Лабораторная работа №7. Элементы криптографии. Однократное гаммирование

дисциплина: Информационная безопасность

Рыбалко Элина Павловна

Содержание

Список иллюстраций

Список таблиц

Цель работы

Освоить на практике применение режима однократного гаммирования.

Объект/Предмет исследования

Режим однократного гаммирования.

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.

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. [1] (#список-литературы).

Выполнение лабораторной работы

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 (см. рис. -@fig:001).

2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (см. рис. -@fig:001).

```
[22] import string
     import random

     def f_hkey(text):
       return ' '.join(hex(ord(i))[2:] for i in text)

     def f_key(size):
       return ''.join(random.choice(string.ascii_letters+string.digits) for _ in range(size))

     def enscryption(text, key):
       return ''.join(chr(a^b) for a,b in zip (text, key))

     def decryption(text, encrypt):
       return ''.join(chr(a^b) for a,b in zip (text, encrypt))
```

```
[23] #message = 'С Новым Годом, друзья!'
     message = (input("Введите сообщение: "))

     key = f_key(len(message))
     hex_key = f_hkey(key)
     print("Используемый ключ: ", key)
     print("Ключ в шестнадцатиричносм виде: ", hex_key)

     encrypt = enscryption([ord(i) for i in message], [ord(i) for i in key])
     hex_encrypt = f_hkey(encrypt)
     print("Зашифрованное сообщение", hex_encrypt)

     decrypt = enscryption([ord(i) for i in encrypt], [ord(i) for i in key])
     print("Расшифрованное сообщение", decrypt)
```

```
Введите сообщение: С Новым Годом, друзья!
Используемый ключ:  xgSgFPPeQu9PkAMfVwwnkq
Ключ в шестнадцатиричносм виде:  78 67 53 67 46 50 50 65 51 75 39 50 6b 41 4d 66 56 77 77 6e 6b 71
Зашифрованное сообщение 459 47 44e 459 474 41b 46c 45 442 44b 40d 46e 457 6d 6d 452 416 434 440 422 424 50
Расшифрованное сообщение С Новым Годом, друзья!
```

```
[24] compute_key = decryption([ord(i) for i in message], [ord(i) for i in encrypt])
     decrypt_compute_key = enscryption([ord(i) for i in encrypt], [ord(i) for i in key])
     print("Исходный ключ: ", key)
     print("Расшифровка открытого текста", decrypt_compute_key)
```

```
Исходный ключ:  xgSgFPPeQu9PkAMfVwwnkq
Расшифровка открытого текста С Новым Годом, друзья!
```

Рис. 1: Разработанное приложение

Вывод

Освоили на практике применение режима однократного гаммирования.

Список литературы

1. Лабораторная работа №7
2. Использование однократного гаммирования
3. Руководство по формуле Cmd Markdown
4. Руководство по оформлению Markdown файлов