Презентация по лабораторной работе №7

Основы информационной безопасности

Дворкина Е. В

22 апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Дворкина Ева Владимировна
- студентка группы НКАбд-01-22
- Российский университет дружбы народов
- https://vk.com/yuri.kamori

Цель

Освоить на практике применение режима однократного гаммирования

Задание

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

Выполнение лабораторной работы

Я выполнала лабораторную работа на языке программирования Python, листинг программы и результаты выполнения приведены в отчете.

Выполнение лабораторной работы

Требуется разработать программу, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Начнем с создания функции для генерации случайного ключа

```python
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

Выполнение лабораторной работы

Необходимо определить вид шифротекста при известном ключе и известном открытом тексте. Так как операция исключающего или отменяет сама себя, делаю одну функцю и для шифрования и для дешифрования текста

```python
#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

Выполнение лабораторной работы

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

```
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

Проверка работы всех функций. Шифрование и дешифрование происходит верно, как и нахождение ключей, с помощью которых можно расшифровать верно только кусок текста

```
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]))
```

```
Открытый текст:  С Новым Годом, друзья!
Ключ:  ApV5aBX756a1RJPixoVYII
Шифротекст:  ѠPыЋѓЉЄ◻ЦJsЏѮfpйиЬwEIh
Исходный текст:  С Новым Годом, друзья!
Возможные ключи:  ['ApV5aBX', 'ѱӂ\x16m;/Ы', 'jЫN7Vќ\x1a', '*ɵ\x14ZXm4', 'rЩyЩ\x14Ci', '(фЊ\x18:\x1e3', 'E7;6gDR', 'жI\x15k=%њ', '\x07ШН1\\Эь', ')v\x12Pєл
a', 'tЯsjт\x16\x04', '.юФюos\x10', 'OFӝc\ng]', 'чP@\x06\x1e*)', 'ёѽ%\x12S^:', "|И1_'Mє"]
Расшифрованный фрагмент:  С НовымViўѠѮЬ>1ЭѮЙЀїў)
```

Рис. 4: Результат работы программы

Вывод

В ходе выполнения данной лабораторной работы мной было освоено на практике применение режима однократного гаммирования.

:::