Лабораторная работа №7

Дисциплина: информационная безопасность

Студент: Подорога Виктор Александрович

Цель работы

Освоить на практике применение режима однократного гаммирования.

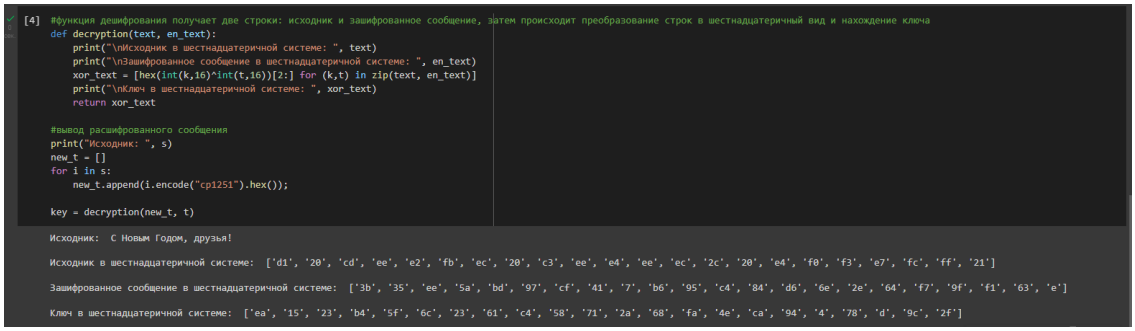
Выполнение лабораторной работы

1. Напишем код программы для зашифровки сообщения "С Новым Годом, друзья!":

```
[2] #импортируем библиотеки
    import numpy as np
    import operator as op
    import sys

    #функция шифрования, она получает на вход текст, затем переводит его в шестнадцатеричную систему счисления, создает ключ шифрования, считает зашифрованное сообщение и переводит в строку
    def encryption(text):
        print("Исходник: ", text)
        new_text = []
        for i in text:
            new_text.append(i.encode("cp1251").hex())
        print("\nИсходник в шестнадцатеричной системе: ", new_text)
        r = np.random.randint(0, 255, len(text))
        key = [hex(i)[2:] for i in r]
        print("\nКлюч в шестнадцатеричной системе: ", key)
        xor_text = [hex(int(k,16)^int(t,16))[2:] for (k,t) in zip(key, new_text)]
        print("\nЗашифрованное сообщение: ", xor_text)
        return key, xor_text

    #входное сообщение
    s = "С Новым Годом, друзья!"
    #вывод результата
    k, t = encryption(s)

    Исходник:  С Новым Годом, друзья!

    Исходник в шестнадцатеричной системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

    Ключ в шестнадцатеричной системе:  ['ea', '15', '23', 'b4', '5f', '6c', '23', '61', 'c4', '58', '71', '2a', '68', 'fa', '4e', 'ca', '94', '4', '78', 'd', '9c', '2f']

    Зашифрованное сообщение:  ['3b', '35', 'ee', '5a', 'bd', '97', 'cf', '41', '7', 'b6', '95', 'c4', '84', 'd6', '6e', '2e', '64', 'f7', '9f', 'f1', '63', 'e']
```

Рис. 1. Программа для зашифровки сообщения

2. Напишем код программы для расшифровки зашифрованного сообщения и нахождения ключа шифрования:



```
[4] #функция дешифрования получает две строки: исходник и зашифрованное сообщение, затем происходит преобразование строк в шестнадцатеричный вид и нахождение ключа
    def decryption(text, en_text):
        print("\nИсходник в шестнадцатеричной системе: ", text)
        print("\nЗашифрованное сообщение в шестнадцатеричной системе: ", en_text)
        xor_text = [hex(int(k,16)^int(t,16))[2:] for (k,t) in zip(text, en_text)]
        print("\nКлюч в шестнадцатеричной системе: ", xor_text)
        return xor_text

    #вывод расшифрованного сообщения
    print("Исходник: ", s)
    new_t = []
    for i in s:
        new_t.append(i.encode("cp1251").hex());

    key = decryption(new_t, t)

    Исходник:  С Новым Годом, друзья!

    Исходник в шестнадцатеричной системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

    Зашифрованное сообщение в шестнадцатеричной системе:  ['3b', '35', 'ee', '5a', 'bd', '97', 'cf', '41', '7', 'b6', '95', 'c4', '84', 'd6', '6e', '2e', '64', 'f7', '9f', 'f1', '63', 'e']

    Ключ в шестнадцатеричной системе:  ['ea', '15', '23', 'b4', '5f', '6c', '23', '61', 'c4', '58', '71', '2a', '68', 'fa', '4e', 'ca', '94', '4', '78', 'd', '9c', '2f']
```

Рис. 2. Программа для расшифровки и нахождения ключа

3. Проверим правильность нахождения ключа:

```
[6] #проверка на правильность найденного ключа
    if k == key:
        print("Ключ найден верно")
    else:
        print("Ключ наадйен неверно")

    Ключ найден верно
```

Рис. 3. Проверка

Поясните смысл однократного гаммирования.

Гаммирование – выполнение операции XOR между элементами гаммы и элементами подлежащего сокрытию текста. 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.

Перечислите 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.

Перечислите 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 последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P.

Почему длина открытого текста должна совпадать с длиной ключа?

Если ключ короче текста, то операция XOR будет применена не ко всем элементам и конец сообщения будет не закодирован. Если ключ будет длиннее, то появится неоднозначность декодирования.

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

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.

Как по открытому тексту и ключу получить шифротекст?

В таком случае задача сводится к правилу: $C_i = P_i \oplus K_i$, т.е. мы поэлементно получаем символы зашифрованного сообщения, применяя операцию исключающего или к соответствующим элементам ключа и открытого текста.

Как по открытому тексту и шифротексту получить ключ?

Подобная задача решается путем применения операции исключающего или к последовательностям символов зашифрованного и открытого сообщений: $K_i = P_i \oplus C_i$.

В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

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

Вывод

В ходе лабораторной работы я научился на практике применять режим однократного гаммирования.