

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

Лабораторная работа 3.

Дисциплина: Научное программирование

Студент: Румянцева Александра Сергеевна, 1132223493

Группа: НПМмд-02-22

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

Москва 2022

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Шифрование гаммированием	9
5	Библиография	11
6	Выводы	12

List of Figures

3.1	Рис. 1. Принципы алгоритма шифрования гаммированием	7
4.1	Рис. 2. 1 часть программного кода реализации гаммирования конечной гаммой	9
4.2	Рис. 3. 2 часть программного кода реализации гаммирования конечной гаммой	10
4.3	Рис. 4. Результат шифрования сообщений с использованием гаммирования конечной гаммой	10

List of Tables

1 Цель работы

Целью данной лабораторной работы является ознакомление с шифрованием гаммированием, а так же реализация шифрования гаммирования конечной гаммой.

2 Задание

Реализовать алгоритм шифрования гаммированием конечной гаммой.

3 Теоретическое введение

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле. Например, в поле Галуа суммирование принимает вид операции «исключающее ИЛИ (XOR)» [1].

В криптографии простой шифр XOR является разновидностью аддитивного шифра, алгоритма шифрования, который работает в соответствии с принципами [2]:

$$A \oplus 0 = A,$$

$$A \oplus A = 0,$$

$$A \oplus B = B \oplus A,$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C),$$

$$(B \oplus A) \oplus A = B \oplus 0 = B,$$

Figure 3.1: Рис. 1. Принципы алгоритма шифрования гаммированием

где \oplus обозначает операцию исключающей дизъюнкции (XOR). Эта операция иногда называется сложением по модулю 2 (или вычитанием, что идентично).

С помощью данной логики строка текста может быть зашифрована путем применения побитового оператора XOR к каждому символу с использованием заданного ключа. Для расшифровки результата достаточно повторно применить функцию XOR с ключом, чтобы снять шифр [2].

Шифры гаммирования (аддитивные шифры) являются самыми эффективными с точки зрения стойкости и скорости преобразований (процедур зашифрования и дешифрования). По стойкости данные шифры относятся к классу совершенных. Для зашифрования и дешифрования используются элементарные арифметические операции – открытое/зашифрованное сообщение и гамма, представленные в числовом виде, складываются друг с другом по модулю (mod) [3].

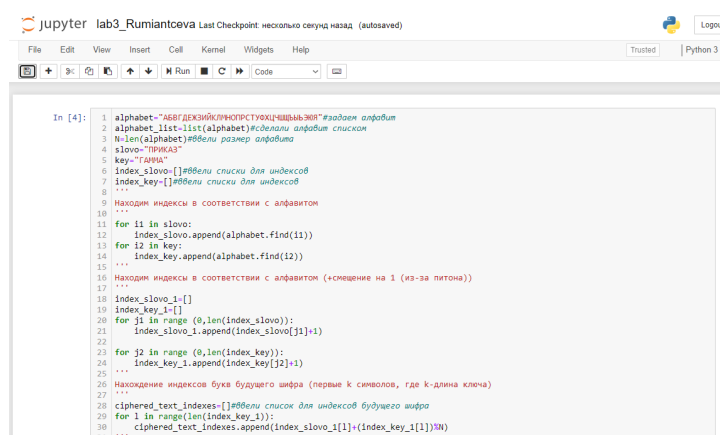
Пусть символам исходного алфавита соответствуют числа от 0 (А) до 32 (Я). Если обозначить число, соответствующее исходному символу, x , а символу ключа – k , то можно записать правило гаммирования следующим образом: $z = x + k \pmod{N}$, где z – закодированный символ, N - количество символов в алфавите, а сложение по модулю N - операция, аналогичная обычному сложению, с тем отличием, что если обычное суммирование дает результат, больший или равный N , то значением суммы считается остаток от деления его на N [4].

4 Выполнение лабораторной работы

Примечание: комментарии по коду представлены на скриншотах к каждому из проделанных заданий.

4.1 Шифрование гаммированием

В соответствии с заданием, была написана программа для шифрования гаммированием. Программный код представлен ниже (см. рис. 2,3).



```
In [4]: 1 alphabet="АБВГДЕЖЗИЙКЛМНОПРСТУХЦЧШЩЪЫЬЭЮЯ" # задан алфавит
2 alphabet_list=list(alphabet)#сделали алфавит список
3 N=len(alphabet)#белли размер алфавита
4 slovo="ПРИКАЗ"
5 key="ГАРМА"
6 index_slovo=[]#белли список для индексов
7 index_key=[]#белли список 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=[]#белли список 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 4.1: Рис. 2. 1 часть программного кода реализации гаммирования конечной гаммой

```

31 '''
32 Поиск новых индексов для шифра
33 '''
34 difference=len(index_slovo_1)-len(index_key_1)#Ввели разницу в длине
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]))%M
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 #внимай 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 питоне индексация с 1!
52 print(ciphered_text_indexes)
53 print('Криптограмма:', ''.join(ciphered_text), '')
54

```

Figure 4.2: Рис. 3. 2 часть программного кода реализации гаммирования конечной гаммой

Результаты выполнения программы представлены ниже (см. рис. 4). В качестве параметров системы были взяты данные из описательной части лабораторной работы портала ТУИС.

```

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

```

Figure 4.3: Рис. 4. Результат шифрования сообщений с использованием гаммирования конечной гаммой

5 Библиография

1. Википедия. Гаммирование [Электронный ресурс]. Википедия, свободная энциклопедия, 2022. URL: <https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5> (дата обращения: 14.11.2022).
2. Wikipedia. XOR cipher [Электронный ресурс]. Wikipedia, free Encyclopedia, 2022. URL: https://en.wikipedia.org/wiki/XOR_cipher (дата обращения: 14.11.2022).
3. Викторович А.В. 6.1 Шифры гаммирования [Электронный ресурс]. Учебная и научная деятельность Анисимова Владимира Викторовича, 2021. URL: <https://www.sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema6> (дата обращения: 14.11.2022).
4. Интерактивная система обучения. Методы шифрования с закрытым ключом [Электронный ресурс]. Электроника для всех, 2017. URL: <https://emkelektron.webnode.com/news/metody-shifrovaniya-zamenoj-podstanovkoj/> (дата обращения: 14.11.2022).

6 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.