

Лабораторная работа №3.

Шифрование гаммированием

*Дисциплина: Математические основы защиты информации
и информационной безопасности*

Студент: Банникова Екатерина Алексеевна

Группа: НФИмд-02-23

2023, Москва

Цели и задачи работы

Целью данной лабораторной работы является ознакомление с шифрованием гаммированием, – а так же реализация шифра на произвольном языке программирования.

Реализовать алгоритм шифрования гаммированием конечной гаммой.

Ход выполнения и результаты

Гаммирование Реализация

```
alphabet="АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"#задаем алфавит
alphabet_list=list(alphabet)#сделали алфавит списком
N=len(alphabet)#ввели размер алфавита
slovo="ПРИКАЗ"
key="ГАММА"
index_slovo=[]#ввели списки для индексов
index_key=[]#ввели списки для индексов
...
Находим индексы в соответствии с алфавитом
...
for i1 in slovo:
    index_slovo.append(alphabet.find(i1))
for i2 in key:
    index_key.append(alphabet.find(i2))
...
Находим индексы в соответствии с алфавитом (+смещение на 1 (из-за питона))
...
index_slovo_1=[]
index_key_1=[]
for j1 in range (0,len(index_slovo)):
    index_slovo_1.append(index_slovo[j1]+1)

for j2 in range (0,len(index_key)):
    index_key_1.append(index_key[j2]+1)
```

Figure 1: 1 часть программного кода реализации гаммирования конечной гаммой

Гаммирование. Реализация

```
...
Нахождение индексов букв будущего шифра (первые k символов, где k-длина ключа)
...
ciphered_text_indexes=[]#авели список для индексов будущего шифра
for l in range(len(index_key_1)):
    ciphered_text_indexes.append(index_slovo_1[l]+(index_key_1[l])%N)
...

Поиск новых индексов для шифра
...
difference=len(index_slovo_1)-len(index_key_1)#авели разницу в длине
index_key_2=0#авели индекс символа ключа, с которого будем начинать
index_slovo_2=len(index_key_1)#авели индекс символа слова, с которого будем начинать
while difference>0:
    ciphered_text_indexes.append(index_slovo_1[index_slovo_2]+(index_key_1[index_key_2])%N)
    difference=difference-1
    index_key_2+=1
    index_slovo_2=index_slovo_2-1
    if index_key_2==len(index_key_1):
        index_key_2=0
#ВНИМАНИЕ! ДЛЯ ТОГО, ЧТОБЫ СХОДИЛОСЬ С ОТВЕТОМ,
#ДАННЫМ В ЛАБОРАТОРНОЙ РАБОТЕ НЕОБХОДИМО ВЗЯТЬ АЛАВИТ БЕЗ БУКВЫ Ё (т.е. 32 символа)
...

Поиск шифра с помощью полученных индексов и алфавита
...
ciphered_text=[]
for i in range(len(ciphered_text_indexes)):
    ciphered_text.append(alphabet_list[ciphered_text_indexes[i]-1])#вспомнили что в питоне индексация с 1!
print(ciphered_text_indexes)
print('Криптограмма:',"".join(ciphered_text),'')
```

Figure 2: 2 часть программного кода реализации гаммирования конечной гаммой

☞ [20, 18, 22, 24, 2, 12]
Криптограмма: " УСХЧБЛ "

Figure 3: Результат шифрования сообщений с использованием гаммирования конечной гаммой

Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: я ознакомилась с шифрованием гаммированием, а так же мне удалось реализовать алгоритм шифрования конечной гаммой на языке программирования Python.