Отчет по лабораторной работе №8

Дисциплина: Основы информационной безопасности

Иванов Сергей Владимирович

Содержание

# 1 Цель работы

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

# 2 Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты и в режиме однократного гаммирования. Приложение должно определить вид шифротекстов и обоих текстов и P2 при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

# 3 Выполнение лабораторной работы

Я выполнял 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. 1).

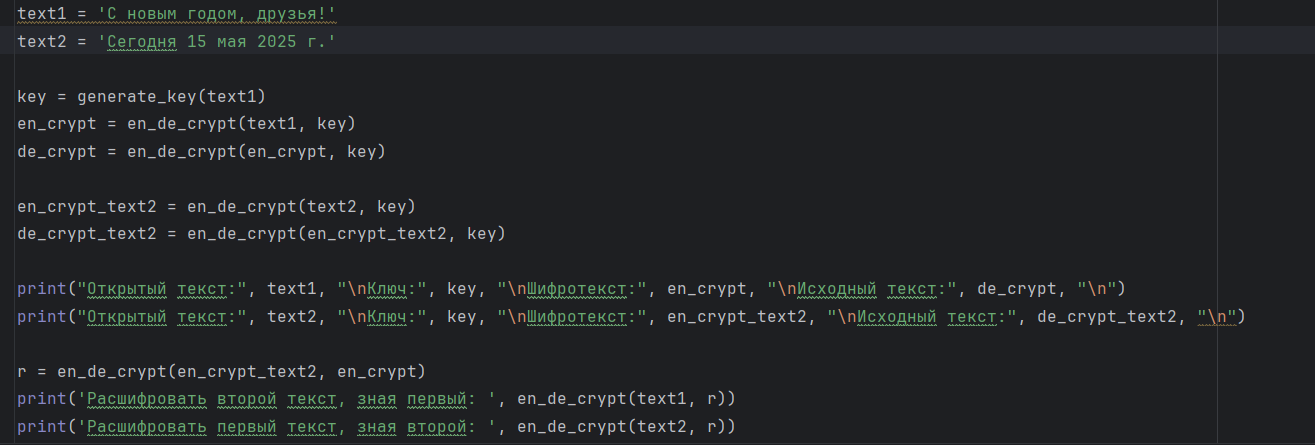


Рис. 1: Шифрование двух текстов

Расшифровываю оба текста сначала с помощью одного ключа, затем предполагаю, что мне неизвестен ключ, но извествен один из текстов и уже расшифровываю второй, зная шифротексты и первый текст (рис. 2).

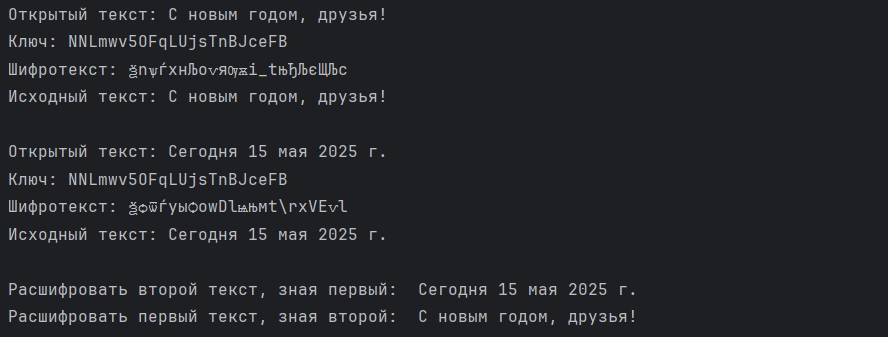


Рис. 2: Результат работы программы

Листинг программы 1

import random  
import string  
  
# генерация ключа  
def generate\_key(text):  
 key = ''  
 for i in range(len(text)):  
 key += random.choice(string.ascii\_letters + string.digits)  
 return key  
  
# шифрование и дешифрование текста  
def en\_de\_crypt(text, key):  
 new\_text = ''  
 for i in range(len(text)):  
 new\_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))  
 return new\_text  
  
# функция для нахождения возможных ключей  
def find\_keys(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  
  
text1 = 'С новым годом, друзья!'  
text2 = 'Сегодня 15 мая 2025 г.'  
  
key = generate\_key(text1)  
en\_crypt = en\_de\_crypt(text1, key)  
de\_crypt = en\_de\_crypt(en\_crypt, key)  
  
en\_crypt\_text2 = en\_de\_crypt(text2, key)  
de\_crypt\_text2 = en\_de\_crypt(en\_crypt\_text2, key)  
  
print("Открытый текст:", text1, "\nКлюч:", key, "\nШифротекст:", en\_crypt, "\nИсходный текст:", de\_crypt, "\n")  
print("Открытый текст:", text2, "\nКлюч:", key, "\nШифротекст:", en\_crypt\_text2, "\nИсходный текст:", de\_crypt\_text2, "\n")  
  
r = en\_de\_crypt(en\_crypt\_text2, en\_crypt)  
print('Расшифровать второй текст, зная первый: ', en\_de\_crypt(text1, r))  
print('Расшифровать первый текст, зная второй: ', en\_de\_crypt(text2, r))

# 4 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.

# 5 Выводы

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