Отчёт по лабораторной работе

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Сасин Ярослав игоревич НФИбд-03-18

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

2 Указание к работе

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. Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов. Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом.

3 Выполнение лабораторной работы

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Функция, которая определяет вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе Задаем алфавит из заглавных, строчных букв русского алфавита, !, ?, ., , и пробела. На вход поступает два открытых текста, в виде массива символов, и ключ — гамму. Анализируем длину текста, «растягиваем» гамму до нужного размера и выполняем посимвольное сложение. Функция выводит два шифротекста. (рис. -fig. 3.1)

```
В [1]:
import re

В [2]:
'ж','э','и','й','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','ъ','ы','ь','э','ю','я','!','?','.',',',' ']

В [6]:
def encrypt(text1, text2, gamma):
    textlen1 = len(text1)
    textlen2 = len(text2)
    gammalen = len(gamma)

    key_text = []
    for i in range (textlen1 // gammalen):
        for symb in gamma:
            key_text.append(symb)

    for i in range (textlen1 % gammalen):
        key_text.append(gamma[i])

    code1 = []
    code2 = []
    for i in range(textlen1):
        code1.append(alph[(alph.index(text1[i]) + alph.index(key_text[i])) % 71])
    for i in range(textlen1):
        code2.append(alph[(alph.index(text2[i]) + alph.index(key_text[i])) % 71])

    return(print(*code1,sep='')),(print(*code2,sep=''))

В [7]:
encrypt ('С Новым Годом, друзья!','С Левым Годом, друзья!', 'ааъайаааЁЫаааааааааааа')

С голым дедом, друзья!
С белым дедом, друзья!
```

Figure 3.1: первая функция

Функция, которая позволяет злоумышленнику прочитать оба текста, не зная ключа и не стремясь его определить. Если у злоумышленника есть оба шифротекста и один из открытых текстов, достаточно сложить по модулю 2 оба шифротекста и открытый текст, и получим второй открытый текст, не зная ключа. (рис. -fig. 3.2)

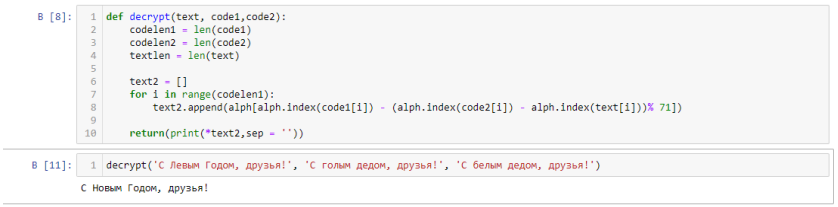

Figure 3.2: вторая функция

4 Выводы

В результате выполнения работы я освоил на практике применение шифрования (кодирования) различных исходных текстов одним ключом.