Отчёт по лабораторной работе №8

Дисциплина: Информационная безопасность

Андрианова Марина Георгиевна

Содержание

[Цель работы 1](#_Toc179909274)

[Выполнение лабораторной работы 1](#_Toc179909275)

[Ответы на контрольные вопросы 3](#_Toc179909276)

[Выводы 3](#_Toc179909277)

# Цель работы

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

# Выполнение лабораторной работы

Я выполняла лабораторную работу на языке программирования Python, используя функции, реализованные в лабораторной работе №7.

Используя функцию для генерации ключа, генерирую ключ, затем шифрую два разных текста одним и тем же ключом (рис. [-@fig:001]).

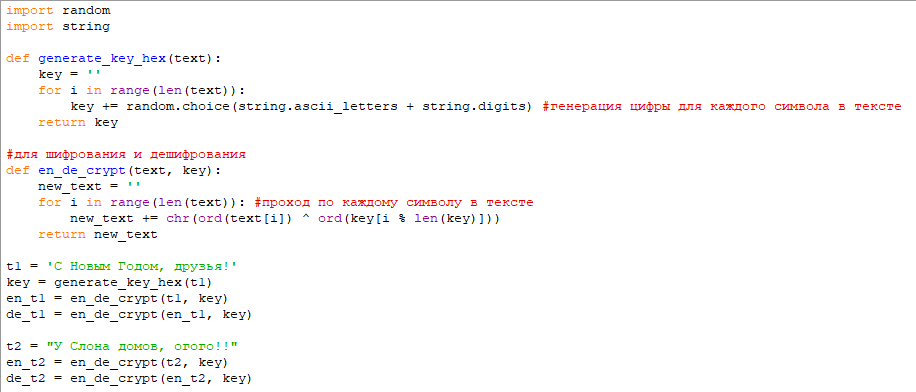


Рисунок 1. Шифрование двух текстов

Расшифровываю оба текста сначала с помощью одного ключа, затем предполагаю, что мне неизвестен ключ, но известен один из текстов и уже расшифровываю второй, зная шифротексты и первый текст (рис. [-@fig:002]).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2. Расшифровывание двух текстов

Запускаю написанный код (рис. [-@fig:003]).

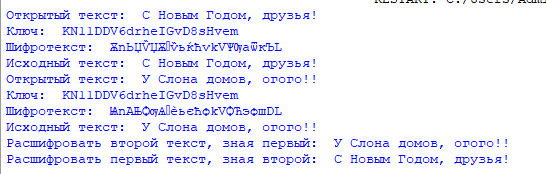


Рисунок 3. Результат работы программы

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  
  
t1 = 'С Новым Годом, друзья!'  
key = generate\_key\_hex(t1)  
en\_t1 = en\_de\_crypt(t1, key)  
de\_t1 = en\_de\_crypt(en\_t1, key)  
  
t2 = "У Слона домов, огого!!"  
en\_t2 = en\_de\_crypt(t2, key)  
de\_t2 = en\_de\_crypt(en\_t2, key)  
  
print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', en\_t1, '\nИсходный текст: ', de\_t1,)  
print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', en\_t2, '\nИсходный текст: ', de\_t2,)  
  
r = en\_de\_crypt(en\_t2, en\_t1) #С1^C2  
print('Расшифровать второй текст, зная первый: ', en\_de\_crypt(t1, r))  
print('Расшифровать первый текст, зная второй: ', en\_de\_crypt(t2, r))

# Ответы на контрольные вопросы

1. Как, зная один из текстов ( или ), определить другой, не зная при этом ключа? - Для определения другого текста () можно просто взять зашифрованные тексты , далее применить XOR к ним и к известному тексту: .
2. Что будет при повторном использовании ключа при шифровании текста? - При повторном использовании ключа мы получим дешифрованный текст.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? - Режим шифрования однократного гаммирования одним ключом двух открытых текстов осуществляется путем XOR-ирования каждого бита первого текста с соответствующим битом ключа или второго текста.
4. Перечислите недостатки шифрования одним ключом двух открытых текстов - Недостатки шифрования одним ключом двух открытых текстов включают возможность раскрытия ключа или текстов при известном открытом тексте.
5. Перечислите преимущества шифрования одним ключом двух открытых текстов - Преимущества шифрования одним ключом двух открытых текстов включают использование одного ключа для зашифрования нескольких сообщений без необходимости создания нового ключа и выделения на него памяти.

# Выводы

В ходе лабораторной работы я освоила на практике навыки применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.