Информационная безопасность лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Ким Илья Владиславович НФИбд-01-21

Содержание

Цель работы

Освоить на практике применение режима однократного гаммирования

Выполнение лабораторной работы

1. Подключил библиотеки и ввел сообщение “С Новым Годом, друзья!” (рис. [-@fig:001])

```
import numpy as np
import pandas as pd
```

```
a="С Новым Годом, друзья!"
```

2. Перевод сообщения в шестнадцатеричную систему счисления (рис. [-@fig:002])(рис. [-@fig:003])

```
def cr(a):
    print ("Текст: ", a)
    print("\n№1 Кодировка текста")
    print("_____________________\n")
    text=[]
    for i in a:
        text.append(i.encode("cp1251").hex().upper())
    print ("Закодированный текст: ", *text)
```

```
Текст:  С Новым Годом, друзья!

№1 Кодировка текста
_____________________

Закодированный текст:  D1 20 CD EE E2 FB EC 20 C3 EE E4 EE EC 2C 20 E4 F0 F3 E7 FC FF 21
```

3. Создание ключа (рис. [-@fig:004])(рис. [-@fig:005])

```
print("\n№2 Создание ключа")
print("_____________________\n")
k=np.random.randint(0,255,len(a))
key=[hex(i).upper()[2:] for i in k]
print("Ключ:                  ", *key)
```

```
№2 Создание ключа
_____________________

Ключ:                  98 54 57 C4 A0 4 0 24 83 18 33 13 43 B3 23 C7 3A 9E 85 3C 7F 8E
```

4. Кодировка текста ключем (рис. [-@fig:006])(рис. [-@fig:007])

```
print("\n№3 Кодировка текста ключем")
print("_____________________\n")
newa=[]
for i in range(len(text)):
    newa.append("{:02x}".format(int(key[i],16)^int(text[i],16)).upper())
print("Закодированный ключем текст:",*newa )
```

```
№3 Кодировка текста ключем
_____________________

Закодированный ключем текст: 49 74 9A 2A 42 FF EC 04 40 F6 D7 FD AF 9F 03 23 CA 6D 62 C0 80 AF
```

5. Создание нового ключа (рис. [-@fig:008])(рис. [-@fig:009])

```
print("\n№4 Создание нового ключа")
print("_____________________\n")
newtext=[]
b=np.random.randint(0,255,len(a))
newkey=[hex(i).upper()[2:] for i in b]
print("Созданный ключ:            ", *newkey)
print("\n№5 Раскодирование текста новым ключем")
```

```
№4 Создание нового ключа
_____________________

Созданный ключ:            C9 56 72 74 37 A5 35 3D EE 79 62 3 CC 62 70 73 CB E4 96 F7 6D DF
```

6. Раскодировка текста новым ключем (рис. [-@fig:010])(рис. [-@fig:011])

```
print("\n№5 Раскодирование текста новым ключем")
print("_____________________\n")
for i in range(len(text)):
    newtext.append("{:02x}".format(int(newkey[i],16)^int(newa[i],16)).upper())
print("Раскодированный ключем текст:",*newtext )
```

```
№5 Раскодирование текста новым ключем
_____________________

Раскодированный ключем текст: 80 22 E8 5E 75 5A D9 39 AE 8F B5 FE 63 FD 73 50 01 89 F4 37 ED 70
```

7. Полученный текст (рис. [-@fig:012]) (рис. [-@fig:013])

```
print("\n№6 Полученный текст")
print("____________________\n")
faketext=bytearray.fromhex("".join(newtext)).decode("cp1251").upper()
print("Текст с новым ключем: ", faketext)
```

```
№6 Полученный текст

____________________

Текст с новым ключем:  Ъ"И^UZЩ9®ЦМЮСЭSP⍰‰Ф7НР
```

8. Поиск нужного ключа по исходному и закодированному тексту (рис. [-@fig:014])(рис. [-@fig:015])

```
print("\n№7 Поиск нужного ключа по исходному тексту и закодированному")
print("____________________________________________________________\n")
findkey=[]
for i in range(len(text)):
    findkey.append("{:02x}".format(int(newa[i],16)^int(text[i],16)).upper())
print("Найденный ключ:",*findkey )
```

```
№7 Поиск нужного ключа по исходному тексту и закодированному
____________________________________________________________

Найденный ключ: 98 54 57 C4 A0 04 00 24 83 18 33 13 43 B3 23 C7 3A 9E 85 3C 7F 8E
```

9. Расшифровка текста по найденному ключу (рис. [-@fig:016])(рис. [-@fig:017])

```
print("\n№8 Расшифрованный текст по найденному ключу")
print("_____________________________________________\n")
truetext=[]
for i in range(len(text)):
    truetext.append("{:02x}".format(int(findkey[i],16)^int(newa[i],16)).upper())
truetext=bytearray.fromhex("".join(truetext)).decode("cp1251")
print("Правильный текст: ", truetext)
return
```

```
№8 Расшифрованный текст по найденному ключу

_____________________________________________

Правильный текст:  С Новым Годом, друзья!
```

Листинг

import numpy as np

import pandas as pd

a=“С Новым Годом, друзья!”

def cr(a): print (“Текст:”, a)

```
print("\n№1 Кодировка текста")

print("_____________________\n")
text=[]
for i in a:
    text.append(i.encode("cp1251").hex().upper())
print ("Закодированный текст: ", *text)

print("\n№2 Создание ключа")
print("_____________________\n")
k=np.random.randint(0,255,len(a))
key=[hex(i).upper()[2:] for i in k]
print("Ключ:                 ", *key)

print("\n№3 Кодировка текста ключем")
print("__________________________\n")
newa=[]
for i in range(len(text)):
```

```
    newa.append("{:02x}".format(int(key[i],16)^int(text[i],16)).upper())
print("Закодированный ключем текст:",*newa )

print("\n№4 Создание нового ключа")
print("_________________________\n")
newtext=[]
b=np.random.randint(0,255,len(a))
newkey=[hex(i).upper()[2:] for i in b]
print("Созданный ключ:           ", *newkey)

print("\n№5 Раскодирование текста новым ключем")
print("_____________________________________\n")
for i in range(len(text)):
    newtext.append("{:02x}".format(int(newkey[i],16)^int(newa[i],16)).upper())
print("Раскодированный ключем текст:",*newtext )

print("\n№6 Полученный текст")
print("__________________\n")
faketext=bytearray.fromhex("".join(newtext)).decode("cp1251").upper()
print("Текст с новым ключем: ", faketext)

print("\n№7 Поиск нужного ключа по исходному тексту и закодированному")
print("____________________________________________________________\n")
findkey=[]
for i in range(len(text)):
    findkey.append("{:02x}".format(int(newa[i],16)^int(text[i],16)).upper())
print("Найденный ключ:",*findkey )

print("\n№8 Расшифрованный текст по найденному ключу")
```

```
print("_____________________________________\n")
truetext=[]
for i in range(len(text)):
    truetext.append("{:02x}".format(int(findkey[i],16)^int(newa[i],16)).upper())
truetext=bytearray.fromhex("".join(truetext)).decode("cp1251")
print("Правильный текст: ", truetext)
return
```

Выводы

Освоил на практике применение однократного гаммирования