**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Новгородский государственный университет имени Ярослава Мудрого»**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра ИТиС

Лабораторная работа №4

По дисциплине: Защита информации

Отчет

Выполнил: студент гр. 9091

\_\_\_\_\_\_\_\_\_\_\_Чалый.С.М

Проверил: Преподаватель

\_\_\_\_\_\_\_\_\_\_\_Жгун.Т.В

Великий Новгород

2022

1. **Формулировка цели и задач**

Реализовать режимы шифрования DES для имеющейся процедуры шифрования.

Есть дети, есть отцы, деды – а собственно мужчины героинь будто бы запропали куда-то.

Видимо, это миф, когда говорят, что мужчины вполне могут обойтись без женщин, зато вот женщины без сильного пола не могут совсем, никак.

Могут.

По крайней мере, с недавнего времени.

Мир, описанный в этой книге, движим женщиной. Женщины здесь – тихие подвижники не быта (или не только быта), а бытия.Не сказать, что мужчины вовсе ни при чем: их, в общем-то, еще ждут, по ним даже скучают – но они все равно где-то на периферии женского зрения и сознания.

На мужчин будто бы махнули рукой: что взять с них – нечего.Мужчина никакой не центр женского мироздания – как зачастую кажется самим мужчинам. В лучшем случае это нелишний атрибут. Только непонятно чего. Никак не счастья.

Читая эту книжку, очень хотелось понять, как эпоха отражается в женских зрачках, что за вкус у нее, что за цвет. Какие детали эпохи наглядны, каких нет вовсе.

И в первую очередь выяснилось, говорю, что тут нет мужчины-героя. Во всех смыслах слова «герой».Нету его.Есть Аркаша в повести «Вариант нормы» Анны Андроновой – но он скорей сын своей матери, зато муж никакой, да и будущая жена тоже воспринимает его как большого непричесанного ребенка.

Муж время от времени бросает реплики в другой повести Андроновой – «Я не зайчик», но он во всей этой круговерти событий все равно эпизодическое лицо, он почти случаен в семье: за все время повествования его с трудом хватает на то, чтоб приделать уши маске зайца. Дальше он опять засыпает и спит. Все время спит.

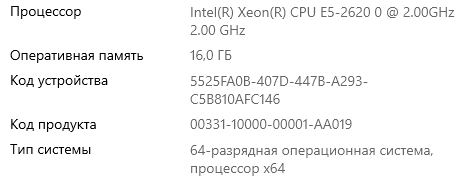
Гриша в «Кукуше» Майи Кучерской – тоже почти сын героини; в любом случае не муж точно. Муж, правда, у нее тоже есть, но он в буквальном смысле убегает. Не смог перенести ожидания – жена застряла в лифте, и пришлось полчаса ее ждать. Разве это возможно? Сбежал.

В качестве ключа шифрования выступает пароль «Пустой колос голову кверху носит» При этом раундовые ключи являются последовательными отрезками ключа шифрования.

Визуализировать результаты тестами

1. **Текст программы**

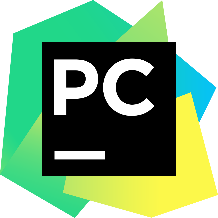
**2.1 Тестирование и запуск.**



Необходимые средства применяемые мною при выполнении данной работы:

PyCharm

* Version: 2021.3.2
* Build: 213.6777.50
* 31 January 2022
* Community
* Python 3.10



"**2.1 Тестирование и запуск**".

-\*- coding: utf8 -\*-  
import module  
import matplotlib.pyplot as plt  
  
text = "Есть дети, есть отцы, деды а собственно мужчины героинь будто бы запропали куда то.Видимо, это миф, когда говорят, что мужчины вполне могут обойтись без женщин, зато вот женщины без сильного пола не могут совсем, никак.Могут.По крайней мере, с недавнего времени.Мир, описанный в этой книге, движим женщиной. Женщины здесь тихие подвижники не быта или не только быта, а бытия.Не сказать, что мужчины вовсе ни при чем: их, в общем то, еще ждут, по ним даже скучают но они все равно где то на периферии женского зрения и сознания.На мужчин будто бы махнули рукой: что взять с них нечего.Мужчина никакой не центр женского мироздания как зачастую кажется самим мужчинам. В лучшем случае это нелишний атрибут. Только непонятно чего. Никак не счастья.Читая эту книжку, очень хотелось понять, как эпоха отражается в женских зрачках, что за вкус у нее, что за цвет. Какие детали эпохи наглядны, каких нет вовсе.И в первую очередь выяснилось, говорю, что тут нет мужчины героя. Во всех смыслах слова герой.Нету его.Есть Аркаша в повести Вариант нормы Анны Андроновой но он скорей сын своей матери, зато муж никакой, да и будущая жена тоже воспринимает его как большого непричесанного ребенка.Муж время от времени бросает реплики в другой повести Андроновой Я не зайчик, но он во всей этой круговерти событий все равно эпизодическое лицо, он почти случаен в семье: за все время повествования его с трудом хватает на то, чтоб приделать уши маске зайца. Дальше он опять засыпает и спит. Все время спит.Гриша в Кукуше Майи Кучерской тоже почти сын героини в любом случае не муж точно. Муж, правда, у нее тоже есть, но он в буквальном смысле убегает. Не смог перенести ожидания жена застряла в лифте, и пришлось полчаса ее ждать. Разве это возможно? Сбежал"  
key = "Пустой колос голову кверху носит"  
  
key = key.replace(" ", "")  
key = key.lower()  
key = key + "."  
text = text.replace(":", "")  
text = text.replace(".", "")  
text = text.replace(",", "")  
text = text.replace("(", "")  
text = text.replace(")", "")  
text = text.replace("?", "")  
text = text.replace(";", "")  
text = text.lower()  
text = text + "."  
  
  
def ECB(binArr, keyArr, point=18):  
 timeLine = ""  
 keyLine = ""  
 keyInd = 0  
 encryptedArr = []  
 for i in range(0, len(binArr)):  
 timeLine += binArr[i]  
 keyLine += keyArr[keyInd]  
 if len(timeLine) == point:  
 encryptedArr.append(module.getXorResult(timeLine, keyLine))  
 timeLine = ""  
 keyLine = ""  
 keyInd += 1  
 if keyInd >= len(keyArr):  
 keyInd = 0  
 return encryptedArr  
  
  
def CBC(binArr, keyArr, point=18):  
 timeLine = ""  
 keyLine = ""  
 keyInd = 0  
 arrInd = 0  
 encryptedArr = []  
 for i in range(0, len(binArr)):  
 timeLine += binArr[i]  
 keyLine += keyArr[keyInd]  
 if len(timeLine) == point:  
 if len(encryptedArr) != 0:  
 res = module.getXorResult(encryptedArr[arrInd], timeLine)  
 encryptedArr.append(module.getXorResult(res, keyLine))  
 arrInd += 1  
 else:  
 encryptedArr.append(module.getXorResult(timeLine, keyLine))  
  
 timeLine = ""  
 keyLine = ""  
 keyInd += 1  
 if keyInd >= len(keyArr):  
 keyInd = 0  
 return encryptedArr  
  
  
def CFB(binArr, keyArr, point=18):  
 timeLine = ""  
 keyLine = ""  
 keyInd = 0  
 arrInd = 0  
 encryptedArr = []  
 for i in range(0, len(binArr)):  
 timeLine += binArr[i]  
 keyLine += keyArr[keyInd]  
 if len(timeLine) == point:  
 if len(encryptedArr) != 0:  
 res = module.getXorResult(keyLine, timeLine)  
 encryptedArr.append(module.getXorResult(res, encryptedArr[arrInd]))  
 arrInd += 1  
 else:  
 encryptedArr.append(module.getXorResult(timeLine, keyLine))  
  
 timeLine = ""  
 keyLine = ""  
 keyInd += 1  
 if keyInd >= len(keyArr):  
 keyInd = 0  
 return encryptedArr  
  
  
def OFB(binArr, keyArr, point=18):  
 timeLine = ""  
 keyLine = ""  
 keyInd = 0  
 encryptedArr = []  
 Keys = []  
 keysInd = 0  
 for i in range(0, len(binArr)):  
 timeLine += binArr[i]  
 keyLine += keyArr[keyInd]  
 if len(timeLine) == point:  
 if len(encryptedArr) != 0:  
 res = module.getXorResult(keyLine, Keys[keysInd])  
 Keys.append(res)  
 encryptedArr.append(module.getXorResult(res, timeLine))  
 keysInd += 1  
 else:  
 Keys.append(keyLine)  
 encryptedArr.append(module.getXorResult(timeLine, keyLine))  
  
 timeLine = ""  
 keyLine = ""  
 keyInd += 1  
 if keyInd >= len(keyArr):  
 keyInd = 0  
 return encryptedArr  
  
  
def translateTextIntoBin(text):  
 needLine = ""  
 for ch in text:  
 if ch != " " and ch != ".":  
 needLine += module.getSymbol(ch)  
 return needLine  
  
  
def translateBinIntoText(binArr):  
 needString = ""  
 for value in binArr:  
 timeLine = ""  
 for ch in value:  
 timeLine += ch  
 if len(timeLine) == 6:  
 needString += module.getAlp(timeLine)  
 timeLine = ""  
 return needString  
  
  
Name = []  
Values = []  
  
  
def makeGraph(str, title):  
 Name.clear()  
 Values.clear()  
 fig, ax = plt.subplots()  
 for i in str:  
 counter = 0  
 for g in str:  
 if i == g:  
 counter += 1  
 Name.append(i)  
 Values.append(counter)  
  
 plt.bar(Name, Values)  
 plt.title(title)  
 plt.show()  
  
  
def makePoints(str, title):  
 A = []  
 B = []  
 counter = 0  
 for i in str:  
 if counter == 0:  
 A.append(module.getOctFromBinValue(module.getSymbol(i)))  
 elif counter == len(str) - 1:  
 B.append(module.getOctFromBinValue(module.getSymbol(i)))  
 else:  
 B.append(module.getOctFromBinValue(module.getSymbol(i)))  
 A.append(module.getOctFromBinValue(module.getSymbol(i)))  
 counter += 1  
 plt.title(title)  
 plt.scatter(A, B, label='high income low saving', color='b')  
 plt.legend()  
 plt.show()  
  
  
def graphAnalysis(str, title):  
 Mass = []  
 Zeros = 0  
 Ones = 0  
 for i in str:  
 value = module.getSymbol(i)  
 Mass.append(value)  
 for i in Mass:  
 for g in i:  
 if g == "0":  
 Zeros += 1  
 elif g == "1":  
 Ones += 1  
 plt.title(title)  
 plt.bar(1, Zeros)  
 plt.bar(2, Ones)  
 plt.show()  
  
  
def monno(str, title):  
 max = 0  
 counterMax = 0  
  
 min = 0  
 counterMin = 0  
  
 countOfMon = {}  
  
 for i in str:  
 res = module.getSymbol(i)  
 value = module.getOctFromBinValue(res)  
 if value < min:  
 min = value  
 if value > max:  
 max = value  
 for i in str:  
 g = module.getOctFromBinValue(module.getSymbol(i))  
 if g > min and g < max:  
 if g in countOfMon:  
 value = countOfMon[g]  
 value += 1  
 countOfMon[g] = value  
 else:  
 countOfMon[g] = 1  
 A = []  
 B = []  
 for key in countOfMon:  
 A.append(key)  
 B.append(countOfMon[key])  
 plt.title(title)  
 plt.bar(A, B)  
 plt.show()  
  
  
text2 = text.replace(" ", "")  
text2 = text2.replace(".", "")  
makeGraph(text2, "Исходный текст")  
makePoints(text2, "Исходный текст")  
graphAnalysis(text2, "Исходный текст")  
# monno(text2, "Исходный текст")  
  
# ECB  
textToBin = translateTextIntoBin(text)  
keyToBin = translateTextIntoBin(key)  
binArr = ECB(textToBin, keyToBin)  
encryptedText = translateBinIntoText(binArr)  
makeGraph(encryptedText, "ECB")  
makePoints(encryptedText, "ECB")  
graphAnalysis(encryptedText, "ECB")  
monno(encryptedText, "ECB")  
  
# CBC  
binArr = CBC(textToBin, keyToBin)  
encryptedText = translateBinIntoText(binArr)  
makeGraph(encryptedText, "CBC")  
makePoints(encryptedText, "CBC")  
graphAnalysis(encryptedText, "CBC")  
monno(encryptedText, "CBC")  
  
# CFB  
binArr = CFB(textToBin, keyToBin)  
encryptedText = translateBinIntoText(binArr)  
makeGraph(encryptedText, "CFB")  
makePoints(encryptedText, "CFB")  
graphAnalysis(encryptedText, "CFB")  
monno(encryptedText, "CFB")  
  
# OFB  
binArr = OFB(textToBin, keyToBin)  
encryptedText = translateBinIntoText(binArr)  
makeGraph(encryptedText, "OFB")  
makePoints(encryptedText, "OFB")  
graphAnalysis(encryptedText, "OFB")  
monno(encryptedText, "OFB")

Symbols = {}  
#################################A\_L\_P###############################  
Symbols["а"] = "000001" #a  
Symbols["б"] = "000010" #б  
Symbols["в"] = "000011" #в  
Symbols["г"] = "000100" #г  
Symbols["д"] = "000101" #д  
Symbols["е"] = "000110" #е  
Symbols["ё"] = "000111" #ё  
Symbols["ж"] = "001000" #ж  
Symbols["з"] = "001001" #з  
Symbols["и"] = "001010" #и  
Symbols["й"] = "001011" #й  
Symbols["к"] = "001100" #к  
Symbols["л"] = "001101" #л  
Symbols["м"] = "001110" #м  
Symbols["н"] = "001111" #н  
Symbols["о"] = "010000" #о  
Symbols["п"] = "010001" #п  
Symbols["р"] = "010010" #р  
Symbols["с"] = "010011" #с  
Symbols["т"] = "010100" #т  
Symbols["у"] = "010101" #у  
Symbols["ф"] = "010110" #ф  
Symbols["х"] = "010111" #х  
Symbols["ц"] = "011000" #ц  
Symbols["ч"] = "011001" #ч  
Symbols["ш"] = "011010" #ш  
Symbols["щ"] = "011011" #щ  
Symbols["ъ"] = "011100" #ъ  
Symbols["ы"] = "011101" #ы  
Symbols["ь"] = "011110" #ь  
Symbols["э"] = "011111" #э  
Symbols["ю"] = "100000" #ю  
Symbols["я"] = "100001" #я  
#################################A\_L\_P###############################  
  
#################################O\_C\_T###############################  
Octs = {}  
Octs["000001"] = 1  
Octs["000010"] = 2  
Octs["000011"] = 3  
Octs["000100"] = 4  
Octs["000101"] = 5  
Octs["000110"] = 6  
Octs["000111"] = 7  
Octs["001000"] = 8  
Octs["001001"] = 9  
Octs["001010"] = 10  
Octs["001011"] = 11  
Octs["001100"] = 12  
Octs["001101"] = 13  
Octs["001110"] = 14  
Octs["001111"] = 15  
Octs["010000"] = 16  
Octs["010001"] = 17  
Octs["010010"] = 18  
Octs["010011"] = 19  
Octs["010100"] = 20  
Octs["010101"] = 21  
Octs["010110"] = 22  
Octs["010111"] = 23  
Octs["011000"] = 24  
Octs["011001"] = 25  
Octs["011010"] = 26  
Octs["011011"] = 27  
Octs["011100"] = 28  
Octs["011101"] = 29  
Octs["011110"] = 30  
Octs["011111"] = 31  
Octs["100000"] = 32  
Octs["100001"] = 33  
#################################O\_C\_T###############################  
def getSymbol(sym):  
 if isinstance(sym, str) == False:  
 sym = str(sym)  
 for key in Symbols:  
 if sym == key:  
 return Symbols[key]  
 return "-"  
  
def getAlp(value):  
 returningSym = ""  
 for key in Symbols:  
 if Symbols[key] == value:  
 returningSym = key  
 break  
  
 return returningSym  
  
def getOctFromBinValue(value):  
 if isinstance(value, str) == False:  
 value = str(value)  
 res = ""  
 for key in Octs:  
 if key == value:  
 return Octs[key]  
  
def getBinFromOctValue(value):  
 print(value)  
 if isinstance(value, int) == False:  
 value = int(value)  
 for key in Octs:  
 if Octs[key] == value:  
 return key  
  
def getXorResult(arg1, arg2):  
 if len(arg2) > len(arg1):  
 newValue = ""  
 middlePoint = len(arg2) - len(arg1)  
 while len(newValue) != middlePoint:  
 newValue += "0"  
 arg1 = newValue + arg1  
 elif len(arg1) > len(arg2):  
 newValue = ""  
 middlePoint = len(arg1) - len(arg2)  
 while len(newValue) != middlePoint:  
 newValue += "0"  
 arg2 = newValue + arg2  
 return getXor(arg1, arg2)  
  
def getXor(arg1, arg2):  
 res = ""  
 for i in range(0, len(arg1)):  
 if arg1[i] != arg2[i]:  
 res += "1"  
 elif arg1[i] == arg2[i]:  
 res += "0"  
 return res  
  
def bitwiseShift(str, count = 1):  
 swap = str[count : len(str)]  
 swap += str[0 : count]  
 return swap  
def bitwiseShiftL(str, count = 1):  
 swap = str[0 : len(str) - 1]  
 swap = str[len(str) - 1 : len(str)] + swap  
 return swap  
  
  
def getStringAllValues():  
 str = ""  
 for key in Symbols:  
 str += Symbols[key] + "\n"  
 return str

