Федеральное государственное автономное

образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Прикладная математика и компьютерная безопасность

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 5**

по Криптографические методы защиты информации

наименование дисциплины

Шифр Кардано

тема

Преподаватель В.И. Вайнштейн

подпись**,** дата инициалы, фамилия

Студент КИ17-01, 031722011 К.А. Василенко

номер группы, зачетной книжки подпись**,** дата инициалы, фамилия

Красноярск 2021

# ВВЕДЕНИЕ

Цель работы: реализовать шифр Кардано (шифрование и расшифрование) на любом языке программирования. Предусмотреть графический интерфейс.

Задачи:

1. реализовать программно шифр Кардано, предусматривая графический интерфейс;
2. провести тесты на работоспособность программы;
3. сделать отчёт о проделанной работе.

**Реализация шифра**

1. **Описание шифра**

Шифрование с добавлением «мусора»

Решётка — квадрат N х N клеток, некоторые из которых вырезаны (в том числе, возможно, и имеющие общую сторону или вершину). Клетки должны иметь такой размер, чтобы в каждую помещалась ровно одна буква. Вырезанные клетки должны располагаться таким образом, чтобы никакие две из них не оказывались в одном и том же месте при поворотах решётки.

Чтобы зашифровать сообщение, нужно разместить решётку на бумаге и вписать часть текста в вырезанные клетки, затем повернуть решётку на 90° и вписать следующую часть и т. д. После этого в оставшиеся пустыми места на бумаге нужно вписать произвольные символы (разумеется, для лучшей маскировки нужно использовать символы из того же алфавита, что и символы, из которых состоит сообщение).

Шифрование без добавления «мусора»

Этот способ шифрования отличается от предыдущего тем, что клетки в решётке вырезаются таким образом, чтобы при её поворотах можно было записать по букве в каждое место на бумаге. Опять же, клетки должны быть вырезаны так, чтобы при поворотах не было таких пар, которые оказываются на одном и том же месте. Очевидно, такая решётка может быть с чётным N — числом клеток в одной строке или столбце. Для N=2 такая решётка единственная. Но при увеличении N количество таких решёток очень быстро растёт: С = , где C — количество решёток со стороной N. Уже для N = 8 можно сделать 416 — более 4 миллиардов различных решёток, поэтому на практике определить, какой именно решёткой пользовались при шифровании, невозможно. Однако этот способ является простым перестановочным шифром, который поддается взлому (например, с помощью частотного анализа).

Для облегчения создания решёток Кардано, удовлетворяющих всем написанным выше требованиям, можно воспользоваться конструктором решёток. Например, для решётки 8 х 8 (см. рисунок) и шифрования «без мусора» нужно вырезать строго одну из четырёх клеток с номером 1, строго одну из четырёх клеток с номером 2 и т. д. Для шифрования «с мусором» некоторые из номеров можно не вырезать.

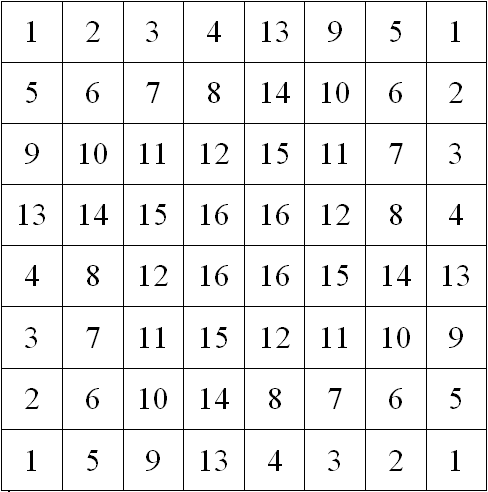
{\displaystyle c\_{j}=(m\_{j}+k\_{j})\mod {n}}{\displaystyle m\_{j}=(c\_{j}+n-k\_{j})\mod {n}}

Рисунок 1 – Решётка Кардано.

1. **Программная реализация**

Программа написана на языке Python с использованием библиотеки PyQt5 и Qt designer для отрисовки графического интерфейса.

1. **Листинг с описанием основного алгоритма программы**

**cardano.py**

**from** numpy **import** rot90, hstack, vstack, array, add, append, dtype, where  
**from** random **import** randint, choice  
**import** string  
**from** PyQt5 **import** QtWidgets  
  
**def** codingCardano(text, size):  
 **if not** text:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle(**"Ошибка"**)  
 msgBox.setText(**"Введите текст!"**)  
 msgBox.exec\_()  
 **return** (**""**)  
  
 *#4 матрицы размера size на size, для мусора* **global** res  
 **while** len(text)%(4\*size\*size) != 0:  
 *#text += "\*"* text += choice(string.ascii\_letters)  
  
 arr1 = []  
 k = 1  
 **for** i **in** range(size):  
 mtx = []  
 **for** j **in** range(size):  
 mtx.append(str(k))  
 k+=1  
 arr1.append(mtx)  
  
 arr3 = rot90(arr1)  
 arr4 = rot90(arr3)  
 arr2 = rot90(arr4)  
  
 arr1 = array(arr1, dtype=**"U25"**)  
 arr3 = array(arr3, dtype=**"U25"**)  
 arr2 = array(arr2, dtype=**"U25"**)  
 arr4 = array(arr4, dtype=**"U25"**)  
  
 *#tmp=[2, 4, 4, 1, 3, 4, 2, 1, 2]* **for** k **in** range(1, size\*size + 1):  
 *#numberMtx=tmp[k-1]* numberMtx = randint(1, 4)  
 **if** numberMtx == 1:  
 **for** i **in** range(len(arr1)):  
 **for** j **in** range(len(arr1)):  
 **if** arr1[i][j] == str(k):  
 arr1[i][j] += **'X'  
 if** numberMtx == 2:  
 **for** i **in** range(len(arr2)):  
 **for** j **in** range(len(arr2)):  
 **if** arr2[i][j] == str(k):  
 arr2[i][j] += **'X'  
 if** numberMtx == 3:  
 **for** i **in** range(len(arr3)):  
 **for** j **in** range(len(arr3)):  
 **if** arr3[i][j] == str(k):  
 arr3[i][j] += **'X'  
 if** numberMtx == 4:  
 **for** i **in** range(len(arr4)):  
 **for** j **in** range(len(arr4)):  
 **if** arr4[i][j] == str(k):  
 arr4[i][j] += **'X'** arr11 = hstack((arr1, arr2)) *# вправо* arr12 = hstack((arr3, arr4)) *# вправо* arr = vstack((arr11, arr12)) *# вниз* print(arr)  
  
 restext=**""  
 for** r **in** range(len(text)//(size\*size\*4)):  
 res = []  
 **for** i **in** range(size \* 2):  
 res.append([**""**] \* size \* 2)  
  
 *#поворачиваем 4 раза и кодируем, символов = size\*size по 4 раза* **for** p **in** range(4):  
 **for** k **in** range(1, size\*size + 1):  
 i, j = where(arr == str(k) + **"X"**)  
 res[int(i)][int(j)] = text[k-1]  
 arr = rot90(arr)  
 text = text[size\*size:]  
  
 **for** i **in** range(len(res)):  
 **for** k **in** range(len(res[i])):  
 restext+=res[i][k]  
 print(restext)  
 **return** arr, restext  
  
**def** decodingCardano(arr, text):  
 size = len(arr)//2  
  
 res = **""  
 for** r **in** range(len(text)//(size\*size\*4)):  
 mtx = []  
 **for** i **in** range(size\*2):  
 mtx1 = []  
 **for** j **in** range(size\*2):  
 mtx1.append(text[0])  
 text=text[1:]  
 mtx.append(mtx1)  
 **for** i **in** mtx:  
 print(i)  
  
 **for** p **in** range(4):  
 **for** k **in** range(1, size \* size + 1):  
 i, j = where(arr == str(k) + **"X"**)  
 res += mtx[int(i)][int(j)]  
 arr = rot90(arr)  
 print(res)  
 **return** res  
  
*#arr, res = codingCardano("Привет, мир!", 4)  
#print(res)  
#res1 = decodingCardano(arr, res)  
#print(res1)*

1. **Примеры работы программы**

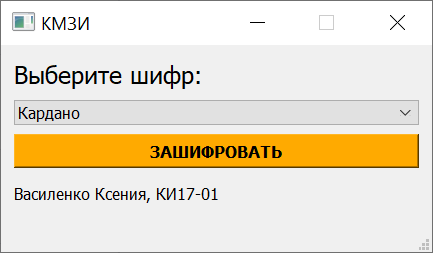


Рисунок 2 – Главное окно.

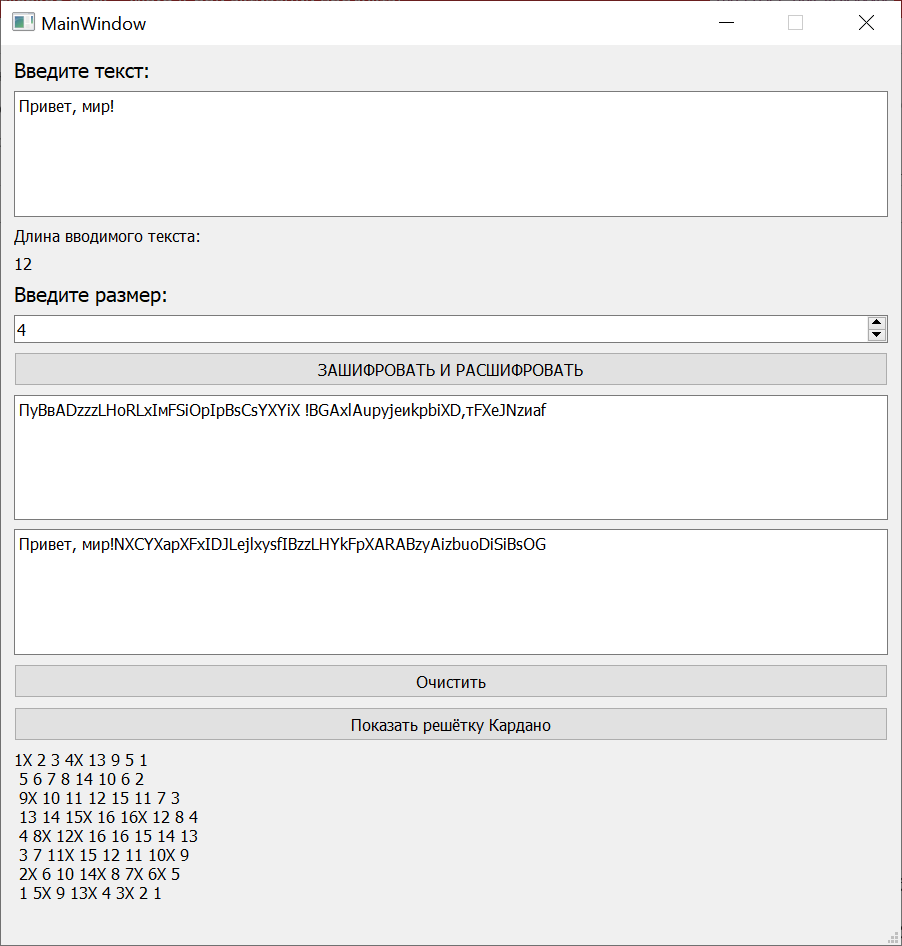


Рисунок 3 – Шифр Кардано.

# ЗАКЛЮЧЕНИЕ

Во время выполнения лабораторной работы №5 мною был изучен и реализован программно шифр Кардано, а также предусмотрен графический интерфейс с помощью PyQt5 и Qt designer.