Отчет по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Гурбангельдиев Мухаммет НФИбд-03-18

Содержание

Список иллюстраций

Цель работы

Освоить на практике применение режима однократного гаммирования

Последовательность выполнения работы

1. Блок функции для расчетов. (рис. -@fig:001)

```
In [23]: import string
         import random

In [24]: def hexx(text):
             return ''.join(hex(ord(i))[2:] for i in text)
         def gen_key(size):
             return ''.join(random.choice(string.ascii_letters + string.digits) for _ in range (size))
         def encrypted(text, key):
             return ''.join(chr(a^b) for a, b in zip(text, key))

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

Рис. 1: Блок функции для расчетов

2. Определил вид шифротекста при известном ключе и известном открытом тексте. (рис. -@fig:002)

```
In [25]: message= 'С Новым Годом, друзья!'

         key=gen_key(len(message))
         hex_key=hexx(key)
         print("Используемый ключ:", key)
         print("Ключ в шестнадцатиричном виде:", hex_key)
         encrypt = encrypted([ord(i) for i in message], [ord(i) for i in key])
         hex_encrypt=hexx(encrypt)
         print("Зашифрованное сообщение:", hex_encrypt)
         decryptt = encrypted([ord(i) for i in encrypt], [ord(i) for i in key])
         print("Расшифрованное сообщение:", decryptt)

         Используемый ключ: BuFNEGBLdBPrpd6v5bE0xi
         Ключ в шестнадцатиричном виде: 4275464e4547424c6442507270643676356245307869
         Зашифрованное сообщение: 4635545b47047740c47e6c47747c46444c44c4816442475421472747c43748
         Расшифрованное сообщение: С Новым Годом, друзья!

In [26]: compute_key = compute_key([ord(i) for i in message], [ord(i) for i in encrypt])
         decrypt_compute_key= encrypted([ord(i) for i in encrypt], [ord(i) for i in key])
         print("Вариант прочтения открытого тескта:", decrypt_compute_key)

         Вариант прочтения открытого тескта: С Новым Годом, друзья!
```

Рис. 2: Получение шифротекста

3. Определил ключ,с помощью которого шифротекст может быть преобразован в некоторый фрагменттекста,представляющий собой один из возможных вариантов прочтения открытого текста. (рис. -@fig:003)

```
In [26]: compute_key = compute_key([ord(i) for i in message], [ord(i) for i in encrypt])
         decrypt_compute_key= encrypted([ord(i) for i in encrypt], [ord(i) for i in key])
         print("Вариант прочтения открытого тескта:", decrypt_compute_key)

         Вариант прочтения открытого тескта: С Новым Годом, друзья!
```

Рис. 3: Прочтение открытого текста

Контрольные вопросы

1. Поясните смысл однократного гаммирования.

Гаммирование—метод симметричного шифрования,заключающийся в «наложении» последовательности,состоящей из случайных чисел,на открытый текст. Последовательность случайных чисел называется гаммапоследовательностью и используется для зашифровывания и расшифровывания данных.

2. Перечислите недостатки однократного гаммирования.

Ключ одного размера с сообщением,на один ключ используется только один текст.

3. Перечислите преимущества однократного гаммирования.

Простота и криптостойкость.

4. Почему длина открытого текста должна совпадать с длиной ключа?

Каждый символ текста попарно складывается с символом ключа.

5. Какая операция используется в режиме однократного гаммирования,назовите её особенности?

Сложение по модулю 2.Особенность в симметричности–оерация при повторном применении дает исходний результат.

6. Как по открытому тексту и ключу получить шифротекст?

Сложить по модулю 2 каждый символ открытого текста и ключа.

7. Как по открытому тексту и шифротексту получить ключ?

Сложить по модулю 2 каждый символ открытого текста и шифротекста.

8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

- полная случайность ключа;
- равенство длин ключа и открытого текста;
- однократное использование ключа.

Выводы

Освоил на практике применение режима однократного гаммирования.