Лабораторная работа № 7. Элементы криптографии. Однократное гаммирование

Радикорский Павел Михайлович НФИбд-03-18

11.12.2021

RUDN University, Moscow, Russian Federation

Цели и задачи

Цель

Освоить на практике применение режима однократного гаммирования

Выполнение

Процесс выполнения

Функция шифрования. Задаем алфавит из заглавных, строчных букв русского алфавита, !, ?, ., , и пробела. На вход поступает открытый текст, в виде массива символов, и ключ — гамму. Анализируем длину текста, «растягиваем» гамму до нужного размера и выполняем посимвольное сложение.

```
In [1]: import re

In [19]: alphabet = list ('АаБбВвГгДдЕеЁёЖжЗзИиЙйКкЛлМмНнОоПпРрСсТтУуФфХхЦцЧчШшЩщЪъЫыЬьЭэЮюЯя ,!')

In [22]: def encrypt (text, gamma):
             textLen = len(text)
             gammaLen = len(gamma)

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

             code = []
             for i in range (textLen):
                 code.append(alphabet[(alphabet.index(text[i]) + alphabet.index(keyText[i])) % 71])

             return (print(*code, sep = ''))

In [26]: encrypt('С Новым Годом, Друзья!', 'АЙАААААААААААААААААЙ')
         СЗНовым Годом, Друзьр!
```

Рис. 1: Функция шифрования

Процесс выполнения

Функция расшифрования Работает аналогично. «Растягиваем» гамму и выполняем посимвольное вычитание ее из текста.

```
In [45]: def decrypt (text, gamma):
             textLen = len(text)
             gammaLen = len(gamma)

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

             code = []
             for i in range (textLen):
                 code.append(alphabet[(alphabet.index(text[i]) - alphabet.index(keyText[i]) + 71) % 71])

             return (print(*code, sep = ''))

In [46]: decrypt('СЭНовым Годом, Друзьр!', 'АЙАААААААААААААААААу')

         С Новым Годом, Друзья!
```

Рис. 2: Функция расшифрования

Процесс выполнения

Функция, которая определяет ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста. Работает аналогично функции расшифрования, но на вход поступает не зашифрованный текст и ключ, а зашифрованный и открытый текст

In [99]:
```
def crypt (text, code):
    textLen = len(text)
    codeLen = len(code)

    keyText = []
    for i in range (textLen // codeLen):
        for symb in text:
            keyText.append(symb)
    for i in range (textLen & codeLen):
        keyText.append(code[i])

    gamma = []
    for i in range (textLen):
        gamma.append(alphabet[(alphabet.index(text[i]) - alphabet.index(keyText[i]) + 71) % 71])

    return (print(*gamma, sep = ''))
```

In [100]:
```
crypt('СЗНовым Годом, Друзьр!', 'С Новым Годом, Друзья!')
```

```
AAAAAAAAAAAAAAAAAAAAA
```

Рис. 3: Функция получения ключа

Выводы

Выводы

В результате выполнения работы я освоил на практике применение режима однократного гаммирования.