Лабораторная работа № 7

Элементы криптографии. Однократное гаммирование.

Юрченко Артём Алексеевич

Содержание

# 1 Цель работы

Освоить на практике применение режима однократного гаммирования.

# 2 Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». 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.

# 3 Теоретические сведения

## 3.1 Шифр гаммирования

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (например, используя операцию сложения по модулю 2). Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Метод гаммирования становится бессильным, если известен фрагмент исходного текста и соответствующая ему шифрограмма. В этом случае простым вычитанием по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Метод гаммирования с обратной связью заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных. Например, если рассматривать гамму шифра как объединение непересекающихся множеств H(j), то процесс шифрования можно пердставить следующими шагами:

1. Генерация сегмента гаммы H(1) и наложение его на соответствующий участок шифруемых данных.
2. Подсчет контрольной суммы участка, соответствующего сегменту гаммы H(1).
3. Генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гамм H(2).
4. Подсчет контрольной суммы участка данных, соответствующего сегменту данных H(2) и т.д.

# 4 Выполнение лабораторной работы

## 4.1 Реализация шифратора и дешифратора Python

import string  
import random  
  
def to\_hex(text):  
 return " ".join(hex(ord(char))[2:] for char in text)  
  
def generate\_key(size):  
 key = "".join(random.choice(string.ascii\_letters + string.digits) for \_ in range(size))  
 return key  
  
def custom\_encoder(text, key):  
 return "".join(chr(a ^ b) for a, b in zip(text, key))  
  
message = "С Новым годом, друзья!"  
encryption\_key = generate\_key(len(message))  
hex\_key = to\_hex(encryption\_key)  
print("Ключ: ", hex\_key)  
  
encrypted\_text = custom\_encoder([ord(char) for char in message], [ord(char) for char in encryption\_key])  
hex\_text = to\_hex(encrypted\_text)  
print("Зашифрованное сообщение: ", hex\_text)  
  
decrypted\_text = custom\_encoder([ord(char) for char in encrypted\_text], [ord(char) for char in encryption\_key])  
print("Расшифрованный текст: ", decrypted\_text)

## 4.2 Контрольный пример (рис. [1](#fig:001))

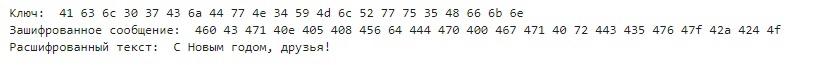


Figure 1: Работа алгоритма гаммирования

# 5 Выводы

В рамках данной лабораторной работы было освоено на практике применение режима однократного гаммирования.

# Список литературы

[1] https://www.kaspersky.ru/resource-center/definitions/encryption

[2] https://xakep.ru/2019/07/18/crypto-xor/