Пшенко Артем, ФИТ 3-4

Информационная безопасность

Отчет по лабораторной работе № 5.

Исследование криптографических шифров на основе

перестановки символов

Вариант 9

В данной лабораторной работе необходимо было реализовать спиральную матрицу из исходного сообщения. Исходный код приведен в листинге 1.

Листинг 1. Исходный код спиральной матрицы

|  |
| --- |
|  |

Следует отметить, что для дешифрования ис

Результат работы алгоритма приведен рисунке 1.

|  |
| --- |
|  |

Рисунок 1. Результат спиральной перестановки

Также необходимо было замерить время выполнения данного алгоритма. Для этого был написан декоратор. Его исходный код приведен в листинге 2.

Листинг 2. Исходный код декоратора

|  |
| --- |
|  |

Алгоритм перевода текста из спиральной матрицы в «обычный» вид приведен в листинге 3.

Листинг 3. Исходный код возврата текста от спиральной матрицы к обычному виду

|  |
| --- |
|  |

Результат работы алгоритма приведен на рисунке 2.

|  |
| --- |
|  |

Рисунок 2. Результат возврата от спиральной матрицы

Частотная характеристика данного сообщения приведена на рисунке 3.

|  |
| --- |
|  |

Рисунок 3. Частотная характеристика сообщения

После этого необходимо было реализовать алгоритм множественной перестановки с ключевыми словами (собственная фамилия и имя). Данный алгоритм работает следующим образом: берется исходное сообщение, из него формируется матрица. Далее генерируются индексы для строк и столбцов матрицы от 1 до n для столбцов и строк, где n – количество строк/столбцов. После этого переставляются столбцы таким образом, чтобы индексы выстроились по порядку (т.е. столбец с индексом 1 станет первым, с индексом 2 – вторым и т.д.). Далее аналогичным образом переставляются строки. Итоговое зашифрованное сообщение состоит из символов, которые берутся по столбцам.

Для расшифрования сообщения используется реверсный алгоритм: сначала переставляем строки в исходное положения исходя из индексов, а далее – столбцы. Реализация алгоритма приведена в листинге 4.

Листинг 4. Исходный код алгоритма множественной перестановки.

|  |
| --- |
| text = "пшенко артем федорович"  num\_rows = 4  num\_cols = 6  if len(text) < num\_rows \* num\_cols:      text += ' ' \* (num\_rows \* num\_cols - len(text))  text\_list = list(text)  matrix\_text = np.array(text\_list).reshape(num\_rows, num\_cols)  print('Исходная матрица:\n', matrix\_text, '\n')  row\_indices = np.random.permutation(num\_rows) + 1  # индексы строк  col\_indices = np.random.permutation(num\_cols) + 1  # индексы столбцов  print('Индексы строк: ', row\_indices, '\n')  print('Индексы столбцов: ', col\_indices, '\n')  matrix\_text = matrix\_text[:, np.argsort(col\_indices)]  print('столбцы:\n', matrix\_text, '\n')  matrix\_text = matrix\_text[np.argsort(row\_indices), :]  print('строки:\n', matrix\_text, '\n')  print('Матрица после перестановки строк и столбцов:\n', matrix\_text)  final\_string = ''.join(matrix\_text.flatten(order='F'))  print('Зашифрованное сообщение: ', final\_string, '\n')  matrix\_final = np.array(list(final\_string)).reshape(num\_rows, num\_cols, order='F')  print('Матрица из итоговой строки:\n', matrix\_final, '\n')  matrix\_final = matrix\_final[np.argsort(row\_indices + 1), :]  # Перестановка строк  matrix\_final = matrix\_final[:, np.argsort(np.argsort(col\_indices + 1))]  # Перестановка столбцов  print('Матрица после возвращения к исходному состоянию:')  print(matrix\_final)  decrypted\_string = ''.join(matrix\_final.flatten(order='C'))  print('Расшифрованное сообщение: ', decrypted\_string, '\n') |

Результат работы алгоритма приведен на рисунке 4.

|  |
| --- |
|  |

Рисунок 4. Результат работы алгоритма множественной перестановки