Отчет к лабораторной работе №7

# Common information

discipline: Основы информационной безопасности group: НПМбд-02-21

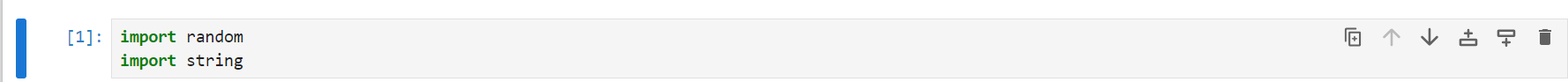
author: Бабина Ю. О.

# Цель работы

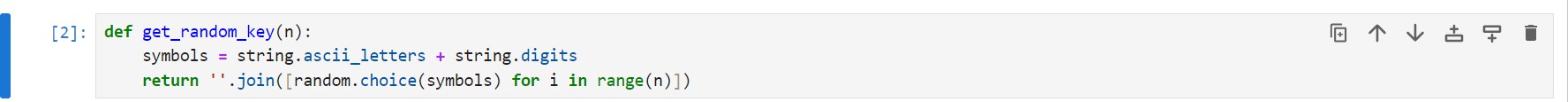
Освоить на практике применение режима однократного гаммирования.

# Выполнение работы

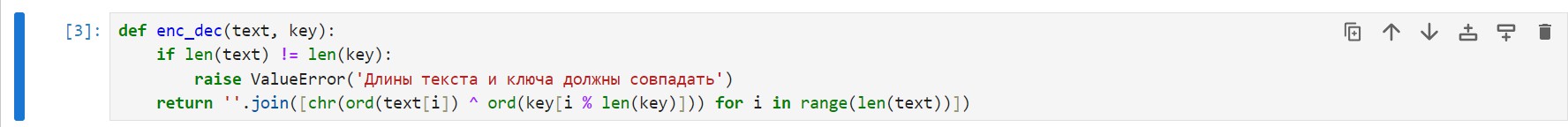
Напишем код на языке программирования Python. Импортируем все необходимые библиотеки:



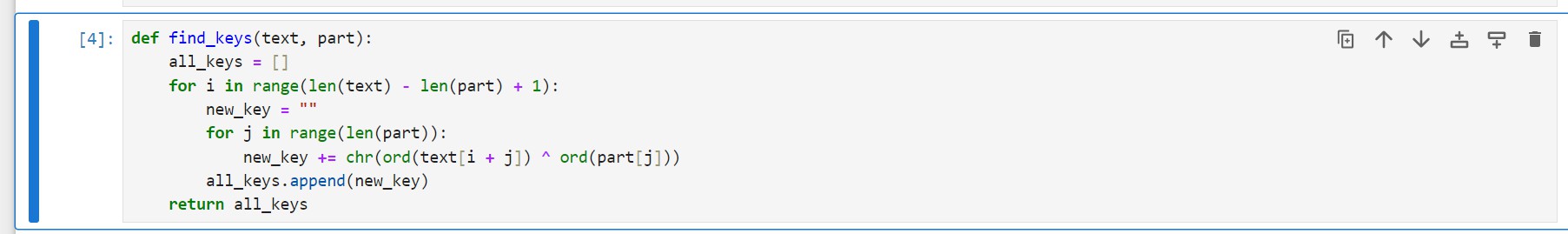
Напишем функцию для генерации случайного ключа заданной длины:



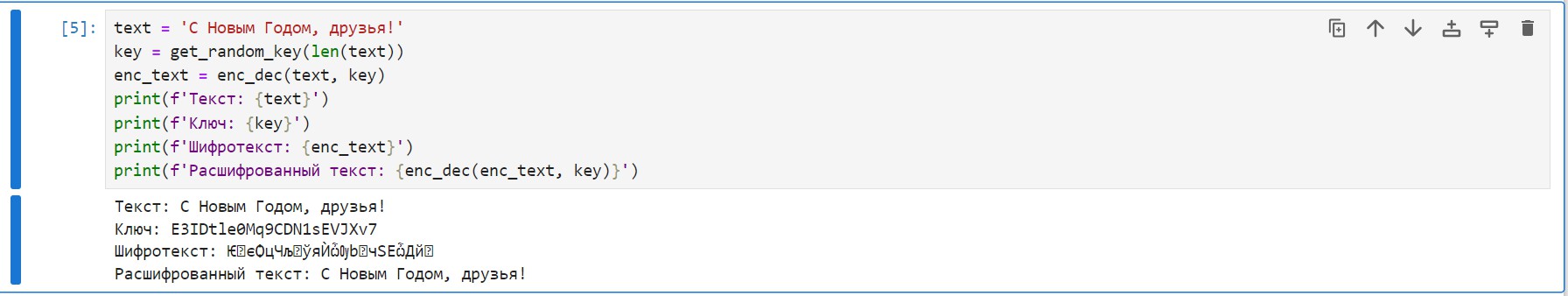
Далее реализуем функцию для шифрования и дешифрования текста по заданному ключу (функция будет одна для двух процессов, так как операция исключающего или отменяет сама себя):



Теперь реализуем функцию, которая подбирает ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста:



Проверим корректность работы написанных нами функций:





В итоге имеем данную программу:

import random import string

def get\_random\_key(n):

symbols = string.ascii\_letters + string.digits

return ''.join([random.choice(symbols) for i in range(n)])

def enc\_dec(text, key):

if len(text) != len(key):

raise ValueError('Длины текста и ключа должны совпадать')

return ''.join([chr(ord(text[i]) ^ ord(key[i % len(key)])) for i in range(len(text))])

def find\_keys(text, part): all\_keys = []

for i in range(len(text) - len(part) + 1): new\_key = ""

for j in range(len(part)):

new\_key += chr(ord(text[i + j]) ^ ord(part[j])) all\_keys.append(new\_key)

return all\_keys

text = 'С Новым Годом, друзья!' key = get\_random\_key(len(text)) enc\_text = enc\_dec(text, key)

print(f'Текст: {text}')

print(f'Ключ: {key}')

print(f'Шифротекст: {enc\_text}')

print(f'Расшифрованный текст: {enc\_dec(enc\_text, key)}')

part='С Новым'

potential\_keys = find\_keys(text, part)

print(f'Список ключей: {potential\_keys}')

# Контрольные вопросы

1. Однократное гаммирование-это метод шифрования, при котором каждый символ открытого текста гаммируется с соответствующим символом ключа только один раз.
2. Недостатки однократного гаммирования:

 Уязвимость к частотному анализу из-за сохранения частоты символов открытого текста в шифротексте.

 Необходимость использования одноразового ключа, который должен быть длиннее самого открытого текста.

 Нет возможности использовать один ключ для шифрования разных сообщений.

1. Преимущества однократного гаммирования:

 Высокая стойкость при правильном использовании случайного ключа.  Простота реализации алгоритма.

 Возможность использования случайного ключа.

1. Длина открытого текста должна совпадать с длиной ключа, чтобы каждый символ открытого текста гаммировался с соответствующим символом ключа.
2. В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ), которая объединяет двоичные значения символов открытого текста и ключа для получения шифротекста. Особенность XOR - если один из битов равен 1, то результат будет 1, иначе 0.
3. Для получения шифротекста по открытому тексту и ключу каждый символ открытого текста гаммируется с соответствующим символом ключа с помощью операции XOR.
4. По открытому тексту и шифротексту невозможно восстановить действительный ключ, так как для этого нужна информация о каждом символе ключа.
5. Необходимые и достаточные условия абсолютной стойкости шифра:

 Ключи должны быть случайными и использоваться только один раз.  Длина ключа должна быть не менее длины самого открытого текста.

 Ключи должны быть храниться и передаваться безопасным способом.

# Вывод

В рамках выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования.

# Список литературы

 https://bugtraq.ru/library/books/crypto/chapter7/  [https://www.youtube.com/watch?v=tAjBUL](http://www.youtube.com/watch?v=tAjBULW_OjQ)W\_OjQ