

# **Лабораторная работа 3**

**Шифрование гаммированием**

Греков Максим Сергеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Описание метода</b>	<b>5</b>
2.1	Стойкость . . . . .	5
2.2	Пример шифрования . . . . .	6
<b>3</b>	<b>Реализация</b>	<b>7</b>
<b>4</b>	<b>Вывод</b>	<b>8</b>

# List of Figures

2.1	Гаммирование . . . . .	5
3.1	Реализация на Python . . . . .	7

# 1 Цель работы

- Ознакомиться с шифрованием гаммированием.
- Исследовать стойкость шифров, основанных на процедуре гаммирования.
- Реализовать алгоритм шифрования гаммированием конечной гаммой.

## 2 Описание метода

**Гаммирование** – метод последовательного симметричного шифрования, суть которого состоит в том, что символы шифруемого текста последовательно складываются с символами некоторой специальной последовательности, которая называется гаммой. (рис. 2.1)

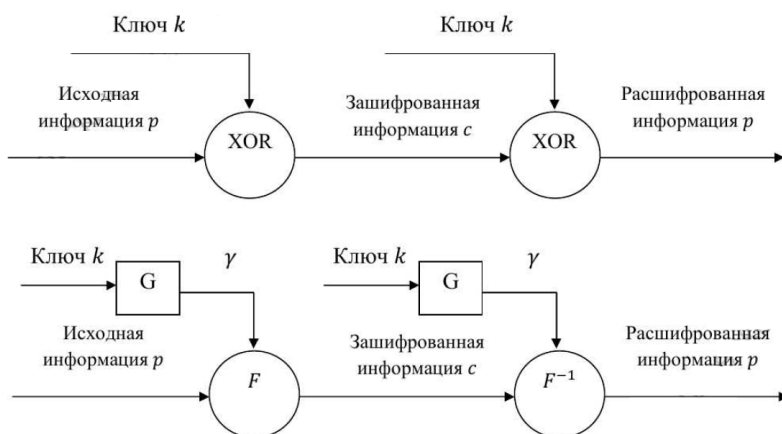


Figure 2.1: Гаммирование

### 2.1 Стойкость

Стойкость шифров, основанных на процедуре гаммирования, зависит от характеристик гаммы - длины и равномерности распределения вероятностей появления знаков гаммы.

При использовании генератора ПСП получаем бесконечную гамму.

Однако, возможен режим шифрования конечной гаммы.

## 2.2 Пример шифрования

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» (« 16 17 09 11 01 08») гаммой «ГАММА» («04 01 13 13 01»).

Будем использовать операцию побитового сложения по модулю 33 ( $\text{mod } 33$ ). (рис. 3.1) Получаем:

$$c_1 = 16 + 4(\text{mod}33) = 20$$

$$c_2 = 17 + 1(\text{mod}33) = 18$$

$$c_3 = 9 + 13(\text{mod}33) = 22$$

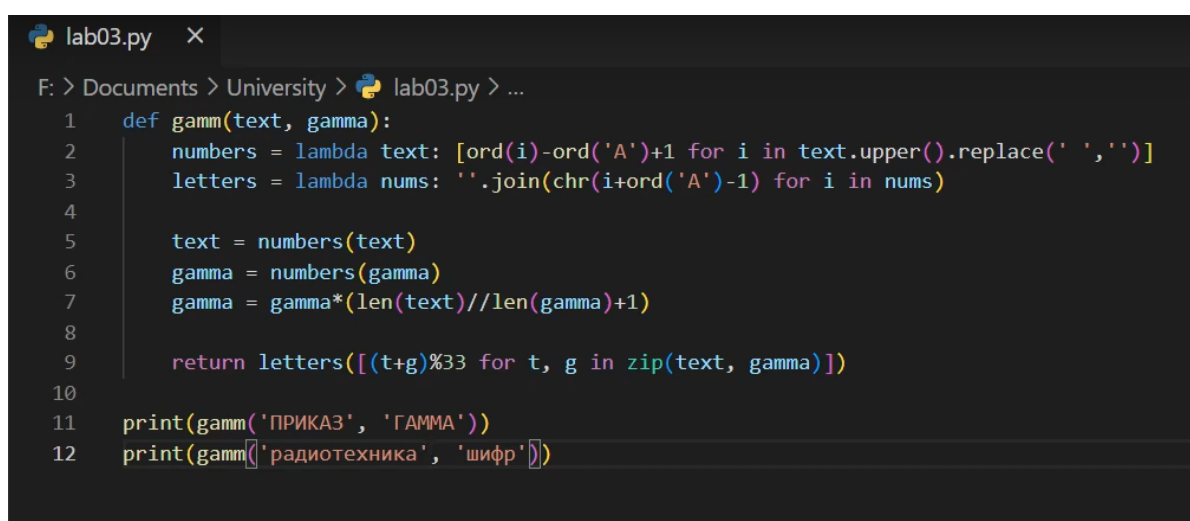
$$c_4 = 11 + 13(\text{mod}33) = 24$$

$$c_5 = 1 + 1(\text{mod}33) = 2$$

$$c_6 = 8 + 4(\text{mod}33) = 12$$

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

### 3 Реализация



```
lab03.py X
F: > Documents > University > lab03.py > ...
1  def gamm(text, gamma):
2      numbers = lambda text: [ord(i)-ord('A')+1 for i in text.upper().replace(' ', '')]
3      letters = lambda nums: ''.join(chr(i+ord('A')-1) for i in nums)
4
5      text = numbers(text)
6      gamma = numbers(gamma)
7      gamma = gamma*(len(text)//len(gamma)+1)
8
9      return letters([(t+g)%33 for t, g in zip(text, gamma)])
10
11 print(gamm('ПРИКАЗ', 'ГАММА'))
12 print(gamm(['радиотехника', 'шифр']))
```

Figure 3.1: Реализация на Python

- Результат 1: УСХЧБЛ
- Результат 2: ИЙЩЩЖЫЪЕЕСЯС

## 4 Вывод

- Ознакомились с шифрованием гаммированием.
- Исследовали стойкость шифров, основанных на процедуре гаммирования.
- Реализовали алгоритм шифрования гаммированием конечной гаммой.