Отчет по лабораторной работе №7

Информационная безопасность

Нгуен Фыок Дат

Содержание

Список таблиц

Список иллюстраций

1 Цель работы

- Освоить на практике применение режима однократного гаммирования.

2 Задание

- Написание программы
- Зашифровка текста по открытому тексту и известному ключу
- Определение ключа по открытому и зашифрованному тексту

3 Теоретическое введение

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой.

Более подробно о см. в [1].

4 Выполнение лабораторной работы

ДляП выполненияП лабораторнойП работыП яП написалП программуП дляП зашифрованияП и расшифровкиП текста.П ИмпортировавП необходимыеП библиотеки,П яП задалП триП функцииП – для генерации ключа по размеру сообщения (выбор случайных букв в кодировке ASCII), для перевода сообщения и ключей в 16-ричную систему (получение кода символа и перевод его в 16-ричную систему) и для шифрования (поэлементный XOR) (рис. 4.1).

```
import string
import random
import sys
```

```
def gen_key(size):
    res =""
    for i in range(size):
        res += random.choice(string.ascii_letters)
    return res

def to_hex(text):
    res =""
    for i in text:
        res += hex(ord(i))[2:] +" "
    return res

def encrypt(text,key):
    res =""
    for (t,k) in zip(text,key):
        res += chr(t^k)
    return res
```

Рис. 4.1: Программа, 1

ДалееП яП задалП изначальноеП открытоеП сообщение,П сгенерировалП ключ,П закодировалП со

общение с помощью этого ключа, а также перевела все в 16-ричную систему. После этого яП раскодировалП закодированноеП сообщение,П чтобыП проверитьП правильностьП работыП программы,П иП определилП используемыйП ключП поП открытомуП сообщениюП иП закодированному сообщению (рис. 4.2).

```python
msg ="С Новым Годом, друзья!"
msg16 = to_hex(msg)
key = gen_key(len(msg))
k = to_hex(key)
encr_m = encrypt([ord(i) for i in msg], [ord(i) for i in key])
encr_m16 = to_hex(encr_m)
decr_m = encrypt([ord(i) for i in encr_m], [ord(i) for i in key])
key_new = encrypt([ord(i) for i in msg], [ord(i) for i in encr_m])

print("open text:", msg, "\nopen text in 16:", msg16, "\nkey:", key, "\nkey in 16:", k)
print("encrypted text:", encr_m, "\nencrypted text in 16", encr_m16, "\ndecrypted msg:", decr_m,"\nfound key:", key_new)
```

Рис. 4.2: Программа, 2

ПолученныеП сообщенияП иП ключиП яП вывелП наП экранП (рис.П 4.3).П СообщениеП былоП но закодировано и раскодировано, а найденный ключ совпадает с тем, который был сгенерирован для кодирования.

```
open text: С Новым Годом, друзья!
open text in 16: 421 20 41d 43e 432 44b 43c 20 413 43e 434 43e 43c 2c 20 434 440 443 437 44c 44f 21
key: hyvCGKJYdNeNvobNPQQuWq
key in 16: 68 79 76 43 47 4b 4a 59 64 4e 65 4e 76 6f 62 4e 50 51 51 75 57 71
encrypted text: щYѫѽvЀѶyѷѰёѰъCBѺАВѦйИP
encrypted text in 16 449 59 46b 47d 475 400 476 79 477 470 451 470 44a 43 42 47a 410 412 466 439 418 50
decrypted msg: С Новым Годом, друзья!
found key: hyvCGKJYdNeNvobNPQQuWq
```

Рис. 4.3: Результат запуска программы

5 Выводы

ВП результатеП лабораторнойП работыП яП получилП представлениеП оП базовыхП элементах криптографииП иП освоилП наП практикеП применениеП режимаП однократногоП гаммирования,П написав программу, позволяющую зашифровывать и расшифровывать тексты и определять использованные для этого ключи.

Список литературы

1. Элементы криптографии. Однократное гаммирование [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/2090284/mod_resource/content/2/007-lab_crypto-gamma.pdf.