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 12**

по Криптографические методы защиты информации

наименование дисциплины

Шифр Вернама

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

Задачи:

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

**Реализация шифра**

1. **Описание шифра**

Шифр Вернама (англ. Vernam cipher) — система симметричного шифрования, изобретённая в 1917 году Гилбертом Вернамом.

Криптосистема была предложена для шифрования телеграфных сообщений, которые представляли собой бинарные тексты, в которых открытый текст представляется в коде Бодо (в виде пятизначных «импульсных комбинаций»). В этом коде, например, буква «А» имела вид (1 1 0 0 0). На бумажной ленте цифре «1» соответствовало отверстие, а цифре «0» — его отсутствие. Секретный ключ должен был представлять собой хаотичный набор букв того же самого алфавита.

Для получения шифротекста открытый текст объединяется операцией «исключающее или» с секретным ключом. Так, например, при применении ключа (1 1 1 0 1) на букву «А» (1 1 0 0 0) получаем зашифрованное сообщение (0 0 1 0 1): {\displaystyle (11000)\oplus (11101)=(00101)}(1 1 0 0 0) xor (1 1 1 0 1) = (0 0 1 0 1) Зная, что для принимаемого сообщения имеем ключ (1 1 1 0 1), легко получить исходное сообщение той же операцией: {\displaystyle (00101)\oplus (11101)=(11000)}(0 0 1 0 1) xor (1 1 1 0 1) = (1 1 0 0 0). Для абсолютной криптографической стойкости ключ должен обладать тремя критически важными свойствами:

1. Иметь случайное равномерное распределение: {\displaystyle P\_{k}(k)=1/2^{N}}*Pk(k) = 1/2N*, где *k* — ключ, а *N* — количество бинарных символов в ключе;
2. Совпадать по размеру с заданным открытым текстом;
3. Применяться только один раз.

{\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**

**vernam.py**

**from** PyQt5 **import** QtWidgets  
  
**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** codingVernam(text, key, flag):  
 **if not** text **or not** key:  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка!")  
 msgBox.setText("Введите текст!")  
 msgBox.exec\_()  
 **return** ("")  
  
 **if** len(text) != len(key):  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка!")  
 msgBox.setText("Длина ключа должна быть равна длине текста!")  
 msgBox.exec\_()  
 msgBox = QtWidgets.QMessageBox()  
 msgBox.setWindowTitle("Ошибка!")  
 msgBox.setText("Длина текста = "+str(len(text)))  
 msgBox.exec\_()  
 **return** ("")  
  
 textCode = []  
 **for** i **in** text:  
 textCode.append(ord(i))  
 print('t', textCode)  
  
 **for** i **in** range(len(textCode)):  
 textCode[i] = (bin(textCode[i])[2:]).zfill(25)  
 print(textCode)  
  
 keyCode = []  
 **for** i **in** key:  
 keyCode.append(ord(i))  
  
 **for** i **in** range(len(keyCode)):  
 keyCode[i] = (bin(keyCode[i])[2:]).zfill(25)  
  
 resText = ""  
 **for** i **in** range(len(textCode)):  
 tmp1 = xor(textCode[i], keyCode[i])  
 #resText += str(chr(int(tmp1)))  
 resText += str(chr(int(tmp1, 2)))  
  
 **return** resText

1. **Примеры работы программы**

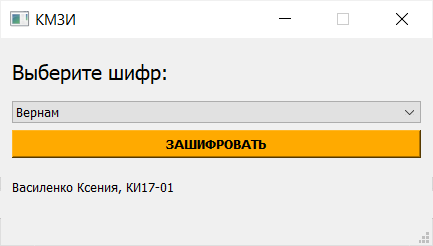


Рисунок 1 – Главное окно.

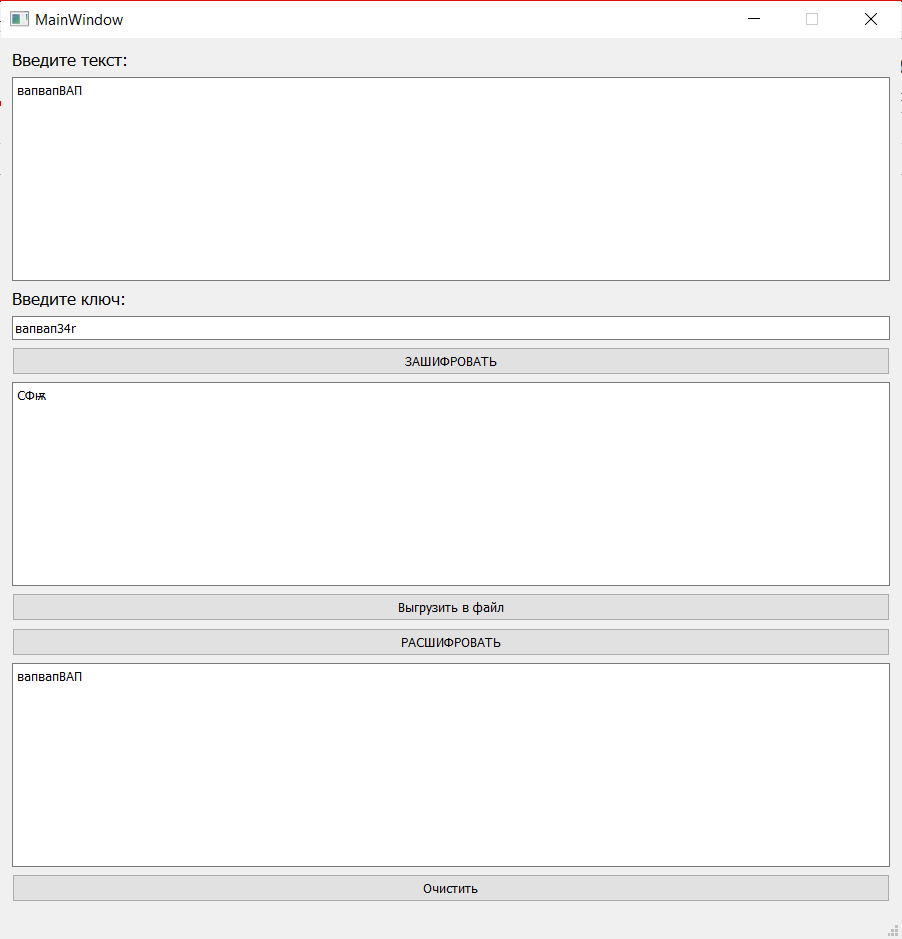


Рисунок 2 – Шифр Вернама.

# ЗАКЛЮЧЕНИЕ

Во время выполнения лабораторной работы №12 мною был изучен и реализован программно шифр Вернама, а также предусмотрен графический интерфейс с помощью PyQt5 и Qt designer.