

Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Ли Тимофей Александрович

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Ответы на контрольные вопросы	9
Выводы	11

Список таблиц

Список иллюстраций

0.1	функция определения шифротекста	6
0.2	результат работы функции1	7
0.3	Функция определения ключа	7
0.4	результат работы функции2	8
0.5	проверка ключа	8

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

Написал функцию для определения вида шифротекста при известном ключе и известном открытом тексте. Функция получает на вход строку, переводит ее в шестнадцатеричную систему счисления. Затем в программе рандомно генерируется ключ. При помощи ключа получаю зашифрованное сообщение в шестнадцатеричной системе счисления. Затем перевожу это сообщение в строковый вид. (рис. @fig:001):

```
In [1]: import numpy as np
import pandas as pd
import sys

In [6]: a="С Новым Годом, друзья!"
def crypt(a):
    print("open text: ", a)
    text=[]
    for i in a:
        text.append(i.encode("cp1251").hex())
    print("open text in 16: ", *text)
    k=np.random.randint(0, 255, len(a))
    key=[hex(i)[2:] for i in k]
    newkey=[]
    for i in key:
        newkey.append(i.encode("cp1251").hex().upper())
    print("key in 16: ", *key)
    b=[]
    for i in range(len(text)):
        b.append("{:02x}".format(int(key[i],16)^int(text[i],16)))
    print("cypher text in 16: ", *b)
    fintext=bytearray.fromhex("".join(b)).decode("cp1251")
    print("cypher text: ", fintext)
    return key, b, fintext
```

Рис. 0.1: функция определения шифротекста

Вывод функции: (рис. @fig:002)

```
In [7]: key, b, fintext=crypt(a)

open text: С Новым Годом, друзья!
open text in 16: d1 20 cd ee e2 fb ec 20 c3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff 21
key in 16: d9 3 78 83 9a e4 ae bc ee fa 60 f0 ab 9 b6 e0 d1 10 8b a a6 64
cypher text in 16: 08 23 b5 6d 78 1f 42 9c 2d 14 84 1e 47 25 96 04 21 e3 6c f6 59 45
cypher text: #мхВъ-,G%-!гlцҮЕ
```

Рис. 0.2: результат работы функции1

Определим ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Функция нахождения ключа получает на вход две строки: открытый текст и зашифрованный. Затем она преобразует строки в шестнадцатеричный формат и выполняет операцию XOR для нахождения ключа: (рис. @fig:003)

```
In [8]: def findkey(a,fintext):
        print("open text: ", a, "\ncypher text: ", fintext)
        newtext=[]
        for i in a:
            newtext.append(i.encode("cp1251").hex())
        print("open text in 16: ", *newtext)
        ftext=[]
        for i in fintext:
            ftext.append(i.encode("cp1251").hex())
        print("cypher text in 16: ", *ftext)
        key=[hex(int(i,16)^int(j,16))[2:] for (i,j) in zip(newtext,ftext)]
        print("found key in 16: ", *key)
        return key
```

Рис. 0.3: Функция определения ключа

Вывод: (рис. @fig:004)

```
In [9]: keyy=findkey(a,fintext)

open text:  С Новым Годом, друзья!
cypher text: #µтхВъ-,,G%-!г1цҮЕ
open text in 16:  d1 20 cd ee e2 fb ec 20 c3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff 21
cypher text in 16:  08 23 b5 6d 78 1f 42 9c 2d 14 84 1e 47 25 96 04 21 e3 6c f6 59 45
found key in 16:  d9 3 78 83 9a e4 ae bc ee fa 60 f0 ab 9 b6 e0 d1 10 8b a a6 64
```

Рис. 0.4: результат работы функции2

В конце проверил полученный ключ и тот, который был изначально сгенерирован:
(рис. @fig:005)

```
In [11]: if key==keyy:
          print("correct key")
        else:
          print("incorrect key")

correct key
```

Рис. 0.5: проверка ключа

Как видим, ключ действительно тот.

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:

$$K_i = P_i (+) C_i.$$

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

Освоил на практике применение режима однократного гаммирования.