Отчёт по лабораторной работе

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

Назарьева Алена Игоревна НФИбд-03-18

Содержание

List of Figures

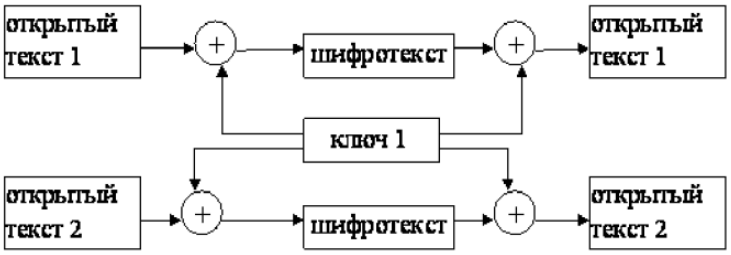
List of Tables

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 в соответствии с (рис. -fig. **??**), 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)

```
In [1]: import re

In [2]: alphabeth = ['А','Б','В','Г','Д','Е','Ё','Ж','З','И','Й','К','Л','М','Н','О','П','Р','С','Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ','Ы',

In [6]: def decrypt(text1, text2, gamma):
            text1Len = len(text1)
            text2Len = len(text2)
            gammaLen = len(gamma)

            keyText = []
            for i in range(text1Len // gammaLen):
                for symb in gamma:
                    keyText.append(symb)
            for i in range(text1Len % gammaLen):
                keyText.append(gamma[i])

            code1 = []
            code2 = []
            for i in range(text1Len):
                code1.append(alphabeth[(alphabeth.index(text1[i]) + alphabeth.index(keyText[i])) % 71])
            for i in range(text2Len):
                code2.append(alphabeth[(alphabeth.index(text2[i]) + alphabeth.index(keyText[i])) % 71])

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

In [7]: decrypt('С Новым Годом, друзья!','С Левым Годом, друзья!', 'ААЬАЙАААААААААААААААА')

        С Голым Годом, друзья!
        С Белым Годом, 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)

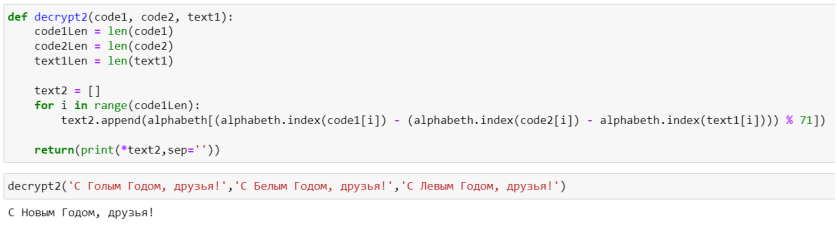


Figure 3.2: вторая функция

4 Выводы

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