xor_string()`, которая складывает два предложения по модулю 2. Сохраняем полученные зашифрованные тексты в переменные `c1` и `c2`. Далее функция представляет их в буквенном виде и показывает нам.

## Результаты выполнения лабораторной работы

3. Далее воспользуемся способом, который даст нам разгадать оба зашифрованных текста, без использования нашего ключа.

Следуем дальше этой схеме, представленной в инструкции к лабораторной работе.

(рис. -@fig:003)

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2.$$

Складываем 2 зашифрованных текста по модулю 2. Получившийся результат складываем с одним из расшифрованных текстов `p1`. Выводим получившийся результат сложения `c1`, `c2` и `p1`. У нас получается расшифрованный текст `p2`, который мы сохраняем. Теперь уже получившийся расшифрованный текст складываем по модулю 2 с `c1` и `c2`. Теперь же у нас получается расшифрованный `p1`.

(рис. -@fig:004)

```
c1_c2 = bytes_print(c1, c2)

0f1e240d03017c0e0770777200090553d081d08fd08ad084
```

```
found = xor_string(c1_c2, p1)
print(found)
```

Важный филиал Банка

```
found2 = xor_string(c1_c2, found)
print(found2)
```

На Вашемходящийот1204

Т.е. мы можем расшифровать любой текст, если у нас имеется два зашифрованных текста одной гаммой, и один расшифрованный текст. Т. е. нам даже не нужно знать шифровальный ключ для данных операций.

## Вывод

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

**{.standout}Спасибо за внимание**

