

Лабораторная работа №7

Ilyinsky A. Arseniy

RUDN University, 2022 Moscow, Russia

Элементы криптографии. Однократное гаммирование

Цель выполнения работы

- 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.

Выполнение работы

Выполнение 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
```

Выполнение работы

```
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
```

Выполнение работы

```
k, t, et = encryption(s)
key = find_key(s, et)

if k == key:
    print("Ключ найден верно")
else:
    print("Ключ найден неверно")
```


Выполнение работы

Открытый текст: С Новым Годом, друзья!

Открытый текст в 16-ой системе: ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Ключ в 16-ой системе: ['5b', '54', '50', 'e0', '5', 'b', '9d', 'c8', 'e4', '28', '23', '3d', '1', '1b', 'ee', '86', '6b', '49', '7b', '7e', 'b7', '52']

Шифротекст в 16-ой системе: ['8a', '74', '9d', '0e', 'e7', 'f0', '71', 'e8', '27', 'c6', 'c7', 'd3', 'ed', '37', 'ce', '62', '9b', 'ba', '9c', '82', '48', '73']

Шифротекст: ЪtkЉзrqi'ЖЗУн70b>ень,Hs

Рис.1 Вывод функции encryption

Выполнение работы

Открытый текст: С Новым Годом, друзья!

Шифротекст: ЉtkЉзrqи'ЖЗУн70b>єь,Нs

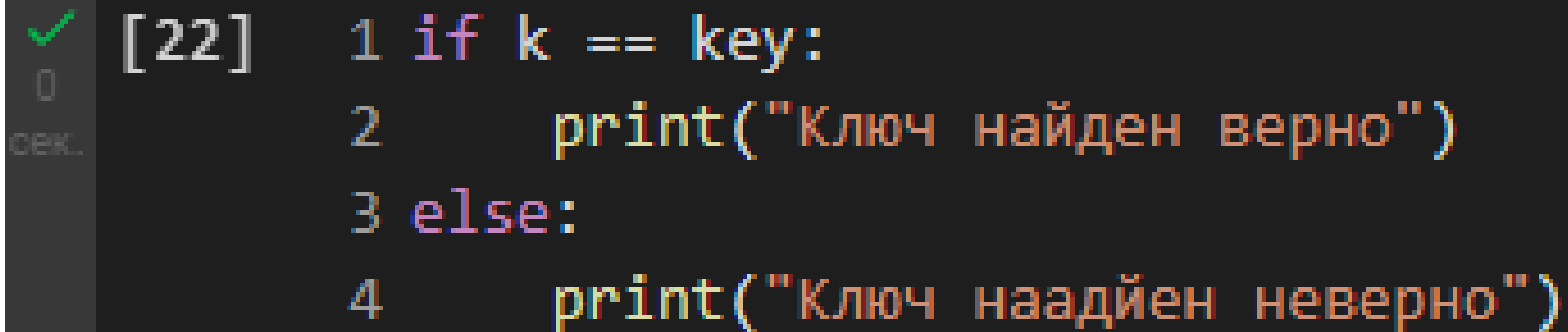
Открытый текст в 16-ой системе: ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Шифротекст текст в 16-ой системе: ['8a', '74', '9d', '0e', 'e7', 'f0', '71', 'e8', '27', 'c6', 'c7', 'd3', 'ed', '37', 'ce', '62', '9b', 'ba', '9c', '82', '48', '73']

Найденный ключ в 16-ой системе: ['5b', '54', '50', 'e0', '5', 'b', '9d', 'c8', 'e4', '28', '23', '3d', '1', '1b', 'ee', '86', '6b', '49', '7b', '7e', 'b7', '52']

Рис.2 Вывод функции decryption

Выполнение работы



The image shows a Jupyter Notebook cell with a green checkmark icon and a '0 сек.' (0 seconds) execution time indicator. The code is as follows:

```
[22] 1 if k == key:  
      2     print("Ключ найден верно")  
      3 else:  
      4     print("Ключ наадйен неверно")
```

Ключ найден верно

Рис.3 Проверка полученного ключа

Вывод

Спасибо за внимание