

Министерство науки и высшего образования
Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

АЛГОРИТМЫ ДОНАУЧНОЙ КРИПТОГРАФИИ

Лабораторная работа №1
по дисциплине «Защита информации»

Обучающийся гр. 437-2:

_____ Тимофеев Д.О.

___. ___. 2021

Руководитель

Д.т.н. каф. АСУ:

_____ Горитов А.Н.

___. ___. 2021

1. Цель лабораторной работы.

Познакомиться и научиться работать с алгоритмами донаучной криптографии.

2. Индивидуальное задание.

Вариант 2. Перестановочный шифр.

Написать программу, позволяющую зашифровать и расшифровать сообщения с использованием перестановочного шифра. Количество столбцов и строк задается в программе. Входные и выходные данные запишите в файл типа .txt.

3. Описание алгоритма шифрования.

Перестановочный шифр - это метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами.

В данной лабораторной работе шифр перестановками основан на заполнении таблицы в одном порядке (слева-направо сверху-вниз) и считыванием в другом (сверху-вниз слева-направо).

4. Листинг программы.

Листинг программы представлен в приложении А

5. Примеры работы программы.

Примеры работы программы представлены в приложении А

6. Вывод о проделанной работе.

В ходе выполнения лабораторной работы №1, я освоил на практике алгоритм донаучной криптографии – Перестановочный шифр.

Приложение А

Вариант	Выполнил	Группа
7	Тимофеев	437-2

Вариант 7. Перестановочный шифр

Напишите программу, позволяющую зашифровать и расшифровать сообщения с использованием перестановочного шифра. Количество столбцов и строк задается в программе. Входные и выходные данные запишите в файл типа .txt.

Перестановочный шифр:

Перестановочный шифр - это метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами. Элементами текста могут быть отдельные символы (самый распространённый случай), пары букв, тройки букв, комбинирование этих случаев и так далее.

Типичными примерами перестановки являются анаграммы.

Описание алгоритма:

Алгоритм шифровки:

- создаём таблицу с заданным количеством столбцов
- записываем символы входной строки в таблицу построчно (слева-направо сверху-вниз), если в строке таблице нет свободных ячеек продолжаем писать с новой строки
- после того как входящая строка записана - считываем символы по столбцам (сверху-вниз слева-направо)

Алгоритм дешифровки:

- создаём таблицу с заданным количеством столбцов
- рассчитываем количество полных столбцов
- в полных столбцах записываем элементы выходной строки по столбцам (сверху-вниз слева-направо)
- в неполных столбцах оставляем 1 пустой символ
- считываем (слева-направо сверху-вниз) (игнорируя пустые символы в конце)

Описываем класс для шифровки и дешифровки шифра перестановками

У класса есть:

- **Конструктор:** Принимает количество столбцов в таблице перестановок
- **метод encode:** Метод для шифровки сообщения. Принимает текст который необходимо зашифровать
- **метод decode:** Метод для расшифровки сообщения. Принимает текст который необходимо расшифровать

In [1]:

```
from pprint import pprint
```

```

class Permutations:
    def __init__(self, table_cols: int):
        self._T = table_cols

    def encode(self, text: str):
        K: int = len(text) // self._T # Количество полных столбцов
        D: int = len(text) % self._T # Количество элементов в неполной строке
        C: list[str] = []
        for i in range(self._T):
            if D > i:
                d = 1
            else:
                d = 0
            for j in range(K+d):
                C.append(text[j*self._T+i])

        return ''.join(C)

    def decode(self, text):
        K: int = len(text) // self._T # Количество полных столбцов
        D: int = len(text) % self._T # Количество элементов в неполной строке
        M: list[str] = []
        for i in range(K):
            for j in range(D+1):
                k = i+j*(K+1)
                M.append(text[k])
            r = D*(K+1)
            for j in range(1, self._T-D):
                k = i+r+j*K
                M.append(text[k])
        for j in range(D):
            M.append(text[(j+1)*(K+1)-1])
        return ''.join(M)

```

Считываем текст который хотим зашифровать из файла input.txt

```

In [2]: input_name = 'input.txt'

with open(input_name, 'r', encoding='utf-8') as file:
    исходный_текст = file.read()

print(f"Содержание {input_name}:")
pprint(исходный_текст)
исходный_текст

```

Содержание input.txt:
 '123456789 Вот он текст который необходимо зашифровать'

Out[2]: '123456789 Вот он текст который необходимо зашифровать'

Создаём объект Шифратора-Дешифратора

```

In [3]: количество_столбцов = 5
шифр_перестановками = Permutations(количество_столбцов)

```

Шифруем строку и смотрим что получилось

```

In [4]: зашифрованный_текст = шифр_перестановками.encode(исходный_текст)
pprint(зашифрованный_текст)
зашифрованный_текст

```

'16Внст хоиа27о тоно фт38тт редзрь49 екьюиао5 окойбмшв'

Out[4]: '16Внст хоиа27о тоно фт38тт редзрь49 екьюиао5 окойбмшв'

Теперь дешифруем строку обратно

```
In [5]: расшифрованный_текст = шифр_перестановками.decode(зашифрованный_текст)
pprint(расшифрованный_текст)
расшифрованный_текст
```

'123456789 Вот он текст который необходимо зашифровать'

Out[5]: '123456789 Вот он текст который необходимо зашифровать'

Запустим ещё раз с другим файлом

```
In [6]: input_name = 'input2.txt'

with open(input_name, 'r', encoding='utf-8') as file:
    исходный_текст = file.read()

print(f"Содержание {input_name}:")
pprint(исходный_текст)
исходный_текст
```

Содержание input2.txt:
('Мы знаем, что такое байты,\n'
'Система поиска и сайты,\n'
'Умеем файлы создавать,\n'
'Картинки «мышкой» рисовать,\n'
'\n'
'И через интернет общаться,\n'
'И в алгоритмах разбираться,\n'
'И понимать язык программ.\n'
'\n')

Out[6]: 'Мы знаем, что такое байты,\nСистема поиска и сайты,\nУмеем файлы создавать,\nКартинки «мышкой»
рисовать,\n\nИ через интернет общаться,\nИ в алгоритмах разбираться,\nИ понимать язык програм
м.\n\n'

Создаём объект Шифратора-Дешифратора

Изменим количество столбцов

```
In [7]: количество_столбцов = 10
шифр_перестановками = Permutations(количество_столбцов)
```

Шифруем строку и смотрим что получилось

```
In [8]: зашифрованный_текст = шифр_перестановками.encode(исходный_текст)
pprint(зашифрованный_текст)
зашифрованный_текст
```

('Мчбта\n'
'лтк», оИтрпзмытае Уыьи \n'
'иб маоым оймим , р\n'
'нщватнк.з та ес\n'
'«иИта хьи \n'
'нты сеоКмс ета смп\n'
'аа,памзаыочрьлрярек\n'
'ой дршвенсга,томоСитфаткареяз\n'
'ьг,еисыавиотет,рби р ск,йанийз \n'
'ии яа')

Out[8]: 'Мчбта\нлтк», оИтрпзмытае Уыьи \ниб маоым оймим , р\ннщватнк.з та ес\н«иИта хьи \ннты сеоКмс ет
а смп\наа,памзаыочрьлрярек\ной дршвенсга,томоСитфаткареяз\ньг,еисыавиотет,рби р ск,йанийз \н
ии яа'

Теперь дешифруем строку обратно

```
In [9]: расшифрованный_текст = шифр_перестановками.decode(зашифрованный_текст)
pprint(расшифрованный_текст)
расшифрованный_текст
```

```
('Мы знаем, что такое байты,\n'
'Система поиска и сайты,\n'
'Умеем файлы создавать,\n'
'Картинки «мышкой» рисовать,\n'
'\n'
'И через интернет общаться,\n'
'И в алгоритмах разбираться,\n'
'И понимать язык программ.\n'
'\n')
```

```
Out[9]: 'Мы знаем, что такое байты,\nСистема поиска и сайты,\nУмеем файлы создавать,\nКартинки «мышкой»
рисовать,\n\nИ через интернет общаться,\nИ в алгоритмах разбираться,\nИ понимать язык програм
м.\n\n'
```