Отчёт по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Кондрашина Мария Сергеевна

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования

2 Теоретические сведения

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.[1]

3 Выполнение лабораторной работы. Создание программы

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. (fig. 3.1) - (fig. 3.2)

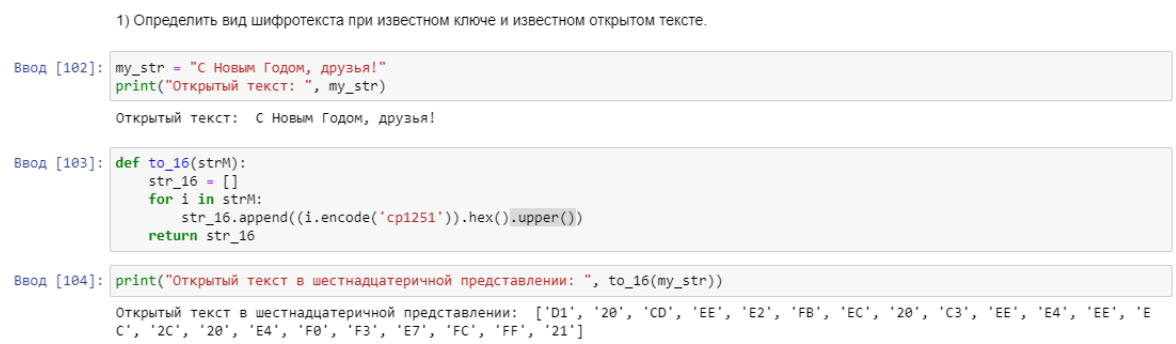

Figure 3.1: Работа с открытым текстом. Перевод в шестнадцатеричное представление.

Ввод [106]:
```
def cipher_text(strM, keyM):
    c_text = []
    for i,k in zip(strM, keyM):
        c_text.append(('{:02x}'.format(int(i,16) ^ int(k, 16))).upper())
    return c_text
```

Ввод [107]:
```
mc_text = cipher_text(to_16(my_str), key)
print("Зашифрованный текст в шестнадцатеричном представлении: ", mc_text)
```

```
Зашифрованный текст в шестнадцатеричном представлении:  ['45', '93', '3E', 'C0', 'B9', 'D0', '39', '84', '50', '02', 'D5', 'F
0', 'C9', '3A', 'B8', '7B', 'A5', 'B7', '9E', 'C0', 'C7', 'F3']
```

Ввод [108]:
```
new_text = [(bytes.fromhex(el)).decode('cp1251') for el in mc_text]
print("Зашифрованный текст: ", new_text)
```

```
Зашифрованный текст:  ['E', '“', '>', 'A', '№', 'P', '9', '„', 'P', '\x02', 'X', 'p', 'Й', ':', 'ё', '{', 'Ґ', '·', 'ћ', 'A',
'З', 'y']
```

Figure 3.2: Создание ключа и шифрование текста

2. Оределить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста,представляющий собой один из возможных вариантов прочтения открытого текста.(fig. 3.3)

2) Оределить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Ввод [109]:
```
print("Открытый текст: ", my_str)
print("Зашифрованный текст: ", new_text)
```

```
Открытый текст:  С Новым Годом, друзья!
Зашифрованный текст:  ['E', '“', '>', 'A', '№', 'P', '9', '„', 'P', '\x02', 'X', 'p', 'Й', ':', 'ё', '{', 'Ґ', '·', 'ћ', 'A',
'З', 'y']
```

Ввод [110]:
```
def find_key(str_16, c_text_16):
    key_new = []
    for i,j in zip(str_16, c_text_16):
        key_new.append(('{:02x}'.format(int(i,16) ^ int(j, 16))).upper())
    return key_new
```

Ввод [111]:
```
print("Найденный ключ: ", find_key(to_16(my_str), to_16(new_text)))
```

```
Найденный ключ:  ['94', 'B3', 'F3', '2E', '5B', '2B', 'D5', 'A4', '93', 'EC', '31', '1E', '25', '16', '98', '9F', '55', '44',
'79', '3C', '38', 'D2']
```

Ввод [115]:
```
if (find_key(to_16(my_str), to_16(new_text)) == key):
    print("Ключ совпал с ранее созданным")
else:
    print("Ключи различны")
```

```
Ключ совпал с ранее созданным
```

Figure 3.3: Определение ключа по отрытому и зашифрованному тексту

4 Выводы

- Выполнила лабораторную работу №7.
- Освоила на практике применение режима однократного гаммирования

5 Список литературы

1. Методические материалы курса. “Информационная безопасность компьютерных сетей” Кулябов Д. С., Королькова А. В., Геворкян М. Н.