

Отчёт по лабораторной работе №8

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

Радикорский Павел Михайлович НФИбд-03-18

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Выводы	8

Список иллюстраций

0.1	первая функция	7
0.2	вторая функция	7

Список таблиц

Цель работы

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

Выполнение лабораторной работы

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитав 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 прочесть 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:002)

```

In [1]: import re

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

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(alphabet[(alphabet.index(text1[i]) + alphabet.index(keyText[i])) % 71])
    for i in range(text2len):
        code2.append(alphabet[(alphabet.index(text2[i]) + alphabet.index(keyText[i])) % 71])

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

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

```

Рис. 0.1: 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:003)

```

def decrypt2(code1, code2, text1):
    code1len = len(code1)
    code2len = len(code2)
    text1len = len(text1)

    text2 = []
    for i in range(code1len):
        text2.append(alphabet[(alphabet.index(code1[i]) - (alphabet.index(code2[i]) - alphabet.index(text1[i])) % 71)])

    return(print(*text2, sep=''))

decrypt2('С Голым Годом, друзья!', 'С Блжм Годом, друзья!', 'С Левым Годом, друзья!')
С Новым Годом, друзья!

```

Рис. 0.2: вторая функция

Выводы

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