Отчёт по лабораторной работе №7

Дисциплина: Основы информационной безопасности

Паращенко Антонина Дмитриевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Выполнение лабораторной работы

1) Функция для генерации ключа. (рис. 2.1)

```
def generate_key(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits)
    return key
```

Рис. 2.1: Генерация ключа

2) Функция для (де)шифрования. (рис. 2.2)

```
def crypt(text, key):
    new_text = ''
    for i in range(len(text)):
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
    return new_text
```

Рис. 2.2: Шифрование

3) Функция для подбора ключа. рис. 2.3)

```
def find_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. 2.3: Подбор ключа

4) Код для вывода результатов и результаты. (рис. 2.4)

```
t = 'С Новым Годом, друзья!'
key = generate_key(t)
encrypt = crypt(t, key)
decrypt = crypt(encrypt, key)
keys = find_key(encrypt, "С Новым Годом")
fragment = "С Новым Годом"

print('Открытый текст:', t, "\nКлюч:", key, "\nШифротекст:", encrypt, "\nИсходный текст:", decrypt)
print('Возможные ключи:', keys)
print('Расшифрованный фрагмент:', crypt(encrypt, keys[0]))
```

```
Открытый текст: С Новым Годом, друзья!
Ключ: xTfijecGs9D8jHwR6lMOOw
Шифротекст: љtọïjЮụgΩЇΨIidWѧѶЯǪЃÈV
Исходный текст: С Новым Годом, друзья!
Возможные ключи: ['xTfijecGs9D8j', 'sħJf\x1c\x14ħp\x14N2hj', 'ZѷE\x10mЬ\\Ч8bњѫ', 'vӪ3as+;è\x15hèѩZ', 'yЎBљRLLЦЕњѢXJ', '\x0fѿǪ^5;:ѶѷѩRH\x13', '~G}9BMjDфX
B\x11F', 'цр\x1aN4\x1djwuH\x1bD?', 'АЧm8dЯѫце\x11N=<', '&è\x1bhiMZi<D7>Ж']
Расшифрованный фрагмент: С Новым Годом└╘ЁПхПΩч%
```

Рис. 2.4: Результаты

5) Листинг программы. (рис. 2.5)

```
import random
import string

def generate_key(text):
    key = ''
    for i in range(len(text)):
        key += random.choice(string.ascii_letters + string.digits)
    return key

def crypt(text, key):
    new_text = ''
    for i in range(len(text)):
        new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
    return new_text

def find_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

t = 'С Новым Годом, друзья!'
key = generate_key(t)
encrypt = crypt(t, key)
decrypt = crypt(encrypt, key)
keys = find_key(encrypt, "С Новым Годом")
fragment = "С Новым Годом"

print('Открытый текст:', t, "\nКлюч:", key, "\nШифротекст:", encrypt, "\nИсходный текст:", decrypt)
print('Возможные ключи:', keys)
print('Расшифрованный фрагмент:', crypt(encrypt, keys[0]))
```

Рис. 2.5: Листинг

3 Вывод

В результате выполнения работы мы научились на практике применять режим однократного гаммирования.

Список литературы

1) https://esystem.rudn.ru/pluginfile.php/2357157/mod_resource/content/2/007-lab_crypto-gamma.pdf