Отчет по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Смородова Дарья Владимировна

2022 Oct 21st

Содержание

List of Tables

List of Figures

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Задание 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.
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[1]

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.

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком $\oplus$) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами: $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$.

Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой. Если известны ключ и открытый текст, то задача нахождения шифротекста заключается в применении к каждому символу открытого текста следующего правила:

[1] Методические материалы к лабораторной работе

$$C_i = P_i \oplus K_i$$

где C_i — i-й символ получившегося зашифрованного послания, P_i — i-й символ открытого текста, K_i — i-й символ ключа. Размерности открытого текста и ключа должны совпадать, и полученный шифротекст будет такой же длины.

Если известны шифротекст и открытый текст, то задача нахождения ключа решается также, а именно, обе части равенства необходимо сложить по модулю 2 с P_i:

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i$$

$$K_i = C_i \oplus P_i$$

Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов.

К. Шеннон доказал абсолютную стойкость шифра в случае, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения. Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P.

Необходимые и достаточные условия абсолютной стойкости шифра:

- полная случайность ключа;

- равенство длин ключа и открытого текста;

- однократное использование ключа.

4 Выполнение лабораторной работы

1. Напишем на Python функцию шифрования, которая определяет вид шифротекста при известном ключе и известном открытом тексте “С Новым Годом, друзья!” (рис. 4.1):

Figure 4.1: Функция шифрования данных

2. Посмотрим работу данной функции (рис. 4.2):

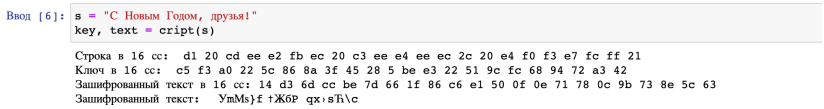

Figure 4.2: Результат работы функции, шифрующей данные

3. Напишем функцию дешифровки, которая определяет ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (рис. 4.3):

Ввод [5]:
```
def foundkey(text, new_text):
    print("Текст: ", text)
    print("Зашифрованнйй текст: ", new_text)
    text_16 = []
    for i in text:
        text_16.append(i.encode('cp1251').hex())
    print ("Текст в 16 сс: ", *text_16)
    array = []
    for i in new_text:
        array.append(i.encode('cp1251').hex())
    print("Текст в 16 сс: ", *array)
    key = [hex(int(i, 16)^int(j, 16))[2:] for (i, j) in zip(text_16, array)]
    print ("Ключ: ", *key)
    return key
```

Figure 4.3: Функция, дешифрующая данные

4. Посмторим на результаты функции дешифрования(рис. 4.4):

Ввод [7]:
```
found_key = foundkey(s, text)
```

```
Текст:  С Новым Годом, друзья!
Зашифрованнйй текст:  YmMs}f †ЖбР qx›sЋ\c
Текст в 16 сс:  d1 20 cd ee e2 fb ec 20 c3 ee e4 ee ec 2c 20 e4 f0
f3 e7 fc ff 21
Текст в 16 сс:  14 d3 6d cc be 7d 66 1f 86 c6 e1 50 0f 0e 71 78 0c
9b 73 8e 5c 63
Ключ:  c5 f3 a0 22 5c 86 8a 3f 45 28 5 be e3 22 51 9c fc 68 94 7
2 a3 42
```

Figure 4.4: Результат работы функции, дешифрующей данные

5.Сравним ключей, полученные с помощью первой и второй функций (рис. 4.5):

Ввод [8]:
```
if key == found_key:
    print("Ключ верный")
else:
    print("Ключ неверный")
```

```
Ключ верный
```

Figure 4.5: Сравнение ключей

Ключи совпали, программа работает верно.

5 Контрольные вопросы

1. Поясните смысл однократного гаммирования.
   Одократное 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.

2. Перечислите недостатки однократного гаммирования.
   Недостатки однократного гаммирования: Абсолютная стойкость шифра доказана только для случая, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения.

3. Перечислите преимущества однократного гаммирования.
   Преимущества однократного гаммирования: во-первых, такой способ симметричен, т.е. двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение; во-вторых, шифрование и расшифрование может быть выполнено одной и той же программой. Наконец, Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые после-

довательности K возможны и равновероятны, а значит, возможны и любые сообщения P.

4. Почему длина открытого текста должна совпадать с длиной ключа?
   Длина открытого текста должна совпадать с длиной ключа, т.к. если ключ короче текста, то операция XOR будет применена не ко всем элементам и конец сообщения будет не закодирован, а если ключ будет длиннее, то появится неоднозначность декодирования.

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?
   Операция XOR используется в режиме однократного гаммирования. Наложение гаммы по сути представляет собой выполнение побитовой операции сложения по модулю 2, т.е. мы должны сложить каждый элемент гаммы с соответствующим элементом ключа. Данная операция является симметричной, так как прибавление одной и той же величины по модулю 2 восстанавливает исходное значение.

6. Как по открытому тексту и ключу получить шифротекст?
   Получение шифротекста по открытому тексту и ключу:
   $C_i = P_i \oplus K_i$

7. Как по открытому тексту и шифротексту получить ключ?
   Получение ключа по окрытому тексту и шифротексту:

$$C_i \oplus P_i = P_i \oplus K_i \oplus P_i = K_i$$

$$K_i = C_i \oplus P_i$$

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

Необходимы и достаточные условия абсолютной стойкости шифра: полная случайность ключа; равенство длин ключа и открытого текста; однократное использование ключа.

6 Выводы

В ходе выполнения данной лабораторной работы, мы освоили на практике применение режима однократного гаммирования.

7 Список литературы

1. Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН”