

# **Математические основы защиты информации и информационной безопасности**

**Отче по лабораторной работе № 3**

Кейела Патачона НПИМд-02-21

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>5</b>
2.1	Гаммированием конечной гаммой . . . . .	5
<b>3</b>	<b>Выполнение работы</b>	<b>7</b>
3.1	Реализация шифра на языке Python . . . . .	7
3.2	Контрольный пример . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>9</b>
	<b>Список литературы</b>	<b>10</b>

# List of Figures

3.1	Работа алгоритма гаммированием конечной гаммой . . . . .	8
-----	--	---

# 1 Цель работы

Изучить алгоритм шифрования гаммированием конечной гаммой.

## 2 Теоретические сведения

### 2.1 Гаммированием конечной гаммой

Гаммирование — процедура наложения при помощи некоторой функции  $F$  на исходный текст гаммы шифра, т.е. псевдослучайной последовательности (ПСП) с выходов генератора  $G$ . Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, т.е. известен алгоритм ее формирования. Чаще Обычно в качестве функции  $F$  берется операция поразрядного сложения по модулю два или по модулю  $N$  ( $N$  число букв алфавита открытого текста).

Простейший генератор псевдослучайной последовательности можно представить рекуррентным соотношением:

$$y_i = ay_{(i-1)} + b \bmod(m), i = (1;m),$$

где  $y_i$  —  $i$ -й член последовательности псевдослучайных чисел,  $a, y_0, b$  — ключевые параметры. Такая последовательность состоит из целых чисел от 0 до  $m - 1$ . Если элементы  $y_i$  и  $y_j$  совпадут, то совпадут и последующие участки. Таким образом, ПСП является периодической. Знание периода гаммы существенно облегчает криптоанализ. Максимальная длина периода равна  $m$ . Для ее достижения необходимо удовлетворить следующим условиям:

1.  $b$  и  $m$  — взаимно простые числа;
2.  $a - 1$  делится на любой простой делитель числа  $m$ ;
3.  $a - 1$  кратно 4, если  $m$  кратно 4.

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. В роли конечной гаммы может выступать фраза. Как и ранее, используется алфавитный порядок букв, т. буква «а» имеет порядковый номер 1, «б» — 2 итд.

## 3 Выполнение работы

### 3.1 Реализация шифра на языке Python

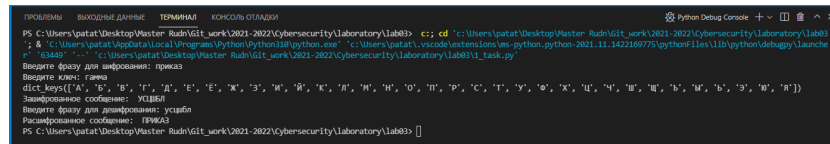
```
# Вводим алфавит и ключ
word_to_encode = input("Введите фразу для шифрования: ").upper()
key_word = input("Введите ключ: ").upper()
# Растягиваем ключ на длину слова
if len(key_word) < len(word_to_encode):
    k = (len(word_to_encode) % len(key_word))
    key_word = '' + key_word * (len(word_to_encode) // len(key_word)) + k
# Формируем алфавит и порядковый словарь
alphabet = 'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
alp_dict = {letter: idx + 1 for idx, letter in enumerate(alphabet)}
# процесс кодировки
encoded_word = ''
for word_letter, key_letter in zip(word_to_encode, key_word):
    encoded_word += list(alp_dict.keys())[alp_dict[word_letter] + alp_dict[key_letter] - 1]
print("Зашифрованное сообщение: ", encoded_word)
# процесс декодировки
word_to_decode = input("Введите фразу для дешифрования: ").upper()
decoded_word = ''
for word_letter, key_letter in zip(word_to_decode, key_word):
```

```

        decoded_word += list(alp_dict.keys())[alp_dict[word_letter] - alp_dict['A']
1]
print("Расшифрованное сообщение: ", decoded_word)

```

## 3.2 Контрольный пример



```

Python Debug Console
PS C:\Users\patat\Desktop\Master_Rudn\Git_work\2021-2022\Cybersecurity\Laboratory\lab03> cd 'C:\Users\patat\Desktop\Master_Rudn\Git_work\2021-2022\Cybersecurity\Laboratory\lab03'
'; & 'C:\Users\patat\AppData\Local\Programs\Python\Python38\python.exe' 'C:\Users\patat\vscode\extensions\ms-python.python-2021.11.1422169775\pythonFiles\lib\python\debugpy\launche
r' 'C:\Users\patat\Desktop\Master_Rudn\Git_work\2021-2022\Cybersecurity\Laboratory\lab03\1_task.py'
Введите фразу для шифрования: приказ
Введите ключ: gamma
dict_keys(['A', 'Б', 'В', 'Г', 'Д', 'Е', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ъ', 'Ы', 'Ь', 'Э', 'Ю', 'Я'])
Зашифрованное сообщение: УСПЕХ
Введите фразу для дешифрования: успех
Расшифрованное сообщение: ПРИКАЗ
PS C:\Users\patat\Desktop\Master_Rudn\Git_work\2021-2022\Cybersecurity\Laboratory\lab03>

```

Figure 3.1: Работа алгоритма гаммированием конечной гаммой



## 4 Выводы

Изучили алгоритм шифрования гаммированием конечной гаммой.

# Список литературы

1. ШИФРЫ ГАММИРОВАНИЯ
2. Гаммирование