

Отчёт по лабораторной работе 3

Шифрование гаммированием

Гаджиев Нурсултан Тофик оглы НФИ-01-22

Содержание

1	Цель работы	5
2	Теоретические сведения	6
3	Выполнение лабораторной работы	10
4	Выводы	14
5	Список литературы	15

List of Tables

List of Figures

3.1	Функция для кодирования текста шифром гаммированием конечной гаммой	10
3.2	Получение шифрования текста методом гаммированием конечной гаммой	11
3.3	Функция для расшифровки текста шифром гаммированием конечной гаммой	12
3.4	Получение расшифровки текста методом гаммированием конечной гаммой	13

1 Цель работы

Реализация алгоритма шифрования гаммированием конечной гаммой.

2 Теоретические сведения

Гаммирование – метод шифрования, основанный на “наложении” гамма-последовательности на открытый текст. Обычно это суммирование в каком-либо конечном поле (суммирование по модулю). Например, в поле $GF(2)$ такое суммирование принимает вид обычного “исключающего ИЛИ”. При расшифровке операция проводится повторно, в результате получается открытый текст.’

В этом способе 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.

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 . Например, пусть сложим по модулю 33 символы Г (3) и Ю (31):

$$3 + 31 \pmod{33} = 1,$$

то есть в результате получаем символ Б, соответствующий числу 1.

Наиболее часто на практике встречается двоичное гаммирование. При этом

используется двоичный алфавит, а сложение производится по модулю два.

Операция сложения по модулю два в алгебре логики называется также “исключающее ИЛИ” или по-английски XOR.

Рассмотрим пример. Предположим, нам необходимо зашифровать десятичное число 14 методом гаммирования с использованием ключа 12. Для этого вначале необходимо преобразовать исходное число и ключ (гамму) в двоичную форму: $14(10)=1110(2)$, $12(10)=1100(2)$. Затем надо записать полученные двоичные числа друг под другом и каждую пару символов сложить по модулю два. При сложении двух двоичных знаков получается 0, если исходные двоичные цифры одинаковы, и 1, если цифры разные

Сложим по модулю два двоичные числа 1110 и 1100:

Исходное число 1 1 1 0

Гамма 1 1 0 0

Результат 0 0 1 0

В результате сложения получили двоичное число 0010. Если перевести его в десятичную форму, получим 2. Таким образом, в результате применения к числу 14 операции гаммирования с ключом 12 получаем в результате число 2.

Каким же образом выполняется расшифрование? Зашифрованное число 2 представляется в двоичном виде и снова производится сложение по модулю 2 с ключом:

Зашифрованное число 0 0 1 0

Гамма 1 1 0 0

Результат 1 1 1 0

Переведем полученное двоичное значение 1110 в десятичный вид и получим 14, то есть исходное число.

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.

Операция сложения по модулю два очень быстро выполняется на компьютере (в отличие от многих других арифметических операций), поэтому наложение гаммы даже на очень большой открытый текст выполняется практически мгновенно.

Благодаря указанным достоинствам метод гаммирования широко применяется в современных технических системах сам по себе, а также как элемент комбинированных алгоритмов шифрования.

Сформулируем, как производится гаммирование по модулю 2 в общем случае:

- символы исходного текста и гамма представляются в двоичном коде и располагаются один под другим, при этом ключ (гамма) записывается столько раз, сколько потребуется;
- каждая пара двоичных знаков складывается по модулю два;
- полученная последовательность двоичных знаков кодируется символами алфавита в соответствии с выбранным кодом.

При использовании метода гаммирования ключом является последовательность, с которой производится сложение – гамма. Если гамма короче, чем сообщение, предназначенное для зашифрования, гамма повторяется требуемое число раз. Так в примере на рис. 2.6 длина исходного сообщения равна двенадцати байтам, а длина ключа – пяти байтам. Следовательно, для зашифрования гамма должна быть повторена 2 раза полностью и еще один раз частично.

При использовании генератора ПСП получаем бесконечную гамму. Однако, возможен режим шифрования конечной гаммы. В роли конечной гаммы может выступать фраза. Как и ранее, используется алфавитный порядок букв, т.е. буква «а» имеет порядковый номер 1, «б» - 2 и т.д.

Например, зашифруем слово «ПРИКАЗ» («17 18 10 12 01 09») гаммой «ГАММА» («04 01 14 14 01»). Будем использовать операцию побитового сложения по модулю 33 (mod33). Получаем:

$$c1 = 17 + 4(\text{mod}33) = 21, c4 = 12 + 14(\text{mod}33) = 26$$

$$c_2 = 18 + 1(\text{mod}33) = 19, c_5 = 1 + 1(\text{mod}33) = 2$$

$$c_3 = 10 + 14(\text{mod}33) = 24, c_6 = 9 + 4(\text{mod}33) = 13.$$

Криптограмма. «УСЦШБЛ» («20 18 22 24 02 12»).

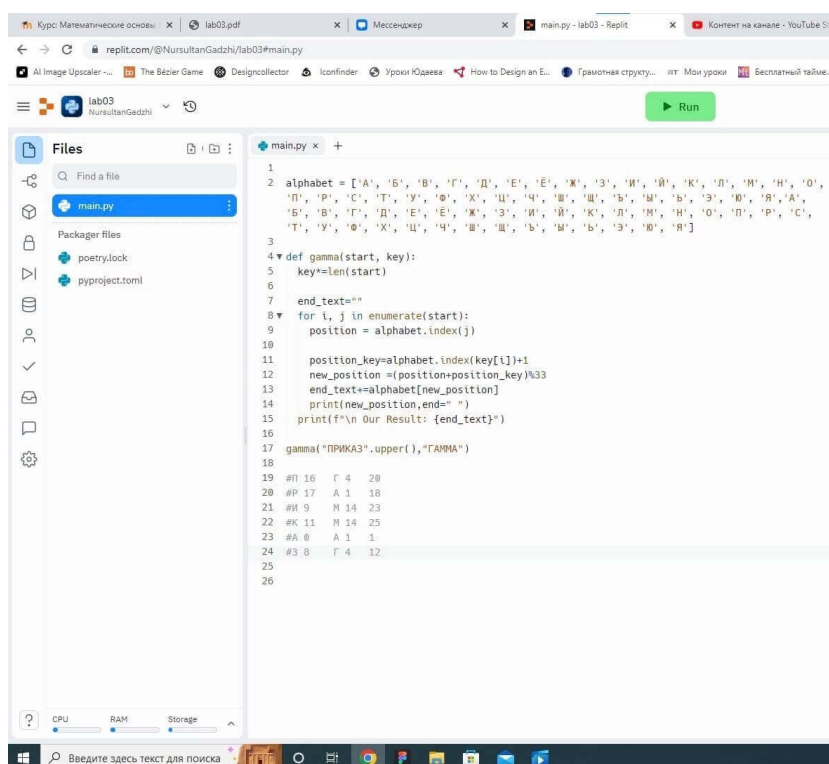
3 Выполнение лабораторной работы

1. Написал функцию гата для шифрования гаммированием конечной гаммой.
(рис. 3.1)

Сначала написал алфавит в виде списка.

Потом определил индекс каждой буквы в сообщении, аналогично ключу.

Как определил позицию, сложил на него позиции ключа. потом распечатал зашифрованный текст.



```
1
2 alphabet = ['А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О',
3             'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'А',
4             'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',
5             'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я']
6
7
8 4 def gamma(start, key):
9     key*=len(start)
10
11     end_text=""
12     for i, j in enumerate(start):
13         position = alphabet.index(j)
14
15         position_key=alphabet.index(key[i])+1
16         new_position =(position+position_key)%33
17         end_text+=alphabet[new_position]
18         print(new_position,end=" ")
19     print(f"\n Our Result: {end_text}")
20
21 gamma("ПРИКАЗ".upper(),"ГАММА")
22
23 #И 16 Г 4 20
24 #П 17 А 1 18
25 #И 9 М 14 23
26 #К 11 М 14 25
27 #А 0 А 1 1
28 #З 8 Г 4 12
29
30
31
```

Figure 3.1: Функция для кодирования текста шифром гаммированием конечной гаммой

Получил результат. (рис. 3.2)

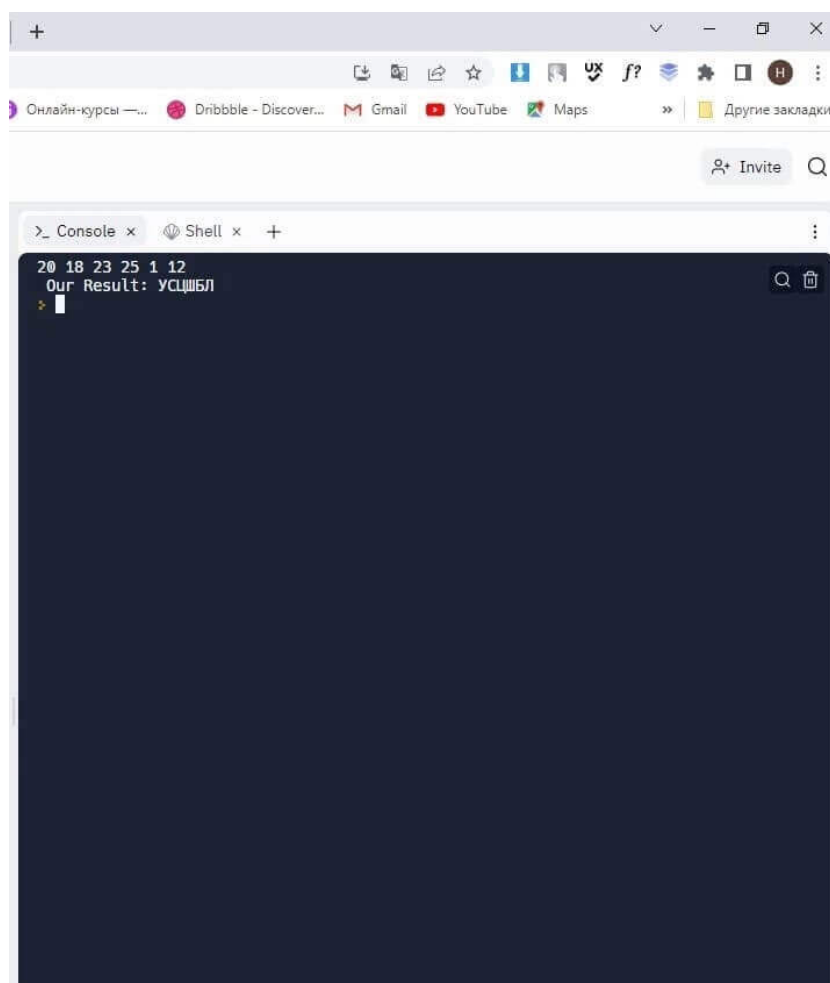


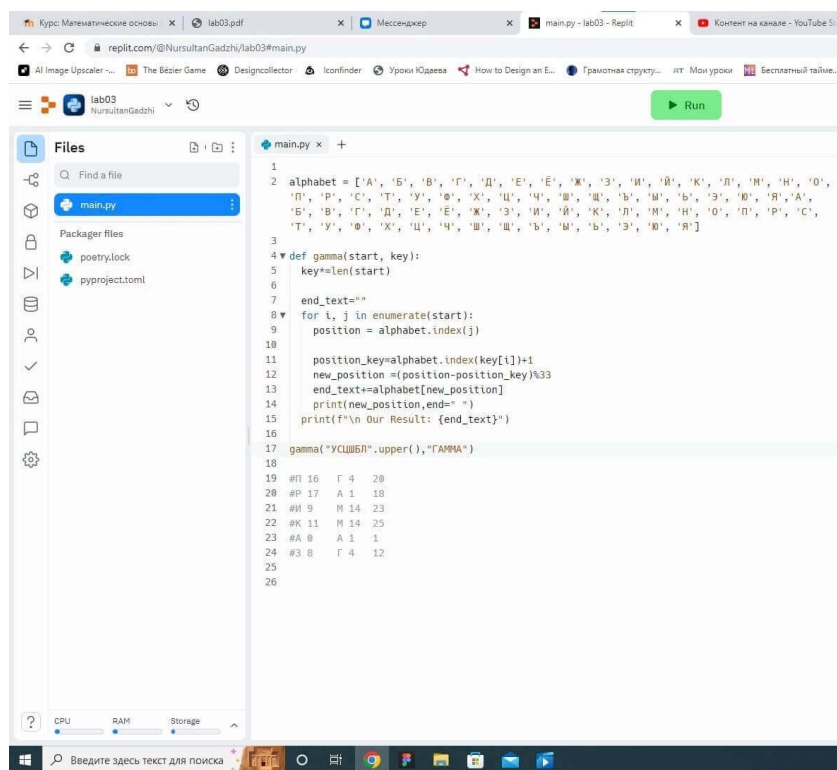
Figure 3.2: Получение шифрования текста методом гаммированием конечной гаммой

2. Написал функцию для расшифровки гаммированием конечной гаммой. (рис. 3.3)

Сначала написал алфавит в виде списка.

Потом определил индекс каждой буквы в сообщении, аналогично ключу.

Как определил позицию, вычитил из него позиции ключа. потом распечатал расшифровки текст.



```
1
2 alphabet = ['А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О',
3             'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я',
4             'Б', 'В', 'Г', 'Д', 'Е', 'Ё', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С',
5             'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я']
6
7 def gamma(start, key):
8     key=len(start)
9     end_text=""
10    for i, j in enumerate(start):
11        position = alphabet.index(j)
12        position_key=alphabet.index(key[i])+1
13        new_position =(position-position_key)%33
14        end_text+=alphabet[new_position]
15    print(new_position, end=" ")
16    print(f"\n Our Result: {end_text}")
17
18 gamma("УСШБЛ".upper(), "ГАММА")
19
20 #П 16 Г 4 20
21 #Р 17 А 1 18
22 #И 9 М 14 23
23 #К 11 М 14 25
24 #А 0 А 1 1
25 #З 8 Г 4 12
26
```

Figure 3.3: Функция для расшифровки текста шифром гаммированием конечной гаммой

Получил результат. (рис. 3.4)

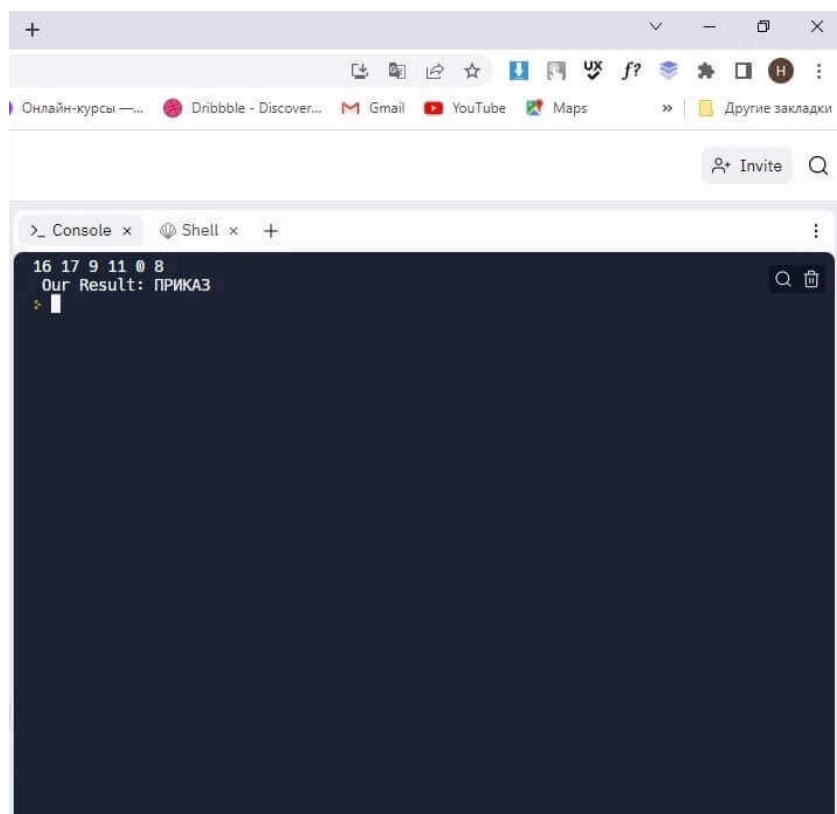


Figure 3.4: Получение расшифровки текста методом гаммированием конечной гаммой

4 Выводы

Реализовал алгоритм шифрования гаммированием конечной гаммой.

5 Список литературы

1. Методы гаммирования [Электронный ресурс] - Режим доступа: <https://www.intuit.ru/studies/courses/691/547/lecture/12373?page=4>. - Дата обращения: 05.04.2019.