Лабораторная работа №3

Математические основы защиты информации и информационной безопасности

Леонтьева Ксения Андреевна | НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Реализовать на языке программирования шифрование гаммированием конечной гаммой.

2 Теоретическое введение

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

Простейший генератор ПСП можно представить рекуррентным соотношением:

$$\gamma_i = (a\gamma_{i-1} + b)mod(m), i = 1, ..., m,$$

где γ_i - i-й член последовательности псевдослучайных чисел, a, γ_0, b - ключевые параметры. Отметим, что ПСП является периодической.

Стойкость шифров, основанных на процедуре гаммирования, зависит от характеристик гаммы - длины и равномерности распределения вероятностей появления знаков гаммы.

Более подробно см. в [1].

3 Выполнение лабораторной работы

Списки word и gamma содержат шифруемое слово и гамму соответственно. Список alphabet заполняем буквами русского алфавита без “ё”. Далее сравниваем размерности word и gamma и если количество букв в word больше количества букв в gamma, то в список gamma_new записываем подряд gamma столько раз, сколько оно целиком входит по количеству букв в word, а затем оставшуюся часть gamma, чтобы сравнять размерности. Если же количество букв в word меньше количества букв в gamma, то в gamma_new из gamma записывается столько букв, сколько их содержится в word. Затем в списки number_word и number_gamma_new записываем номера соответствующих букв из нашего алфавита. i-я буква зашифрованного слова получается по формуле: $(number\_word[i] + number\_gamma\_new[i])mod32$, поскольку в нашем алфавите без “ё” 32 буквы. Наконец, выводим результат на экран.

Код программы (рис. 3.1).

```
import numpy as np
import math
```

```
word = 'приказ'
gamma = 'гамма'
word = list(word.replace(" ", ""))
gamma = list(gamma)

alphabet = []
for i in range(1072,1104):
    alphabet.append(chr(i))
```

```
gamma_new = []

if len(word) > len (gamma):
    for i in range(math.floor(len(word)/len(gamma))):
        gamma_new.append(gamma)
    sum(gamma_new, [])
    for j in range(len(word)%len(gamma)):
        gamma_new.append(list(gamma[j]))

if len(word) < len (gamma):
    for g in range(len(word)):
        gamma_new.append(list(gamma[g]))

gamma_new = sum(gamma_new,[])
```

```
number_word = []
number_gamma_new = []
for i in range(len(word)):
    number_word.append(alphabet.index(word[i])+1)
    number_gamma_new.append(alphabet.index(gamma_new[i])+1)
```

```
chipher = []
for i in range(len(word)):
    k = (number_word[i] + number_gamma_new[i])%32
    chipher.append(alphabet[k-1])
```

```
print(''.join(chipher))
```

```
усхчбл
```

Рис. 3.1: Реализация шифрования гаммированием

4 Выводы

В ходе выполнения данной лабораторной работы было реализовано шифрование гаммированием конечной гаммой на языке программирования Python.

Список литературы

1. Шифры гаммирования [Электронный ресурс]. URL: https://sites.google.com /site/anisimovkhv/learning/kripto/lecture/tema6.