

Отчёт по лабораторной работе №3.

Шифрование гаммированием

*Дисциплина: Математические основы защиты информации
и информационной безопасности*

Студент: Румянцева Александра Сергеевна 1132223493

Группа: НПИМд-02-22

Преподаватель: д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

14 октября, 2022, Москва

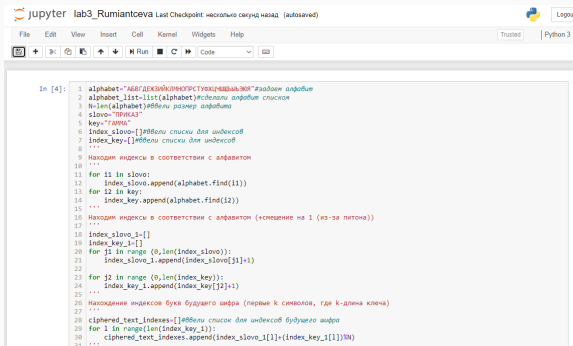
Цели и задание на лабораторную работу

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

Задание: Реализовать алгоритм шифрования гаммированием конечной гаммой.

1. Изучила теорию и указание к лабораторной работе.

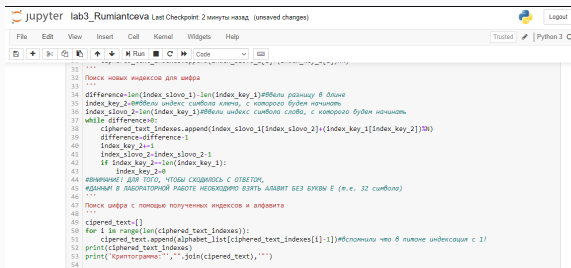
2. Реализация кода для гаммирования конечной гаммой.



```
In [4]: 1 alphabet="абвгдежзийклмнопрстуфхцчшщъыьэюя" #алфавит
2 alphabet_list=list(alphabet) #список алфавита
3 N=len(alphabet) #Величина алфавита
4 slovo="ПРИКАЗ"
5 key="ГАММА"
6 index_slovo=[] #Величина списка для индексов
7 index_key=[] #Величина списка для индексов
8 ...
9 Находим индексы в соответствии с алфавитом
10 ...
11 for i1 in slovo:
12     index_slovo.append(alphabet.find(i1))
13 for i2 in key:
14     index_key.append(alphabet.find(i2))
15 ...
16 Находим индексы в соответствии с алфавитом (+сдвиг на 1 (из-за питона))
17 ...
18 index_slovo_1=[]
19 index_key_1=[]
20 for j1 in range(0,len(index_slovo)):
21     index_slovo_1.append(index_slovo[j1]+1)
22 ...
23 for j2 in range(0,len(index_key)):
24     index_key_1.append(index_key[j2]+1)
25 ...
26 Нахождение индексов букв будущего шифра (первые k символов, где k-длина ключа)
27 ...
28 ciphered_text_indexes=[] #Величина списка для индексов будущего шифра
29 for l in range(len(index_key_1)):
30     ciphered_text_indexes.append(index_slovo_1[l]+(index_key_1[l])%N)
31 ...
```

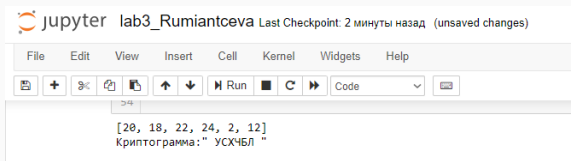
Figure 1: 1 часть программного кода реализации гаммирования конечной гаммой

Гаммирование. Реализация



```
31 ...
32 Поиск новых индексов для шифра
33 ...
34 difference=len(index_slovo_1)-len(index_key_1)#Ввели разницу в длину
35 index_key_2=0#Ввели индекс символа ключа, с которого будем начинать
36 index_slovo_2=len(index_key_1)#Ввели индекс символа слова, с которого будем начинать
37 while difference>0:
38     ciphered_text_indexes.append(index_slovo_1[index_slovo_2]+(index_key_1[index_key_2]))
39     difference=difference-1
40     index_key_2+=1
41     index_slovo_2=index_slovo_2-1
42     if index_key_2==len(index_key_1):
43         index_key_2=0
44 #ВНИМАНИЕ! ДЛЯ ТОГО, ЧТОБЫ СХОДИЛОСЬ С ОТВЕТОМ,
45 #ДАННЫМ В ЛАБОРАТОРНОЙ РАБОТЕ НЕОБХОДИМО ВЗЯТЬ АЛФАВИТ БЕЗ БУКВЫ Е (т.е. 32 символа)
46 ...
47 Поиск шифра с помощью полученных индексов и алфавита
48 ...
49 ciphered_text=[]
50 for i in range(len(ciphered_text_indexes)):
51     ciphered_text.append(alphabet_list[ciphered_text_indexes[i]-1])#Вспомнили что в пиконе индексация с 1/
52     print(ciphered_text_indexes)
53     print('Криптограмма:', ''.join(ciphered_text), '')
54
```

Figure 2: 2 часть программного кода реализации гаммирования конечной гаммой



The screenshot shows a Jupyter Notebook window titled "lab3_Rumiantceva". The top bar indicates "Last Checkpoint: 2 минуты назад (unsaved changes)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar contains icons for file operations, a "Run" button, and a dropdown menu set to "Code". The code cell contains the following text:

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

Figure 3: Результат шифрования сообщений с использованием гаммирования конечной гаммой

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: я ознакомилась с шифрованием гаммированием, а так же реализовала шифр на языке программирования Python.