Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Топонен Никита Андреевич

Содержание

# Цель работы

1. Освоить на практике применение режима однократного гаммирования

# Задание

Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

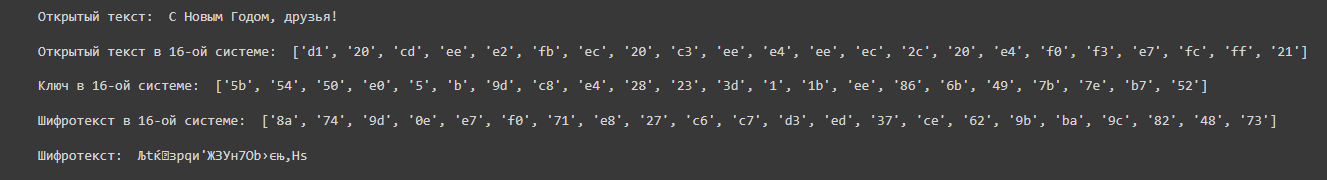
# Теоретическое введение

Для выполнения данной лабораторной нет специальной теории. Необходимы общие знания в области компьютерных наук.

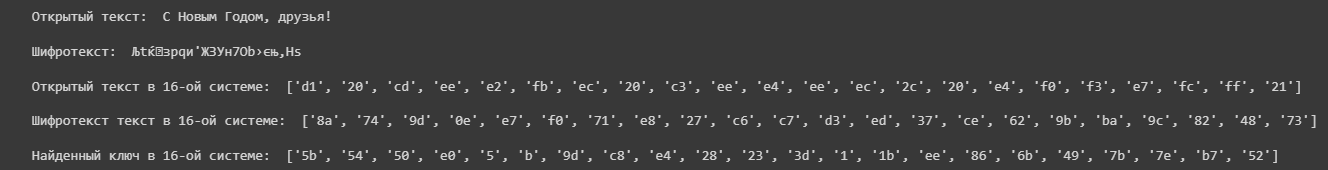
# Выполнение лабораторной работы

import numpy as np  
import operator as op  
import sys  
  
s = "С Новым Годом, друзья!"  
  
def encryption(text):  
 print("Открытый текст: ", text)  
  
 new\_text = []  
 for i in text:  
 new\_text.append(i.encode("cp1251").hex())  
 print("\nОткрытый текст в 16-ой системе: ", new\_text)  
  
 r = np.random.randint(0, 255, len(text))  
 key = [hex(i)[2:] for i in r]  
  
 new\_key = []  
 for i in key:  
 new\_key.append(i.encode("cp1251").hex().upper())  
 print("\nКлюч в 16-ой системе: ", key)  
  
 xor\_text = []  
 for i in range(len(new\_text)):  
 xor\_text.append("{:02x}".format(int(key[i], 16) ^ int(new\_text[i], 16)))  
 print("\nШифротекст в 16-ой системе: ", xor\_text)  
  
 en\_text = bytearray.fromhex("".join(xor\_text)).decode("cp1251")  
 print("\nШифротекст: ", en\_text)  
   
 return key, xor\_text, en\_text  
  
def find\_key(text, en\_text):  
 print("Открытый текст: ", text)  
 print("\nШифротекст: ", en\_text)  
   
 new\_text = []  
 for i in text:  
 new\_text.append(i.encode("cp1251").hex())  
 print("\nОткрытый текст в 16-ой системе: ", new\_text)  
   
 tmp\_text = []  
 for i in en\_text:  
 tmp\_text.append(i.encode("cp1251").hex())  
 print("\nШифротекст текст в 16-ой системе: ", tmp\_text)  
   
 xor\_text = [hex(int(k,16)^int(t,16))[2:] for (k,t) in zip(new\_text, tmp\_text)]  
 print("\nНайденный ключ в 16-ой системе: ", xor\_text)  
 return xor\_text  
  
k, t, et = encryption(s)  
key = find\_key(s, et)  
  
if k == key:  
 print("Ключ найден верно")  
else:  
 print("Ключ найден неверно")

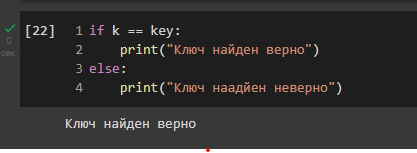
Результаты работы программы:



Вывод функции encryption



Вывод функции decryption



Проверка полученного ключа

# Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования. Гаммирование – выполнение операции XOR между элементами гаммы и элементами подлежащего сокрытию текста. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.
2. Перечислите недостатки однократного гаммирования. Абсолютная стойкость шифра доказана только для случая, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения.
3. Перечислите преимущества однократного гаммирования. Во-первых, такой способ симметричен, т.е. двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение. Во-вторых, шифрование и расшифрование может быть выполнено одной и той же программой. Наконец, Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P.
4. Почему длина открытого текста должна совпадать с длиной ключа? Если ключ короче текста, то операция XOR будет применена не ко всем элементам и конец сообщения будет не закодирован. Если ключ будет длиннее, то появится неоднозначность декодирования.
5. Какая операция используется в режиме однократного гаммирования, назовите её особенности? Наложение гаммы по сути представляет собой выполнение побитовой операции сложения по модулю 2, т.е. мы должны сложить каждый элемент гаммы с соответствующим элементом ключа. Данная операция является симметричной, так как прибавление одной и той же величины по модулю 2 восстанавливает исходное значение.
6. Как по открытому тексту и ключу получить шифротекст? В таком случае задача сводится к правилу: $C\_i = P\_i ⊕ K\_i $ ,т.е. мы поэлементно получаем символы зашифрованного сообщения, применяя операцию исключающего или к соответствующим элементам ключа и открытого текста.
7. Как по открытому тексту и шифротексту получить ключ? Подобная задача решается путем применения операции исключающего или к последовательностям символов зашифрованного и открытого сообщений: .
8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра? Необходимые и достаточные условия абсолютной стойкости шифра:
   * полная случайность ключа;
   * равенство длин ключа и открытого текста;
   * однократное использование ключа.

# Выводы

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

# Список литературы

* [Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №7](https://esystem.rudn.ru/pluginfile.php/1651757/mod_resource/content/2/007-lab_crypto-gamma.pdf)