Лабораторная работа № 7. Элементы криптографии. Однократное гаммирование

Лёшьен Стефани, НФИбд-02-19

Содержание

1 Цель работы **5**

2 Задача **6**

3 Последовательность выполнения работы **7**

4 Выводы **10**

5 Библиография **11**

List of Figures

3.1 Импорт библиотек 7
3.2 Функция определения шифротекста 8
3.3 результат 8
3.4 Вторая функция 8
3.5 Результат 9
3.6 html-файл test.html 9

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Задача

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 Последовательность выполнения работы

1. В новом блокноте в Google Colaboratory я импортировала нужные мне библиотеки.

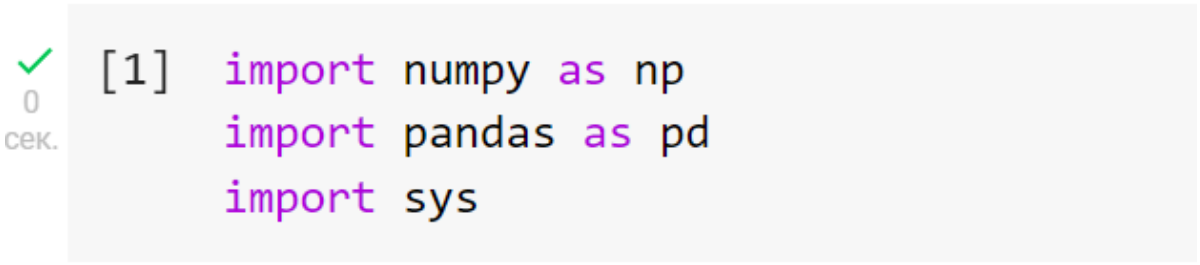

Figure 3.1: Импорт библиотек

2. Далее, я написала фунцию для определения вида шифротекста при известном ключе и известном открытом тексте. Сначала я перевела наш открытый текст в шестнадцатеричную систему и сгенерировала рандомный ключ. При помощи ключа я получаю зашифрованный текст в шестнадцатеричном формате и шифротекст.

```
a=" С Новым годом, друзья"
def crypt(a):
  print("open text : ",a)
  text=[]
  for i in a :
    text.append(str(i).encode("cp1251").hex())
  print("open text in 16: ", *text)
  k=np.random.randint(0, 225, len(a))
  key=[hex(i)[2:] for i in k]
  newkey=[]
  for i in k:
    newkey.append(str(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
```

Figure 3.2: Функция определения шифротекста

3. В результате выполнения этой функции мы получаем на выход открытый текст, ключ в шестнадцатеричной системе счисления.

```
[39] key, b, fintext=crypt(a)

open text :   С Новым годом, друзья
open text in 16:  20 43 20 cd ee e2 fb ec 20 e3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff
key in 16:  d0 4d 95 a3 ba 80 d2 3d 79 44 13 62 84 b5 a0 3e ca 84 21 ac 13 19
cypher text in 16: f0 0e b5 6e 54 62 29 d1 59 a7 fd 86 6a 59 8c 1e 2e 74 d2 4b ef e6
cypher text:  p□µnTb)CY§ə†jY№□.tTKпж
```

Figure 3.3: результат

4. вторая фунция

```
def findKey(a, fintext):
  print("open text : ",a,"\ncyber text:",fintext)
  newText=[]
  for i in a:
    newText.append(str(i).encode("cp1251").hex())
  print("open text in 16 : ", *newText)
  ftext=[]
  for i in fintext:
    ftext.append(str(i).encode("cp1251").hex())
  print("cyber 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
```

Figure 3.4: Вторая функция

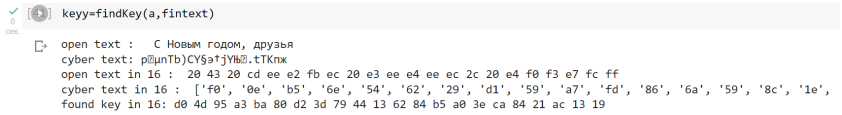

Figure 3.5: Результат

5. Проверяем если полученный ключ совпадает с тем, который мы получили на предыдущем шаге.

```
if key==keyy:
  print("correct key")
else:
  print("incorrect key")
```

```
correct key
```

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

Figure 3.6: html-файл test.html

4 Выводы

Я освоила на практике применение режима однократного гаммирования.

5 Библиография

1. Методические материалы курса