

Отчёт по лабораторной работе

Элементы криптографии. Однократное
гаммирование

Сасин Ярослав Игоревич НФИбд-03-18”

Содержание

| | |
|---------------------------------------|----------|
| Цель работы | 4 |
| Указание к работе | 5 |
| Выполнение лабораторной работы | 6 |
| Выводы | 8 |

Список иллюстраций

| | | |
|----|---------------------------------|---|
| 1. | функция шифрования | 6 |
| 2. | Функция расшифрования | 6 |
| 3. | функция Z | 7 |

Цель работы

Освоить на практике применение режима однократного гаммирования

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.

Выполнение лабораторной работы

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно: 1. Определить вид шифротекста при известном ключе и известном открытом тексте. 2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

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. (рис. -@fig:001)

```

B [2]: 1 import re

B [5]: 1 , 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ь', 'ы', 'э', 'ю', 'я', '!', '?', '.', ',', ' '

B [9]: 1 def encrypt(text, gamma):
2     textlen = len(text)
3     gammalen = len(gamma)
4
5     key_text = []
6     for i in range(textlen // gammalen):
7         for symb in gamma:
8             key_text.append(symb)
9
10    for i in range(textlen % gammalen):
11        key_text.append(gamma[i])
12
13    code = []
14    for i in range(textlen):
15        code.append(alph[(alph.index(text[i]) + alph.index(key_text[i])) % 71])
16
17    return(print(*code, sep=''))

B [16]: 1 encrypt('С Новым Годом, друзья!', 'аааааааааааааааааааа')
С голым дедом, друзья!

```

Рис. 1: функция шифрования

Функция расшифрования Работает аналогично. «Растягиваем» гамму и выполняем посимвольное вычитание ее из текста. (рис. -@fig:002)

```

B [18]: 1 def decrypt(code, gamma):
2     codelen = len(code)
3     gammalen = len(gamma)
4
5     key_text = []
6     for i in range(codelen // gammalen):
7         for symb in gamma:
8             key_text.append(symb)
9
10    for i in range(codelen % gammalen):
11        key_text.append(gamma[i])
12
13    text = []
14    for i in range(codelen):
15        text.append(alph[(alph.index(code[i]) - alph.index(key_text[i]) + 71) % 71])
16
17    return(print(*text, sep=''))

B [19]: 1 decrypt('С голым дедом, друзья!', 'аааааааааааааааааааа')
С Новым Годом, друзья!

```

Рис. 2: Функция расшифрования

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 (рис. -@fig:003)

```
B [14]: 1 def crypt (code,text):
2         codelen = len(code)
3         textlen = len(text)
4
5         gamma = []
6         for i in range(codelen):
7             gamma.append((alpha.index(code[i]) - alpha.index(text[i]) + 71) % 71))
8
9         return(print("gamma, sep = '''))

```

```
B [15]: 1 crypt('С голым лицом, друзья!', 'С Новым Годом, друзья!')

```

Рис. 3: функция 3

Выводы

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