Отчёт по лабораторной работе №7

Дисциплина: Информационная безопасность

Андрианова Марина Георгиевна

Содержание

[Цель работы 1](#_Toc179900014)

[Выполнение лабораторной работы 1](#_Toc179900015)

[Ответы на контрольные вопросы 3](#_Toc179900016)

[Выводы 4](#_Toc179900017)

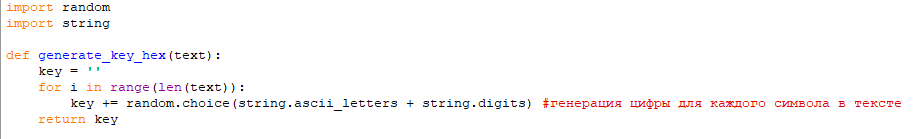
# Цель работы

Освоить на практике применение режима однократного гаммирования.

# Выполнение лабораторной работы

Я выполняла лабораторную работу на языке программирования Python, листинг программы и результаты выполнения приведены в отчете.

Требуется разработать программу, позволяющую шифровать и дешифровать данные в режиме однократного гаммирования. Начнем с создания функции для генерации случайного ключа (рис. [-@fig:001]).



Функция генерации ключа

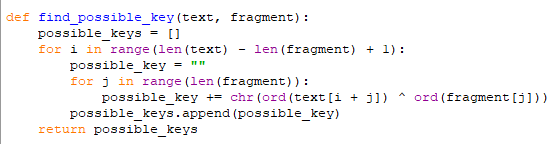
Необходимо определить вид шифротекста при известном ключе и известном открытом тексте. Делаю одну функцию и для шифрования, и для дешифрования текста (рис. [-@fig:002]).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Функция для шифрования текста

Нужно определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Для этого создаю функцию для нахождения возможных ключей для фрагмента текста (рис. [-@fig:003]).



Подбор возможных ключей для фрагмента

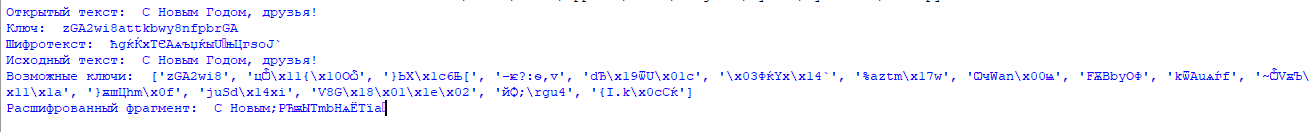
В следующей части кода реализуем шифрование и дешифрование текста, а также поиск возможных ключей для расшифровки (рис. [-@fig:004]).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Шифрование и дешифрование текста

Проверка работы всех функций. Шифрование и дешифрование происходит верно, как и нахождение ключей, с помощью которых можно расшифровать верно только кусок текста (рис. [-@fig:005]).



Результат работы программы

Листинг программы 1:

import random  
import string  
  
def generate\_key\_hex(text):  
 key = ''  
 for i in range(len(text)):  
 key += random.choice(string.ascii\_letters + string.digits) #генерация цифры для каждого символа в тексте  
 return key  
  
#для шифрования и дешифрования  
def en\_de\_crypt(text, key):  
 new\_text = ''  
 for i in range(len(text)): #проход по каждому символу в тексте  
 new\_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))  
 return new\_text  
  
def find\_possible\_key(text, fragment):  
 possible\_keys = []  
 for i in range(len(text) - len(fragment) + 1):  
 possible\_key = ""  
 for j in range(len(fragment)):  
 possible\_key += chr(ord(text[i + j]) ^ ord(fragment[j]))  
 possible\_keys.append(possible\_key)  
 return possible\_keys  
  
t = 'С Новым Годом, друзья!'  
key = generate\_key\_hex(t)  
en\_t = en\_de\_crypt(t, key)  
de\_t = en\_de\_crypt(en\_t, key)  
keys\_t\_f = find\_possible\_key(en\_t, 'С Новым')  
fragment = "С Новым"  
print('Открытый текст: ', t, "\nКлюч: ", key, '\nШифротекст: ', en\_t, '\nИсходный текст: ', de\_t,)  
  
print('Возможные ключи: ', keys\_t\_f)  
print('Расшифрованный фрагмент: ', en\_de\_crypt(en\_t, keys\_t\_f[0]))

# Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования. - Однократное гаммирование - это метод шифрования, при котором каждый символ открытого текста гаммируется с соответствующим символом ключа только один раз.
2. Перечислите недостатки однократного гаммирования. - 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. Перечислите преимущества однократного гаммирования. - Преимущества однократного гаммирования:

* Высокая стойкость при правильном использовании случайного ключа.
* Простота реализации алгоритма.
* Возможность использования случайного ключа.

1. Почему длина открытого текста должна совпадать с длиной ключа? - Длина открытого текста должна совпадать с длиной ключа, чтобы каждый символ открытого текста гаммировался с соответствующим символом ключа.
2. Какая операция используется в режиме однократного гаммирования, назовите её особенности? - 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. Как по открытому тексту и ключу получить шифротекст? - Для получения шифротекста по открытому тексту и ключу каждый символ открытого текста гаммируется с соответствующим символом ключа с помощью операции XOR.
4. Как по открытому тексту и шифротексту получить ключ? - По открытому тексту и шифротексту невозможно восстановить действительный ключ, так как для этого нужна информация о каждом символе ключа.
5. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра - 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.

# Выводы

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