МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5 НА ТЕМУ:**

**Исследование криптографических шифров на основе перестановки символов**

Выполнил студент 3 курса 6 группы

Подобед Владислав

Минск 2024

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации перестановочных шифров.

**Теоретические сведения**

Криптоанализ – это раздел криптологии, занимающийся методами взлома шифров или методами организации криптографических атак на шифры.

Сущность перестановочного шифрования состоит в том, что исходный текст (*М*) и зашифрованный текст (*С*) основаны на использовании одного и того же алфавита, а тайной или ключевой информацией является алгоритм перестановки.

Шифры перестановки относятся к классу симметричных. Элементами текста могут быть отдельные символы (самый распространенный случай), пары, тройки букв и т.д.

В классической криптографии шифры перестановки делятся на два подкласса:

* шифры простой, или одинарной, перестановки – при зашифровании символы открытого текста *Мi* перемещаются с исходных позиций в новые (в шифртексте *Сi*) один раз;
* шифры сложной, или множественной, перестановки – при зашифровании символы открытого текста *Мi* перемещаются с исходных позиций в новые (в шифртексте *Сi*) несколько раз.

Среди шифров рассматриваемого подкласса иногда выделяют ***шифры простой перестановки (или перестановки без ключа)***. Символы открытого текста *Мi* перемешиваются по каким-либо правилам. Формально каждое из таких правил может рассматриваться в качестве ключа.

***Шифры простой блочной перестановки*** строятся по тем же правилам, что и шифры простой перестановки. Блок должен состоять из 2 или более символов. Если общее число таких символов в сообщении не кратно длине сообщения, то последний блок можно дополнить произвольными знаками.

Основой современных ***шифров маршрутной перестановки*** является геометрическая фигура, обычно прямоугольник или прямоугольная матрица. В ячейки этой фигуры по определенному маршруту (слева направо, сверху вниз или каким-либо иным образом) записывается открытый текст. Для получения шифрограммы нужно записать символы этого сообщения в иной последовательности, т. е. по иному маршруту.

***Шифр вертикальной перестановки*** является разновидностью шифра маршрутной перестановки. К особенностям вертикального шифра можно отнести следующие:

* количество столбцов в таблице фиксируется и определяется длиной ключа;
* маршрут вписывания: слева направо, сверху вниз;
* шифрограмма выписывается по столбцам в соответствии с их нумерацией (ключом).

Ключ может задаваться в виде текста (слова или словосочетания). Лексикографическое местоположение символов в ключевом выражении определяет порядок считывания столбцов.

Особенностью ***шифров множественной перестановки*** является минимум двукратная перестановка символов шифруемого сообщения. В простейшем случае это может задаваться перемешиванием не только столбцов, но и строк. Таким образом, этот случай соответствует использованию двух основных ключей: длина одного из них равна числу столбцов, другого – числу строк. К ключевой информацию мы можем относить также способы вписывания сообщения и считывания отдельных символов из текущего столбца матрицы.

**Задание 1:** Разработать авторское приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции:

* выполнять зашифрование/расшифрование текстовых документов (объемом не менее 500 знаков), созданных на основе алфавита языка в соответствии с нижеследующей таблицей вариантов задания; при этом следует использовать шифры подстановки из третьего столбца данной таблицы;

|  |  |  |
| --- | --- | --- |
| Вариант | Алфавит | Шифр |
| 2 | Русский | 1. Маршрутная перестановка (маршрут –по спирали; параметры таблицы – по указанию преподавателя)  2. Множественная перестановка, ключевые  слова – собственные имя и фамилия |

Смысл маршрутной перестановки заключается в том, что сообщение записывается в виде таблицы определенного размера и шифрование происходит по определенному маршруту. При шифровании зигзагом символы считываются по следующему маршруту:

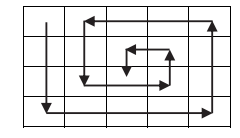


Рисунок 1 – Маршрутная перестановка по спирали

Например, для сообщения «подобед владислав георгиевич», таблица будет выглядеть следующим образом:

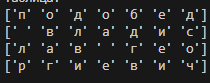


Рисунок 2 – Таблица 7 × 4 для исходного сообщения

Получаем следующий текст «подобедсочивеигрл владиег ва».

Для того чтобы расшифровать данный текст необходимо провести обратные операции – записать текст по маршруту зигзаг и прочитать и его построчно.

Ниже приведен код основных методов программы, реализующей алгоритм, показан на листинге 1:

|  |
| --- |
| def spiral\_route\_cipher(message: str, route\_step : int):      start\_time = datetime.now()      encrypted = encrypt(message, route\_step)      encrypt\_time = datetime.now() - start\_time      print('\nЗашифрованное сообщение:', encrypted)      start\_time = datetime.now()      decrypted = decrypt(encrypted, route\_step)      decrypt\_time = datetime.now() - start\_time      print('\nРасшифрованное сообщение:', decrypted)      print('\nВремя зашифрования:', encrypt\_time)      print('Время расшифрования:', decrypt\_time)      return get\_letters\_amount(encrypted) |

Листинг 1 – Реализация маршрутной перестановки по спирали

С помощью приведенного кода зашифруем и расшифруем сообщение, приведенное в примере:

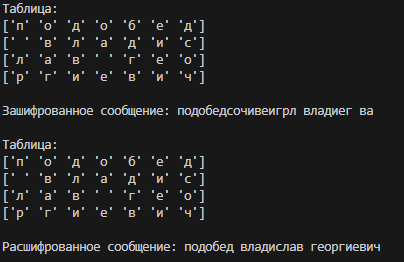


Рисунок 3 – Результат выполнения программы для маршрутной перестановки

Время шифрования в данном алгоритме вышло больше времени дешифрования.

Для **шифра множественной перестановки** были взяты два ключа. Первый – имя – указывает на количество строк, второй – фамилия – на количество столбцов в таблице.

То есть, вверху таблицы записывается первый ключ, слева вертикально записывается второй ключ. Каждому символу в каждом ключе присваивается порядковый номер по следованию букв алфавита. В

Рассмотрим данный шифр на том же примере:

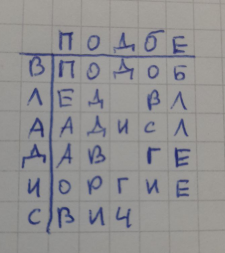


Рисунок 4 – Таблица для множественной перестановки без сортировки

Далее строки сортируются по порядку:

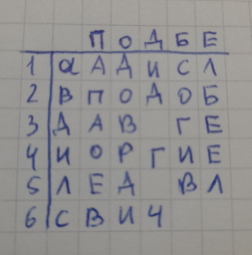


Рисунок 5 – Таблица для множественной перестановки с отсортированными строками

Затем по порядку сортируются столбцы:

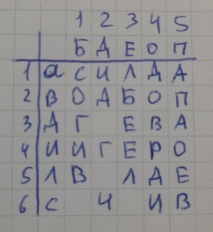


Рисунок 6 – Отсортированная таблица для множественной перестановки

Сообщение считывается по столбцам. Полученное зашифрованное сообщение: «согив ид г члбеел доврдиапаоев».

Дешифрование выполняется в обратном порядке: сообщение записывается по столбцам, восстанавливается порядок столбцов и строк, после чего сообщение считывается построчно.

Ниже приведен код основных методов программы, реализующей алгоритм, показан на листинге 2:

|  |
| --- |
| # формируем матрицу  column\_list = sorted(keyword\_column)  row\_list = sorted(' ' + keyword\_row)  row\_list = np.reshape(row\_list, (len(keyword\_row) + 1, 1))  matrix\_message = np.empty((len(keyword\_row), len(keyword\_column)), str)  for i in range(len(keyword\_column)):  for j in range(len(keyword\_row)):  letter = message[0]  message = message[1:]  matrix\_message[j][i] = letter  matrix = np.vstack([column\_list, matrix\_message])  matrix = np.hstack([row\_list, matrix])  print('\nТаблица с зашифрованным сообщением')  for row in matrix:  print(row)  # сортируем матрицу по ключам  order\_rows = []  for character in sorted(' ' + keyword\_row):  order\_rows.append(list(' ' + keyword\_row).index(character))  matrix = matrix[np.argsort(order\_rows)]  print('\nТаблица с восстановленными строками')  for row in matrix:  print(row)  matrix = np.transpose(matrix)  order\_columns = []  for character in sorted(' ' + keyword\_column):  order\_columns.append(list(' ' + keyword\_column).index(character))  matrix = matrix[np.argsort(order\_columns)]  matrix = np.transpose(matrix)  print('\nТаблица с восстановленными столбцами')  for row in matrix:  print(row)  # записывам соощение по строкам  matrix = matrix[1:, 1:]  res = ''.join(matrix.flatten())  return res  def get\_letters\_amount(seq):  letters\_dictionary = {}  for i in seq:  if i.isalpha():  if i not in letters\_dictionary:  letters\_dictionary[i] = 0  letters\_dictionary[i] += 1  return dict(sorted(letters\_dictionary.items())) |

Листинг 2 – Реализация множественных перестановок

С помощью приведенного кода зашифруем и расшифруем сообщение, приведенное в примере показано на рисунке 7:

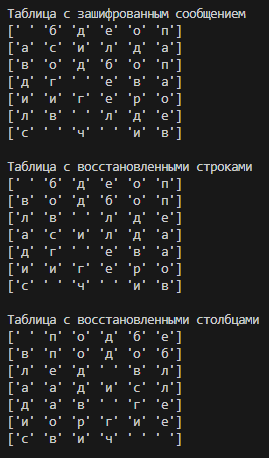


Рисунок 7 – зашифруем и расшифруем сообщение



Рисунок 8 – Результат выполнения программы для множественной перестановки

Время зашифрования превысило время дешифрования. Отсюда можно сделать вывод о невысокой криптостойкости шифров перестановки. Это также подтверждает частотный анализ. Частота появления символов не изменилась после зашифрования.

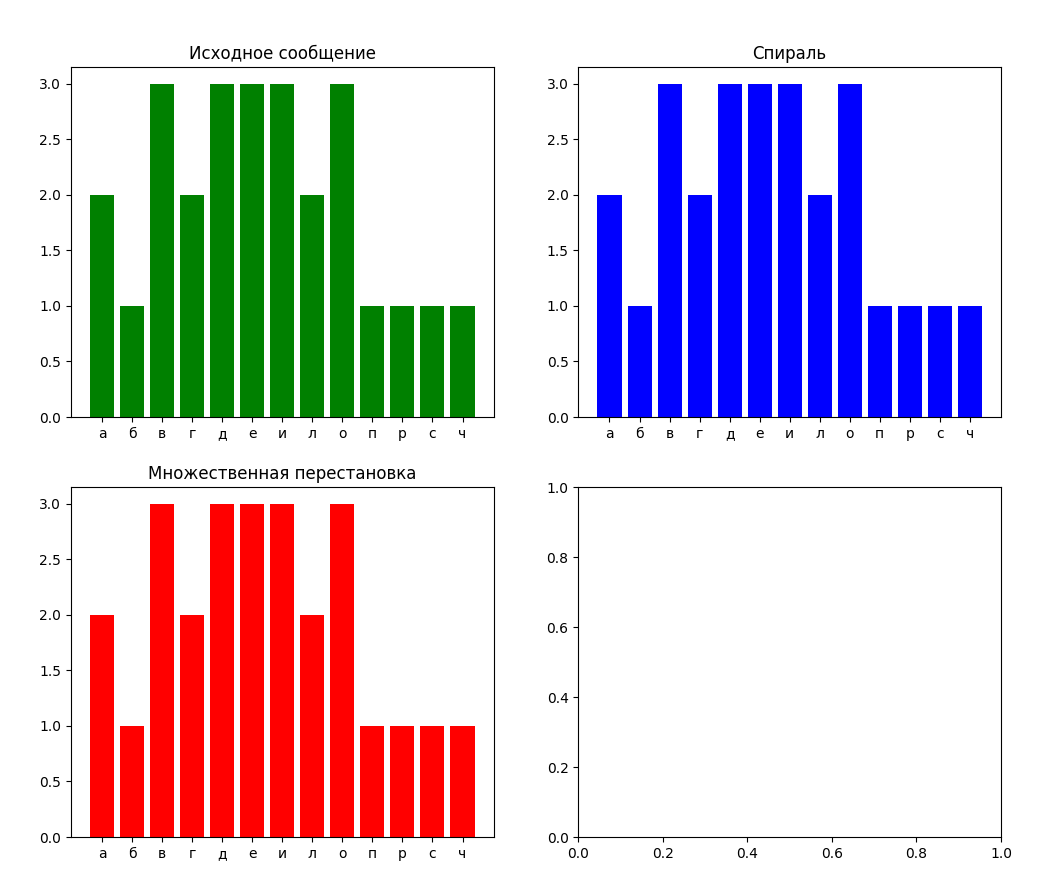


Рисунок 9 – Гистограммы частоты появления символов

**Вывод:** в ходе лабораторной работы было разработано приложение, позволяющее шифровать и расшифровывать сообщения с помощью шифров множественной перестановки и маршрутной перестановки (маршрут зигзаг), были построены гистограммы частоты встречаемости символов и измерено время шифрования/дешифрования. Временные затраты незначительны.