Отчет по лабораторной работе №7

по дисциплине: Информационная безопасность

Логинов Егор Игоревич

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).

```
[7]  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).

```
[8]  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).

```
msg = "С Новым годом, друзья!"
key = generate_key(len(msg))
hex_key = to_hex(key)

enc_text = encoder([ord(i) for i in msg], [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])
```

```
[11] print("Ключ: ", hex_key)
     print("Зашифрованное сообщение: ", hex_text)
     print("Расшифрованный текст: ", decr_text)

     Ключ:  69 46 55 50 74 41 75 61 6f 59 6d 41 46 45 50 69 6a 6a 39 32 52 64
     Зашифрованное сообщение:  448 66 448 46e 446 40a 449 41 45c 467 459 47f 47a 69 70 45d 42a 429 40e 47e 41d 45
     Расшифрованный текст:  С Новым годом, друзья!
```

Рис. 4.3: Кодирование и декодирование строки

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

```
[12] new_msg = "С Новым годом, Логинов Егор!"

     key = encoder([ord(i) for i in enc_text], [ord(i) for i in new_msg])
     print("Ключ: ", to_hex(key))

     Ключ:  69 46 55 50 74 41 75 61 6f 59 6d 41 46 45 50 46 14 1a 36 43 23 477
```

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.