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 15**

по Криптографические методы защиты информации

наименование дисциплины

Метод Гаммирования

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 метод Гаммирования (шифрование и расшифрование) на любом языке программирования. Предусмотреть графический интерфейс.

Задачи:

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

**Реализация шифра**

1. **Описание шифра**

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле.

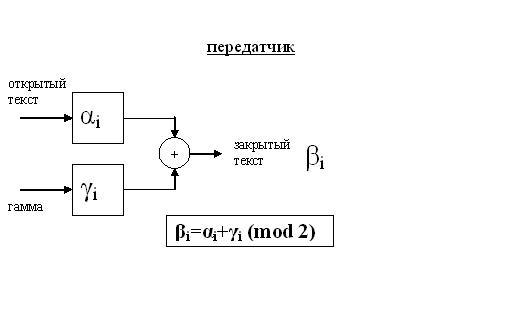


Рисунок 1 – Схема передатчика.

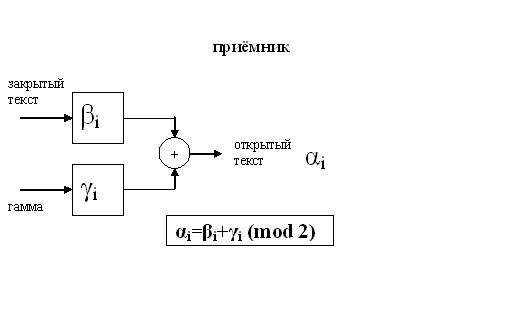


Рисунок 2 – Схема приёмника.

Линейный конгруэнтный метод — один из методов генерации псевдослучайных чисел.

Суть метода заключается в вычислении последовательности случайных чисел Xn, полагая

*Xn+1 = (aXn + c) mod m*,

m - модуль (натуральное число, относительно которого вычисляет остаток от деления; *m >= 2)*, *a* – множитель (0 <= a <= m), c – приращение (*0 <= c < m*), *X0* - начальное значение (*0 <= X0 < m*).

Эта последовательность называется линейной конгруэнтной последовательностью. Например, для m = 10, X0 = a = c = 7 получим последовательность 7, 6, 9, 0, 7, 6, 9, 0, …

Линейная конгруэнтная последовательность, определенная числами m, a, c и X0 периодична с периодом, не превышающим m. При этом длина периода равна m тогда и только тогда, когда:

1. Числа c и m взаимно простые;
2. b=a-1 кратно p для каждого простого p, являющегося делителем m;
3. b кратно 4, если m кратно 4.

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

**gamm.py**

**from** PyQt5 **import** QtWidgets  
  
**def** binar(n):  
 b = ""  
 **while** n > 0:  
 b = str(n % 2) + b  
 n = n // 2  
 **return** b  
  
**def** ten(n):  
 b = ""  
 **while** n > 0:  
 b = str(n % 10) + b  
 n = n // 10  
 **return** b  
  
**def** xor(a, b):  
 res = ""  
 **for** i **in** range(len(a)):  
 **if** a[i] == '1':  
 **if** b[i] == '1':  
 res+='0'  
 **else**:  
 res+='1'  
 **if** a[i] == '0':  
 **if** b[i] == '1':  
 res+='1'  
 **else**:  
 res+='0'  
 **return** res  
  
**def** genGamma(a, b, m, X0, text):  
 **if not** a **or not** b **or not** m **or not** X0 **or not** text:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Заполните все поля!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **try**:  
 a = int(a)  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **try**:  
 b = int(b)  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **try**:  
 m = int(m)  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **try**:  
 X0 = int(X0)  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **if** int(m) < 2:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("m >= 2!")  
 msgBox.exec\_()  
 **return** ("")  
  
 #X0 = 7  
 res = [X0]  
 **while** len(res) != len(text):  
 res.append((a\*X0 + b) % m)  
 X0 = (a\*X0 + b) % m  
 res = str(res)  
 res = res.replace(",", "")  
 res = res.replace("[", "")  
 res = res.replace("]", "")  
 #print(res)  
 **return** res  
  
  
**def** codingGamma(gamma, text):  
 # arrGamm = list(genGamma(a, b, m, X0, text).split(' '))  
 **if not** text **or not** gamma:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Заполните поле ключа и текста!")  
 msgBox.exec\_()  
 **return** ("")  
  
 arrGamm = list(gamma.split(' '))  
 **for** i **in** range(len(arrGamm)):  
 **try**:  
 X0 = int(arrGamm[i])  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **if** len(arrGamm) != len(text):  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Длина ключа должна быть равна длине текста!")  
 msgBox.exec\_()  
 **return** ("")  
  
 gammCodeBin = []  
 **for** i **in** range(len(arrGamm)):  
 gammCodeBin.append((bin(int(arrGamm[i]))[2:]).zfill(25))  
 #gammCodeBin.append((binar(int(arrGamm[i]))).zfill(25))  
 #print('g', gammCodeBin)  
  
 textCode = []  
 **for** i **in** text:  
 textCode.append(ord(i))  
  
 textCodeBin = []  
 **for** i **in** range(len(text)):  
 textCodeBin.append((bin(int(textCode[i]))[2:]).zfill(25))  
 #textCodeBin.append((binar(int(ord(text[i])))).zfill(25))  
 #print('t', textCodeBin)  
  
 resCode = ""  
 **for** i **in** range(len(textCodeBin)):  
 tmp1 = xor(textCodeBin[i], gammCodeBin[i])  
 resCode+=chr(int(tmp1, 2))  
 #print(resCode)  
  
 **return** resCode  
  
**def** codingGamma1(gamma, text):  
 # arrGamm = list(genGamma(a, b, m, X0, text).split(' '))  
 **if not** text **or not** gamma:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Заполните поле ключа и текста!")  
 msgBox.exec\_()  
 **return** ("")  
  
 arrGamm = list(gamma.split(' '))  
 **for** i **in** range(len(arrGamm)):  
 **try**:  
 X0 = int(arrGamm[i])  
 **except** Exception:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Введите число!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **if** len(arrGamm) != len(text):  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка")  
 msgBox.setText("Длина ключа должна быть равна длине текста!")  
 msgBox.exec\_()  
 **return** ("")  
  
 gammCodeBin = []  
 **for** i **in** range(len(arrGamm)):  
 gammCodeBin.append((bin(int(arrGamm[i]))[2:]).zfill(25))  
  
 textCodeBin = []  
 **for** i **in** range(len(text)):  
 textCodeBin.append((bin(text[i])[2:]).zfill(25))  
  
 resCode = bytearray(b"")  
 **for** i **in** range(len(textCodeBin)):  
 tmp1 = xor(textCodeBin[i], gammCodeBin[i])  
 resCode.append(int(tmp1, 2) % 256)  
  
 **return** resCode

1. **Примеры работы программы**

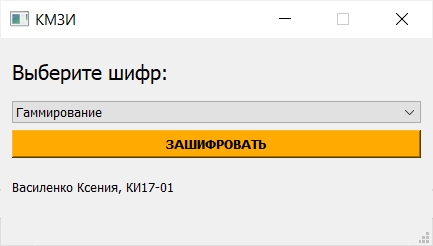


Рисунок 3 – Главное окно.

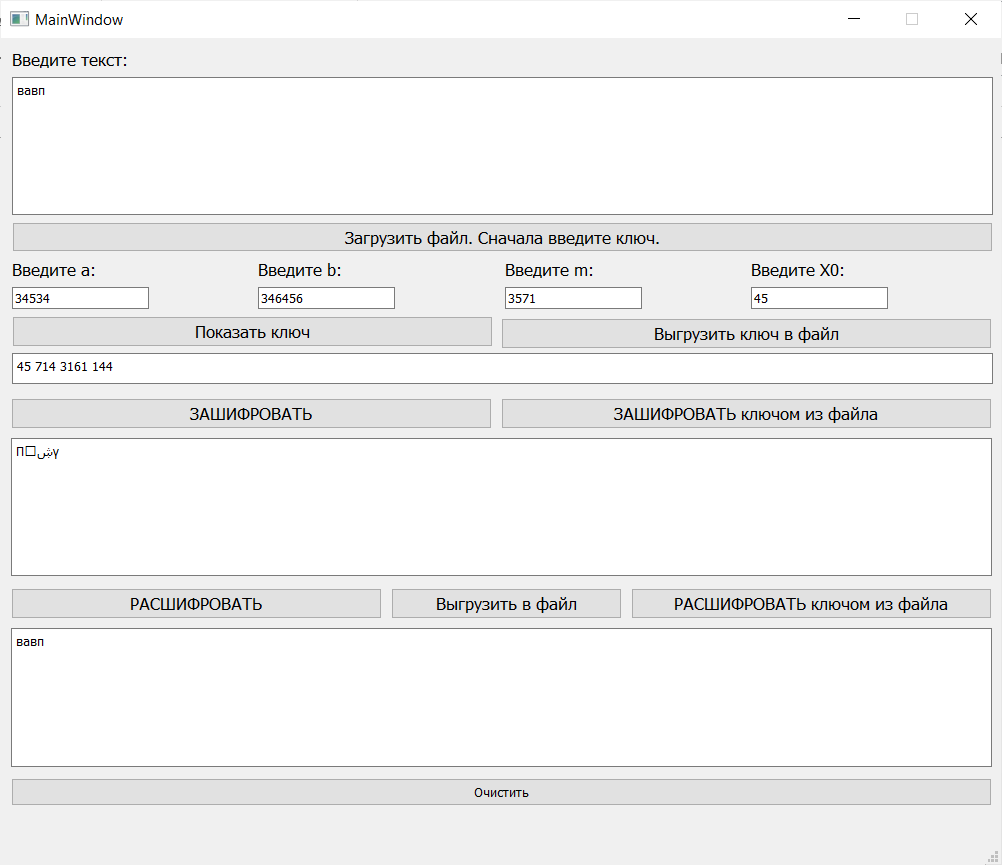


Рисунок 4 – Метод Гаммирования.

# ЗАКЛЮЧЕНИЕ

Во время выполнения лабораторной работы №15 мною был изучен и реализован программно метод Гаммирования, а также предусмотрен графический интерфейс с помощью PyQt5 и Qt designer.