

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

Отчёт по лабораторной работе №3

*Дисциплина: Математические основы защиты информации и
информационной безопасности*

Студент: Майорова О.А., НФИмд-02-21
Преподаватель: д.ф.-м.н. Кулябов Д.С.

Москва 2021

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	11
	Список литературы	12

List of Figures

4.1	Проверка функции 1	10
4.2	Проверка функции 2	10

List of Tables

1 Цель работы

Цель: Ознакомиться с шифрованием гаммированием.

2 Задание

Программно реализовать алгоритм шифрование 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 [1]. Шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите. Если в исходном алфавите, например, 33 символа, то сложение производится по модулю 33. Такой процесс сложения исходного текста и ключа называется в криптографии наложением гаммы. Если гамма короче, чем сообщение, предназначенное для зашифрования, гамма повторяется требуемое число раз. Пусть символам исходного алфавита соответствуют числа от 0 (А) до 32 (Я). Можно записать правило гаммирования следующим образом:

$$z = x + k(mod N),$$

где x - исходный символ, k - символ гаммы, 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 [2].

Различают гаммирование с конечной и бесконечной гаммами. В качестве конечной гаммы может использоваться фраза, в качестве бесконечной - последовательность, вырабатываемая генератором псевдослучайных чисел [3]. Наиболее часто на практике встречается двоичное гаммирование. При этом исполь-

зуется двоичный алфавит, а сложение производится по модулю два [2]. В том случае, если множеством используемых для шифрования знаков сообщения является текст, отличный от двоичного кода, то его символы и символы гаммы заменяются цифровыми эквивалентами, которые затем суммируются по модулю N [3]. Если ключ является фрагментом истинно случайной последовательности с равномерным законом распределения, причем его длина равна длине исходного сообщения и используется этот ключ только один раз, после чего уничтожается, такой шифр является абсолютно стойким, его невозможно раскрыть, даже если криптоаналитик располагает неограниченным запасом времени и неограниченным набором вычислительных ресурсов [3].

Операция сложения по модулю 2 часто обозначается \oplus , то есть можно записать:

$$z = x \oplus k.$$

Таким образом, при гаммировании по модулю 2 нужно использовать одну и ту же операцию как для зашифрования, так и для расшифрования. Это позволяет использовать один и тот же алгоритм, а соответственно и одну и ту же программу при программной реализации, как для шифрования, так и для расшифрования [2].

4 Выполнение лабораторной работы

Для выполнения лабораторной работы был выбран язык Python. Перед началом работы подключим библиотеку numpy:

```
import numpy as np
```

Реализуем шифрование гаммированием конечной гаммой в виде функции:

```
# C - открытый текст
```

```
# g - гамма
```

```
def gamm(C, g):
```

```
    C = C.replace(' ', '')
```

```
    for i in range(int(np.floor(len(C)/len(g) - 1))):
```

```
        g += ''.join(g)
```

```
    g += ''.join(g[:len(C)%len(g)])
```

```
    if ord('a') <= ord(C[0]) <= ord('z'):
```

```
        n = 26
```

```
        s = ord('a')
```

```
    if ord('a') <= ord(C[0]) <= ord('я'):
```

```
        n = 33
```

```
        s = ord('a')
```

```

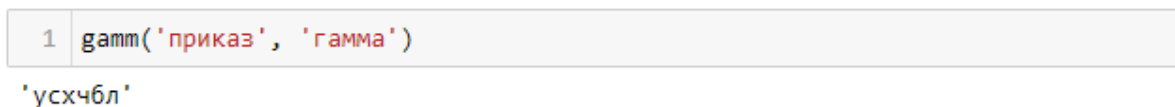
cypher = ''

for i in range(len(C)):
    c = (ord(C[i]) + ord(g[i]) - 2*s) % n + 1
    cypher += ''.join(chr(c + s))

return cypher

```

Результатом запуска функции для примера из задания к лабораторной работе будет рис. 4.1.



```

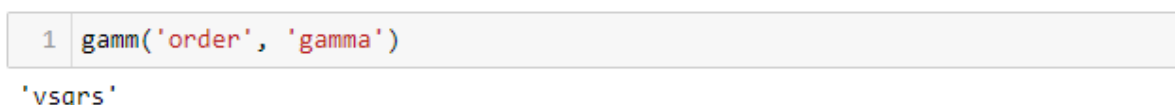
1 gamm('приказ', 'гамма')
'уsxчбл'

```

Figure 4.1: Проверка функции 1

Можно видеть, что полученное зашифрованное сообщение совпадает с приведённым в задании к лабораторной. Таким образом, шифрование функцией было произведено корректно.

Также проверим работу функции для открытого текста и гаммы на английском языке (рис. 4.2).



```

1 gamm('order', 'gamma')
'vsqrs'

```

Figure 4.2: Проверка функции 2

Видим, что функция отработала нормально и для второго возможного алфавита.

5 Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы. Было осуществлено знакомство с новым методом шифрования - шифрованием гаммированием. Также была получена реализация алгоритма шифрования гаммированием конечной гаммой для русского и английского алфавита нижних регистров на языке Python.

Список литературы

1. Гаммирование [Электронный ресурс]. Wikipedia, 2020. URL: <https://ru.wikipedia.org/w/index.php?title=Гаммирование&oldid=111027819>.
2. Методы гаммирования [Электронный ресурс]. НОУ «ИНТУИТ», 2021. URL: https://intuit.ru/studies/mini_mba/5398/courses/547/lecture/12373?page=4.
3. Хамидуллин Р. Р. М.А.В. Бригаднов И. А. Методы и средства защиты компьютерной информации. СПб.: СЗТУ, 2005.