Отчёт по лабораторной работе

Элементы криптографии. Однократное гаммирование

Назарьева Алена Игоревна НФИбд-03-18

Содержание

1 Цель работы **5**

2 Указание к работе **6**

3 Выполнение лабораторной работы **7**

4 Выводы **9**

List of Figures

3.1 функция шифрования 7
3.2 Функция расшифрования 8
3.3 функция 3 8

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования

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.

3 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. 3.1)

Figure 3.1: функция шифрования

Функция расшифрования Работает аналогично. «Растягиваем» гамму и выполняем посимвольное вычитание ее из текста. (рис. -fig. 3.2)

```
[93]: def decrypt(code, gamma):
          codeLen = len(code)
          gammaLen = len(gamma)

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

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

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

[100]: decrypt('С Голым Годом, друзья!', 'ААъАЙАААААААААААААААА')

       С Новым Годом, друзья!
```

Figure 3.2: 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. 3.3)

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

       gamma = []
       for i in range(codeLen):
           text.append(alphabeth[(alphabeth.index(code[i]) - alphabeth.index(text[i]) + 71) % 71])

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

]: decrypt('С Голым Годом, друзья!!', 'С Белым Годом, друзья!')

   ААВЙАААААААААААААААААо
```

Figure 3.3: функция 3

4 Выводы

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