Отчет к лабораторной работе №7

Дисциплина: Информационная безопасность

Боровикова Карина Владимировна

Содержание

# 1 Цель работы

Изучение алгоритма шифрования гаммированием

# 2 Теоретические сведения

## 2.1 Шифр гаммирования

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (например, используя операцию сложения по модулю 2). Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Метод гаммирования становится бессильным, если известен фрагмент исходного текста и соответствующая ему шифрограмма. В этом случае простым вычитанием по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Метод гаммирования с обратной связью заключается в том, что для получения сегмента гаммы используется контрольная сумма определенного участка шифруемых данных. Например, если рассматривать гамму шифра как объединение непересекающихся множеств H(j), то процесс шифрования можно пердставить следующими шагами: 1. Генерация сегмента гаммы H(1) и наложение его на соответствующий участок шифруемых данных. 2. Подсчет контрольной суммы участка, соответствующего сегменту гаммы H(1). 3. Генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гамм H(2). 4. Подсчет контрольной суммы участка данных, соответствующего сегменту данных H(2) и т.д.

# 3 Выполнение раброты

## 3.1 Реализация шифратора и дешифратора

def main():  
 dict = {"а" :0, "б" :1 , "в" :2 ,"г": 3,   
 "д" :4 ,"е" :5 ,"ё" :6 ,"ж": 7,   
 "з":8, "и": 9, "й":10, "к":11,  
 "л":12,"м": 13, "н": 14, "о": 15,  
 "п": 16,"р": 17, "с": 18, "т": 19,   
 "у": 20, "ф": 21, "х": 22, "ц": 23,  
 "ч": 24,"ш": 25, "щ": 26, "ъ": 27,  
 "ы": 28, "ь": 29, "э": 30, "ю": 31,  
 "я": 32  
 }  
 # меняем местами ключ и значение, такой словарь понадобится в будущем  
 dict2 = {v: k for k, v in dict.items()}  
  
 gamma = input('Введите гамму(на русском языке! Да, и пробелы тоже нельзя!): ').lower()  
 text = input('Введите текст для шифрования: ').lower()  
  
 print(dict)  
 print(dict2)  
  
 list\_of\_digits\_of\_text = list() #сюда будем записывать числа букв из текста  
 list\_of\_digits\_of\_gamma = list() #для гаммы  
  
 #то же самое сделаем с гаммой  
 for i in gamma:  
 list\_of\_digits\_of\_gamma.append(dict[i])  
   
 for i in text:  
 list\_of\_digits\_of\_text.append(dict[i])  
 t=0  
  
 while t == 0 :  
 if len(list\_of\_digits\_of\_gamma) < len(list\_of\_digits\_of\_text):  
 list\_of\_digits\_of\_gamma.extend(list\_of\_digits\_of\_gamma)  
 else:   
 t += 1  
  
 print(f'числа гаммы: {list\_of\_digits\_of\_gamma}' )  
 print(f'Числа текста: {list\_of\_digits\_of\_text}')  
  
 list\_of\_digits\_result = list() #сюда будем записывать результат  
 ch = 0  
 for i in text:  
 try:  
 a = dict[i] + list\_of\_digits\_of\_gamma[ch]  
 except:  
 ch=0  
 a = dict[i] + list\_of\_digits\_of\_gamma[ch]  
 if a>=33:  
 a = a%33  
 ch+=1  
 list\_of\_digits\_result.append(a)  
  
 print(f'Числа зашифрованного текста: {list\_of\_digits\_result}')  
 # теперь обратно числа представим в виде букв  
  
 text\_encrypted=''  
  
 for i in list\_of\_digits\_result:  
 text\_encrypted += dict2[i]  
 print(f'Зашифрованный текст: {text\_encrypted}')  
  
 #теперь приступим к реализации алгоритма дешифровки  
 list\_of\_digits = list()  
  
 for i in text\_encrypted:  
 list\_of\_digits.append(dict[i])  
  
 ch = 0  
   
 list\_of\_digits1 = list()  
  
 for i in list\_of\_digits:  
 a = i - list\_of\_digits\_of\_gamma[ch]  
 if a < 0:  
 a = 33 + a  
 list\_of\_digits1.append(a)  
 ch+=1  
  
 text\_decrypted = ''   
 for i in list\_of\_digits1:  
 text\_decrypted += dict2[i]  
  
 print(f'Расшифрованный текст: {text\_decrypted}')

main()

## 3.2 Вывод функции:

Введите гамму(на русском языке! Да, и пробелы тоже нельзя!): пупупупупупузаварюкакафейку  
Введите текст для шифрования: штирлицвыболван  
{'а': 0, 'б': 1, 'в': 2, 'г': 3, 'д': 4, 'е': 5, 'ё': 6, 'ж': 7, 'з': 8, 'и': 9, 'й': 10, 'к': 11, 'л': 12, 'м': 13, 'н': 14, 'о': 15, 'п': 16, 'р': 17, 'с': 18, 'т': 19, 'у': 20, 'ф': 21, 'х': 22, 'ц': 23, 'ч': 24, 'ш': 25, 'щ': 26, 'ъ': 27, 'ы': 28, 'ь': 29, 'э': 30, 'ю': 31, 'я': 32}  
{0: 'а', 1: 'б', 2: 'в', 3: 'г', 4: 'д', 5: 'е', 6: 'ё', 7: 'ж', 8: 'з', 9: 'и', 10: 'й', 11: 'к', 12: 'л', 13: 'м', 14: 'н', 15: 'о', 16: 'п', 17: 'р', 18: 'с', 19: 'т', 20: 'у', 21: 'ф', 22: 'х', 23: 'ц', 24: 'ч', 25: 'ш', 26: 'щ', 27: 'ъ', 28: 'ы', 29: 'ь', 30: 'э', 31: 'ю', 32: 'я'}  
числа гаммы: [16, 20, 16, 20, 16, 20, 16, 20, 16, 20, 16, 20, 8, 0, 2, 0, 17, 31, 11, 0, 11, 0, 21, 5, 10, 11, 20]  
Числа текста: [25, 19, 9, 17, 12, 9, 23, 2, 28, 1, 15, 12, 2, 0, 14]  
Числа зашифрованного текста: [8, 6, 25, 4, 28, 29, 6, 22, 11, 21, 31, 32, 10, 0, 16]  
Зашифрованный текст: зёшдыьёхкфюяйап  
Расшифрованный текст: штирлицвыболван

# 4 Выводы

Изучили алгоритмы шифрования на основе гаммирования

# Список литературы