Защита лабораторной работы №3

Шифрование гаммированием

Асеинова Е.В.

14 октября 2023

Российский университет дружбы народов, Москва, Россия

Цель выполнения лабораторной работы

- Освоение шифрования гаммированием
- Программная реализация алгоритма ширования гаммированием конечной гаммой

Теоретические сведения

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

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

Алгоритм поиска зашифрованного текста на основе принципа формирования шифрования гаммирования:

```
[44] def encrypt(message: str, gamma: str):
         alph = alphabet('eng')
         # if message.lower() not in alph:
         #   alph = alphabet('rus')
         length = len(alph)
         def gamma_en(letters_pair: tuple):
             ind = (letters_pair[0] + 1) + (letters_pair[1]+1) % length
             if ind > length:
               ind = ind-length
             return ind-1
         clear_message = list(filter(lambda s: s.lower() in alph, message))
         clear_gamma = list(filter(lambda s: s.lower() in alph, gamma))
         ind_message = list(map(lambda s: alph.index(s.lower()), clear_message))
         ind_gamma = list(map(lambda s: alph.index(s.lower()), clear_gamma))
         for i in range(len(ind_message) - len(ind_gamma)):
           ind_gamma.append(ind_gamma[i])
         print(f'{message.upper()} -> {ind_message}\n{gamma.upper()} -> {ind_gamma}')
         ind_encrypt = list(map(lambda s: gamma_en(s), zip(ind_message, ind_gamma)))
         print(f'Форма шифрования: {ind_encrypt}\n')
         return ''.join(list(map(lambda s: alph[s], ind_encrypt))).upper()
```

Figure 1: Реализация шифрования гаммирования

Результат выполнения лабораторной работы

Пример шифрования:

```
[40] def check(message:str, gamma: str):
       print(f'Результат шифрования: {encrypt(message, gamma)}')

[41] message = 'ПРИКАЗ'
     gamma = 'ГАММА'
     check(message, gamma)

     ПРИКАЗ -> [15, 16, 8, 10, 0, 7]
     ГАММА -> [3, 0, 12, 12, 0, 3]
     Форма шифрования: [19, 17, 21, 23, 1, 11]

     Результат шифрования: УСХЧБЛ

     message = 'HELLO DARKNESS MY OLD FRIEND'
     gamma = 'TALK'
     check(message, gamma)

     HELLO DARKNESS MY OLD FRIEND -> [7, 4, 11, 11, 14, 3, 0, 17, 10, 13, 4, 18, 18, 12, 24, 14, 11, 3, 5, 17, 8, 4, 13, 3]
     TALK -> [19, 0, 11, 10, 19, 0, 11, 10, 19, 0, 11, 10, 19, 0, 11, 10, 19, 0, 11, 10, 19, 0, 11, 10]
     Форма шифрования: [1, 5, 23, 22, 8, 4, 12, 2, 4, 14, 16, 3, 12, 13, 10, 25, 5, 4, 17, 2, 2, 5, 25, 14]

     Результат шифрования: BFXWIEMCEOQDMNKZFERCCFZO
```

Figure 2: Пример работы алгоритма

Выводы

1. Изучили шифрование гаммированием
2. Реализовали алгоритм шифрования гаммированием конечной гаммой на языке Python