Отчет по лабораторной работе №7

по дисциплине: Информационная безопасность

Го Чаопэн

Содержание

Список иллюстраций

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 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.

3 Теоретическое введение

- Шифрование – это технология кодирования и раскодирования данных.Зашифрованные данные -это результат применения алгоритма для кодирования данных с целью сделать их недоступными для чтения. Данные могут быть раскодированы в исходную форму только путем применения специальный ключа [1].
- Гаммирование — это наложение (или снятие при расшифровке сообщений) на открытое (или зашифрованное) сообщение так называемой криптографической гаммы. Криптографическая гамма — это последовательность элементов данных, которая вырабатывается с помощью определенного алгоритма. [2].

4 Выполнение лабораторной работы

1. Импортируем необходимые модули (4.1).

```
[guo@guo ~]$ python
Python 3.9.16 (main, Sep 12 2023, 00:00:00)
[GCC 11.3.1 20221121 (Red Hat 11.3.1-4)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import string
>>> import random
```

Рис. 4.1: Импорт модулей

2. Создадим функции для преобразования данных в шестнадцатеричный формат, генерации ключа и кодирования, декодирования данных (4.2).

```
>>> def to_hex(text):
...     return " ".join(hex(ord(i))[2:] for i in text)
...
>>> def generate_key(size):
...     key = "".join(random.choice(string.ascii_letters + string.digits) for _ in range(size))
...     return key
...
>>> def encoder(text, key):
...     return "".join(chr(a^b) for a, b in zip(text, key))
...
```

Рис. 4.2: Функции

3. Закодируем и декодируем строку “С Новым годом, друзья!” (4.3).

```
>>> mess = "С Новым годом, друзья!"
>>> key = generate_key(len(mess))
>>> hex_key = to_hex(key)
>>>
>>> enc_text = encoder([ord(i) for i in mess], [ord(i) for i in key])
>>> hex_text = to_hex(enc_text)
>>> decr_text = encoder([ord(i) for i in enc_text], [ord(i) for i in key])
>>>
>>> print("Ключ: ", hex_key)
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 4c 62 4d 37 44 70 6f
>>> print("Зашифрованное сообщение: ", hex_text)
Зашифрованное сообщение:  418 74 45c 44f 462 42f 452 52 441 40e 45d 406 445 62 42 478 422 40e 400 408 43f 4e
>>> print("Расшифрованный текст: ", decr_text)
Расшифрованный текст:  С Новым годом, друзья!
>>>
```

Рис. 4.3: Кодирование и декодирование строки

4. Получим ключ, с помощью которого получим сообщения “С Новым годом, семья”, “С Новым годом, учителя” вместо “С Новым годом, друзья!” при декодировании. Воспользуемся симметричностью кодирования(4.4).

```
>>> new_mess = "С Новым годом, семья!"
>>>
>>> key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_mess])
>>> print("Ключ: ", to_hex(key))
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 39 17 32 4c 47 41e
>>>
>>> new_mess = "С Новым годом, учителя!"
>>>
>>> key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_mess])
>>> print("Ключ: ", to_hex(key))
Ключ:  39 54 41 71 50 64 6e 72 72 30 69 38 79 4e 62 3b 65 36 42 3d 4 401
>>>
```

Рис. 4.4: Получение ключа для другого прочтения открытого текста

5 Выводы

В рамках данной лабораторной работы было освоено на практике применение режима однократного гаммирования.

6 Список литературы

1. Шифрование информации: как защитить свои данные [Электронный ресурс]. URL: https://gb.ru/blog/shifrovanie-informatsii/.

2. Гаммирование [Электронный ресурс]. URL: https://science.fandom.com/ru/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5.