Лабораторная работа 3

Попов Дмитрий Павлович, НФИмд-01-23

Содержание

List of Figures

List of Tables

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

дисциплина: Математические основы защиты информации и информационной безопасности

Преподователь: Кулябов Дмитрий Сергеевич

Студент: Попов Дмитрий Павлович

Группа: НФИмд-01-23

МОСКВА

2023 г.

1 Цель работы

Целью данной работы является приобретение практических навыков в шифровании гаммированием.[1]

2 Выполнение лабораторной работы

Требуется реализовать:

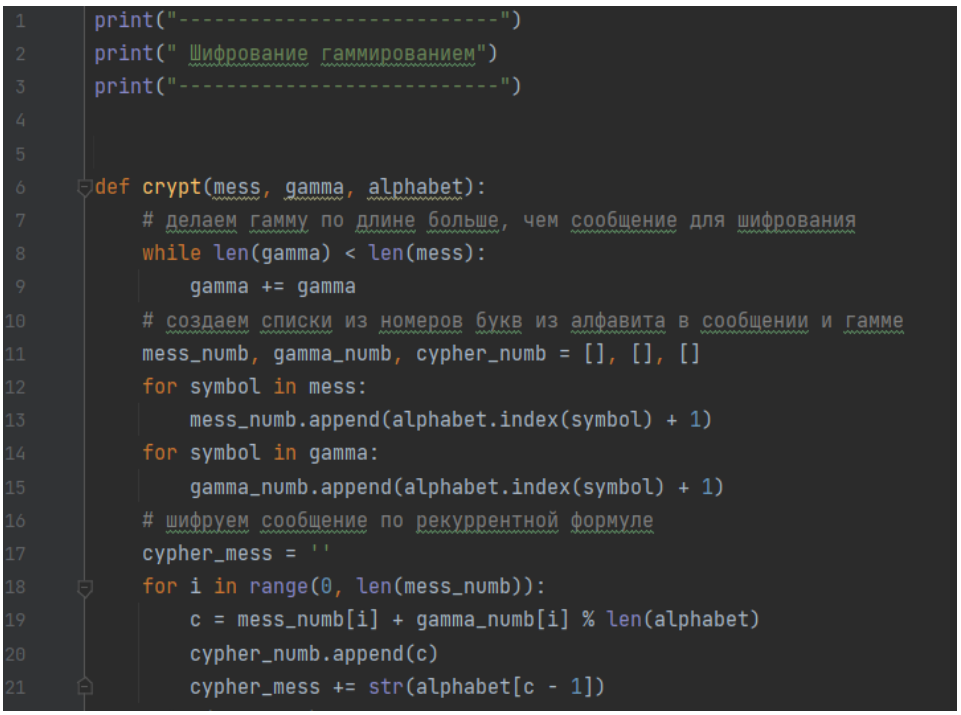
1. Шифрование гаммированием конечной гаммой

2.1 Шифрование гаммированием конечной гаммой

Гаммирование — процедура наложения при помощи некоторой функции F на исходный текст гаммы шифра, т.е. псевдослучайной последовательности (ПСП) с выходов генератора G. Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, т.е. известен алгоритм ее формирования.

Входной текст преобразовывается в номера букв из алфавита, соответствующим буквам фразы, затем так же преобразовывается гамма. Далее используется операция побитового сложения по модулю, равному длине алфавита. Получаем итоговую последовательность чисел, которая переводится в буквы в соответствии с алфавитом и получаем конечную криптограмму.

Чтобы реализовать программу был написал код на Python(fig. 2.1)(fig. 2.2):



```
print("---------------------------")
print(" Шифрование гаммированием")
print("---------------------------")


def crypt(mess, gamma, alphabet):
    # делаем гамму по длине больше, чем сообщение для шифрования
    while len(gamma) < len(mess):
        gamma += gamma
    # создаем списки из номеров букв из алфавита в сообщении и гамме
    mess_numb, gamma_numb, cypher_numb = [], [], []
    for symbol in mess:
        mess_numb.append(alphabet.index(symbol) + 1)
    for symbol in gamma:
        gamma_numb.append(alphabet.index(symbol) + 1)
    # шифруем сообщение по рекуррентной формуле
    cypher_mess = ''
    for i in range(0, len(mess_numb)):
        c = mess_numb[i] + gamma_numb[i] % len(alphabet)
        cypher_numb.append(c)
        cypher_mess += str(alphabet[c - 1])
```

Figure 2.1: gamma1

```
    print(alphabet)
    print("      Введенное сообщение: ", mess.upper(), mess_numb)
    print("   Зашированное сообщение: ", cypher_mess.upper(), cypher_numb)


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', 'ы', 'ь', 'э', 'ю', 'я']
mess = str(input("Введите предложение, которое нужно зашифровать: ")).lower().replace(' ', '')
# mess = 'приказ'.replace(' ', '')
gamma = str(input("Введите гамму: ")).lower()
# gamma = 'гамма'

crypt(mess=mess, gamma=gamma, alphabet=alphabet)
```

Figure 2.2: gamma2

Затем я запустил программу, ввел гамму и исходное сообщение. Получил зашифрованное сообщение. Вывод работы программы (fig. 2.3)

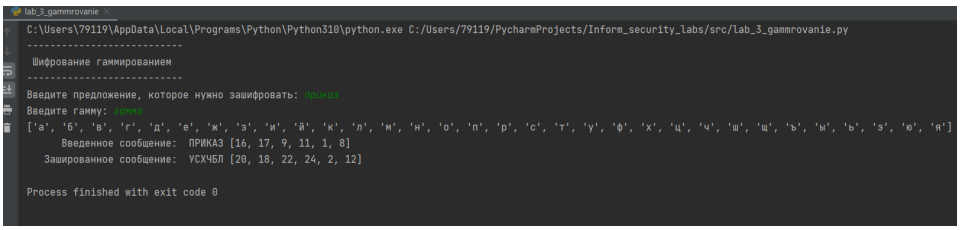


Figure 2.3: gamma_out

3 Выводы

В результате выполнения работы я освоил на практике применение шифрования с гаммированием конечной гаммой.

4 Список литературы

1. Методические материалы курса