Отчёт по лабораторной работе №3

Шифр гаммирования

Бакундукизе Эжид Принц НФИмд-01-21

Содержание

# 1 Цель работы

Изучение алгоритма шифрования гаммированием.

# 2 Теоретические сведения

## 2.1 Шифр гаммирования

Гаммирование — 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 [1].

Схема шифрования гаммированием представлена на рис.1.

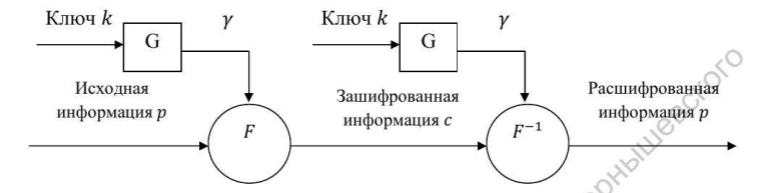


Figure 1: Схема шифрования гаммированием

Гаммирование - процедура наложения при помощи некоторой функции F на исходный текст гаммы шифра, т.е. псевдослучайной последовательности выходов генератора G. Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, т.е. известен алгоритм ее формирования. Обчно в качестве функции F берется операция поразрядного сложения по модулю два или по модулю N (N - число букв алфавита открытого текста).

Простейший генератор пседвослучайной последовательности можно представить рекуррентным соотношением (рис.2):

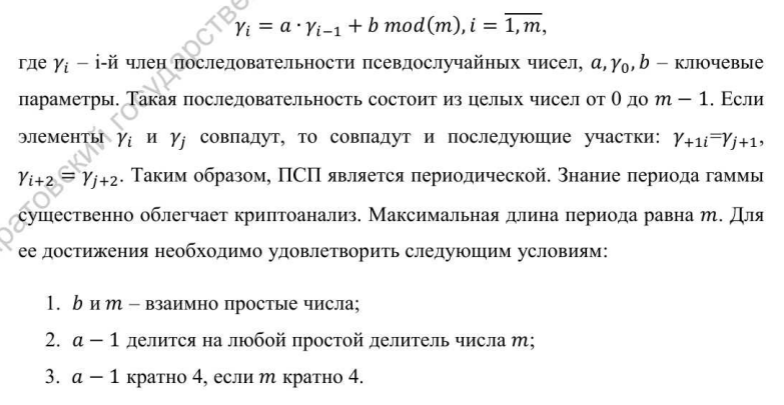


Figure 2: Простейший генератор псевдослучайной последовательности

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) [2].

Стойкость шифров, основанных на процедуре гаммирования, зависит от характеристик гаммы - длины и равномерности распределения вероятностей появления знаков гаммы. При использовании генератора псевдослучайных последовательностей получаем бесконечную гамму. Однако, возможен режим шифрования конеxной гаммы. В роли конечной гаммы выступать фраза.

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.

# 3 Выполнение работы

## 3.1 Реализация алгоритма шифрования гаммирования конечной гаммой на Python

def gamma():  
 #создаем алфавит  
 dict = {"а" :1, "б" :2 , "в" :3 ,"г" :4 ,"д" :5 ,"е" :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 для расшифровки  
 dict2 = {v: k for k, v in dict.items()}  
   
 g = input("Гамма: ").lower()  
 text = input("Текст для шифрования: ").lower()  
   
 list\_text = list() #числа букв из текста  
 list\_gamma = list() #числа букв для гаммы  
   
 #запишем числа в список  
 for i in text:  
 list\_text.append(dict[i])  
 print("Числа текста: ", list\_text)  
   
 #запишем числа для гаммы  
   
 for i in g:  
 list\_gamma.append(dict[i])  
 print("числа гаммы: ", list\_gamma)  
   
 result = list() #сюда будем записывать результат  
   
 ch = 0  
 for i in text:  
 try:  
 a = dict[i] + list\_gamma[ch]  
 except:  
 ch=0  
 a = dict[i] + list\_gamma[ch]  
 if a >=33:  
 a = a%33  
 ch+=1  
 result.append(a)  
   
 print("Числа зашифрованного текста: ", result)  
   
 # теперь обратно числа представим в виде букв  
 text2=""  
 for i in result:  
 text2 += dict2[i]  
 print("Зашифрованный текст: ", text2)  
   
 #Дешифровка  
   
 list\_text2 = list()  
   
 for i in text2:  
 list\_text2.append(dict[i])  
   
 ch = 0  
 list\_text2\_2 = list()  
 for i in list\_text2:  
 try:  
 a = i - list\_gamma[ch]  
 except:  
 ch=0  
 a = i - list\_gamma[ch]  
 if a < 1:  
 a = 33 + a  
   
 list\_text2\_2.append(a)  
 ch+=1  
   
 text\_decrypted = ""  
 for i in list\_text2\_2:  
 text\_decrypted+=dict2[i]  
   
 print("Дешифровка: ", text\_decrypted)

## 3.2 Контрольный пример

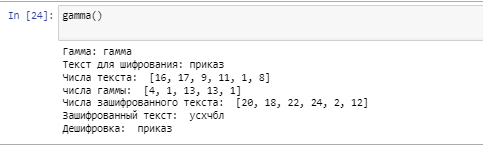


Figure 3: Работа алгоритма гаммирования

# 4 Выводы

В ходе выполнения данной лабораторной работы мы познакомились с алгоритмом шифрования гаммирования. В рамках задания программно были реализованы алгоритмы шифрования и дешифрования гаммированием конечной гаммы.

# Список литературы

1. [Шифрование методом гаммирования](http://altaev-aa.narod.ru/security/XOR.html)
2. [Режим гаммирования в блочном алгоритме шифрования](https://kabinfo.ucoz.ru/index/shifr_reshetka_kardano/0-374)