Отчет по лабораторной работе №7

по дисциплине: Информационная безопасность

Ван И

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).

```
>>> 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 hex_text(text):
...     return " ".join(hex(ord(i))[2:] for i in text)
...
>>> def generate(size):
...     key = "".join(random.choice(string.ascii_letters + string.digits) for _ in range(size))
...     return key
...
>>> def enc(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(len(mess))
>>> hex_key = hex_text(key)
>>>
>>> enc_text = enc([ord(i) for i in mess], [ord(i) for i in key])
>>> hex_text = hex_text(enc_text)
>>> decr_text = enc([ord(i) for i in enc_text], [ord(i) for i in key])
>>>
>>> print("Ключ: ", hex_key, "\nЗашифрованное сообщение: ", hex_text, "\nРасшифрованный текст: ", decr_text)
Ключ:  57 42 55 76 70 49 75 48 6c 72 34 6a 4e 59 71 64 52 7a 73 45 61 44
Зашифрованное сообщение:  476 62 448 448 442 402 449 68 45f 44c 400 454 472 75 51 450 412 439 444 409 42e 65
Расшифрованный текст:  С Новым годом, друзья!
```

Рис. 4.3: Кодирование и декодирование строки

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

```
>>> new_msg = "С Новым годом, друг!"
>>>
>>> key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_msg])
>>> print("Ключ: ", to_hex(key))
Ключ:  48 71 6c 6d 7a 77 36 6c 5a 70 51 58 34 4f 36 73 65 56 72 438
>>>
```

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.