Лабораторная работа №8

Основы информационной безопасности

Серёгина Ирина Андреевна

Содержание

# 1 Цель работы

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

# 2 Задание

Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, незная ключа и не стремясь его определить.

# 3 Теоретическое введение

Исходные данные. Две телеграммы Центра: P1 = НаВашисходящийот1204 P2 = ВСеверныйфилиалБанка Ключ Центра длиной 20 байт: K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54 Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведённой на рис. 8.1. Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования: C1 = P1 ⊕ K, C2 = P2 ⊕ K. (8.1) Открытый текст можно найти в соответствии с (8.1), зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства (8.1) складываются по модулю 2. Тогда с учётом свойства операции XOR 1 ⊕ 1 = 0, 1 ⊕ 0 = 1 (8.2) получаем: C1 ⊕ C2 = P1 ⊕ K ⊕ P2 ⊕ K = P1 ⊕ P2. Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар C1 ⊕ C2 (известен вид обеих шифровок). Тогда зная P1 и учитывая (8.2), имеем: C1 ⊕ C2 ⊕ P1 = P1 ⊕ P2 ⊕ P1 = P2. (8.3) Таким образом, злоумышленник получает возможность определить те символы сообщения P2, которые находятся на позициях известного шаблона сообщения P1. В соответствии с логикой сообщения P2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P2. Затем вновь используется (8.3) с подстановкой вместо P1 полученных на предыдущем шаге новых символов сообщения P2. И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

# 4 Выполнение лабораторной работы

1. Реализую две функции, как и в прошлой лабораторной работе, ввожу необходимые данные (рис. 1).

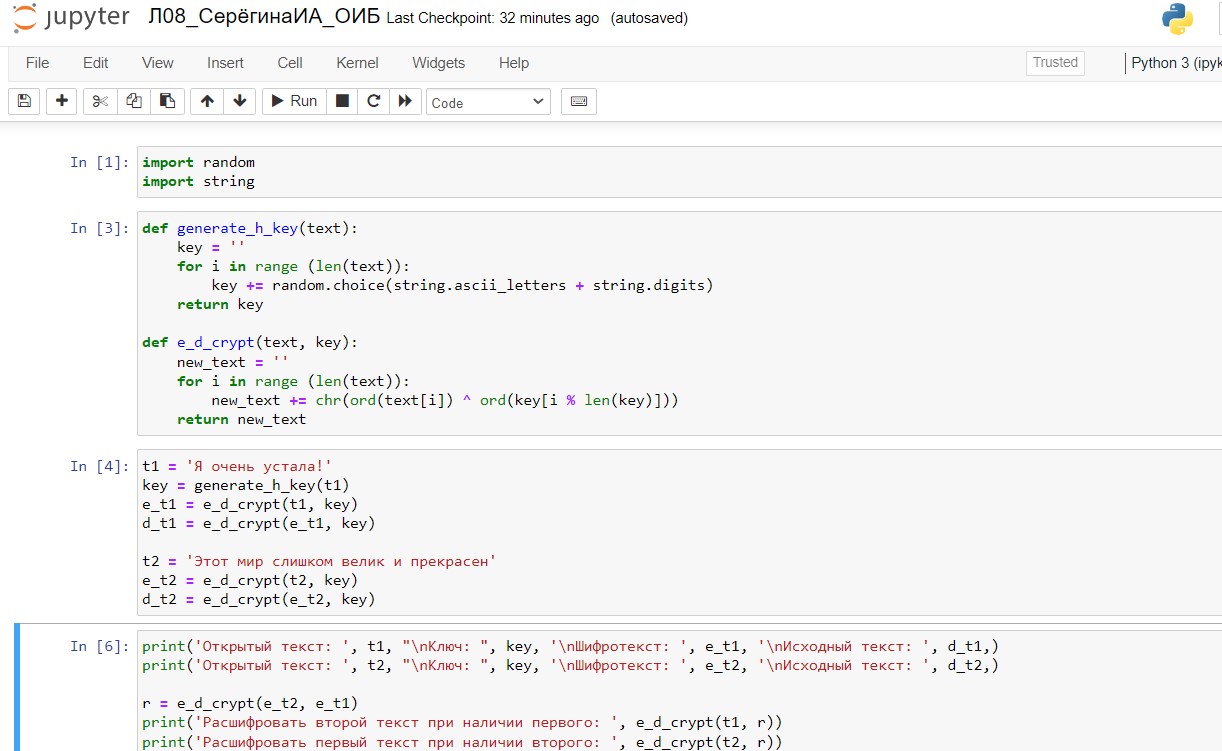


Рис. 1: код программы

1. Получаю необходимый результат (рис. 2).

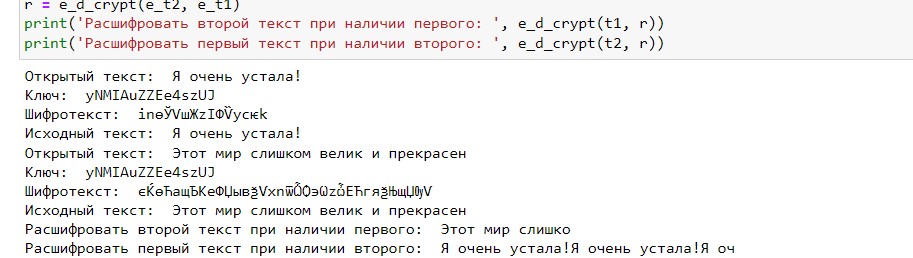


Рис. 2: результат работы программы

#Листинг

import random  
import string  
  
def generate\_h\_key(text):  
 key = ''  
 for i in range (len(text)):  
 key += random.choice(string.ascii\_letters + string.digits)  
 return key  
  
def e\_d\_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  
   
t1 = 'Я очень устала!'  
key = generate\_h\_key(t1)  
e\_t1 = e\_d\_crypt(t1, key)  
d\_t1 = e\_d\_crypt(e\_t1, key)  
  
t2 = 'Этот мир слишком велик и прекрасен'  
e\_t2 = e\_d\_crypt(t2, key)  
d\_t2 = e\_d\_crypt(e\_t2, key)  
  
print('Открытый текст: ', t1, "\nКлюч: ", key, '\nШифротекст: ', e\_t1, '\nИсходный текст: ', d\_t1,)  
print('Открытый текст: ', t2, "\nКлюч: ", key, '\nШифротекст: ', e\_t2, '\nИсходный текст: ', d\_t2,)  
  
r = e\_d\_crypt(e\_t2, e\_t1)   
print('Расшифровать второй текст при наличии первого: ', e\_d\_crypt(t1, r))  
print('Расшифровать первый текст при наличии второго: ', e\_d\_crypt(t2, r))

# 5 Выводы

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

# Список литературы