Лабораторная работа №7

Андрианова Марина Георгиевна
RUDN University, Moscow,
Russian Federation
2024, 15 October

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

Я выполняла лабораторную работу на языке программирования Python.

• Требуется разработать программу, позволяющую шифровать и дешифровать данные в режиме однократного гаммирования. Начнем с создания функции для генерации случайного ключа (рис. [-@fig:001]).

```
import random
import string

def generate_key_hex(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits) #генерация цифры для каждого символа в тексте
    return key
```

Рис.1. Функция генерации ключа

Необходимо определить вид шифротекста при известном ключе и известном открытом тексте.
Делаю одну функцию и для шифрования, и для дешифрования текста (рис. [-@fig:002]).

```
#для шифрования и дешифрования
def en_de_crypt(text, key):
    new_text = ''
    for i in range(len(text)): #проход по каждому символу в тексте
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
    return new_text
```

Рис.2. Функция для шифрования текста

Нужно определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Для этого создаю функцию для нахождения возможных ключей для фрагмента текста (рис. [-@fig:003]).

```
def find_possible_key(text, fragment):
    possible_keys = []
    for i in range(len(text) - len(fragment) + 1):
        possible_key = ""
        for j in range(len(fragment)):
            possible_key += chr(ord(text[i + j]) ^ ord(fragment[j]))
        possible_keys.append(possible_key)
    return possible_keys
```

Рис.3. Подбор возможных ключей для фрагмента

В следующей части кода реализуем шифрование и дешифрование текста, а также поиск возможных ключей для расшифровки (рис. [-@fig:004]).

```
t = 'С Новым Годом, друзья!'
key = generate_key_hex(t)
en_t = en_de_crypt(t, key)
de_t = en_de_crypt(en_t, key)
keys_t_f = find_possible_key(en_t, 'С Новым')
fragment = "С Новым"
print('Открытый текст: ', t, "\nКлюч: ", key, '\nШифротекст: ', en_t, '\nИсходный текст: ', de_t,)

print('Возможные ключи: ', keys_t_f)
print('Расшифрованный фрагмент: ', en_de_crypt(en_t, keys_t_f[0]))
```

Рис.4. Шифрование и дешифрование текста

Проверка работы всех функций. Шифрование и дешифрование происходит верно, как и нахождение ключей, с помощью которых можно расшифровать верно только кусок текста (рис. [-@fig:005]).

```
Открытый текст:  С Новым Годом, друзья!
Ключ:  zGA2wi8attkbwy8nfpbrGA
Шифротекст:  ђgќЌxTЄAaъцќыUЋњЦгsoJ`
Исходный текст:  С Новым Годом, друзья!
Возможные ключи:  ['zGA2wi8', 'цЋ\x11{\x10Oѓ', '}ЬX\x1c6Ѣ[', '-ж?:ѳ,v', 'dЂ\x19ЎU\x01c', '\x03ЌќYx\x14`', '%aztm\x17w', 'ЄчWan\x00ы', 'FЖBbyOЌ', 'kЎAuaґf', '~ЋVжЂ\
x11\x1a', '}жшЦhm\x0f', 'juSd\x14xi', 'V8G\x18\x01\x1e\x02', 'йФ;\rgu4', '{I.k\x0cCќ']
Расшифрованный фрагмент:  С Новым;РЂњЊTmbHаЁTїaЋ
```

Рис.5. Результат работы программы

Выводы

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования.