Отчёт по лабораторной работе №3

Шифрование гаммированием

Волкова Дарья Александровна НПМмд-02-21

Содержание

Цель работы

Изучение алгоритма шифрования гаммированием конечной гаммой, а также его программная реализация.

Теоретические сведения

Частным случаем многоалфавитной подстановки является гаммирование. В этом способе шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите. Если в исходном алфавите, например, 33 символа, то сложение производится по модулю 33. Такой процесс сложения исходного текста и ключа называется в криптографии наложением гаммы.

Под гаммированием понимают наложение на открытые данные по определенному закону гаммы шифра (двоичного числа, сформированного на основе генератора случайных чисел).

Гамма шифра – псевдослучайная последовательность, вырабатываемая по определенному алгоритму, используемая для шифровки открытых данных и дешифровки шифротекста.

Принцип шифрования заключается в формировании генератором псевдослучайных чисел (ГПСЧ) гаммы шифра и наложении этой гаммы на открытые данные обратимым образом, например, путем сложения по модулю два. Процесс дешифрования данных сводится к повторной генерации гаммы шифра и наложении гаммы на зашифрованные данные. Ключом шифрования в данном случае является начальное состояние

генератора псевдослучайных чисел. При одном и том же начальном состоянии ГПСЧ будет формировать одни и те же псевдослучайные последовательности.

Перед шифрованием открытые данные обычно разбивают на блоки одинаковой длины, например, по 64 бита. Гамма шифра также вырабатывается в виде последовательности блоков той же длины.

Стойкость шифрования методом гаммирования определяется главным образом свойствами гаммы – длиной периода и равномерностью статистических характеристик. Последнее свойство обеспечивает отсутствие закономерностей в появлении различных символов в пределах периода. Полученный зашифрованный текст является достаточно трудным для раскрытия. По сути дела гамма шифра должна изменяться случайным образом для каждого шифруемого блока.

Обычно разделяют две разновидности гаммирования – с конечной и бесконечной гаммами. При хороших статистических свойствах гаммы стойкость шифрования определяется только длиной периода гаммы. При этом, если длина периода гаммы превышает длину шифруемого текста, то такой шифр теоретически является абсолютно стойким, т.е. его нельзя вскрыть при помощи статистической обработки зашифрованного текста, а можно раскрыть только прямым перебором. Криптостойкость в этом случае определяется размером ключа.

Выполнение работы

Реализация алгоритмов на языке Python

```
# Шифрование гаммированием конечной гаммой

message = input('Введите строку: ').lower()
gamma = str(input('Введите гамму: ')).lower()

alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

n = len(gamma)
m=len(message)

k = (m % n) # Количество символов которые нужно дополнить
gamma_len = '' + gamma * (m // n) + gamma[:k]
print(message, gamma_len, sep='\n')

c=[]

for i in range(m):
    c.append(alphabet.find(message[i])+(alphabet.find(gamma_len[i])+1)%33)

encrypted = ''

for i in range(len(c)):
    encrypted+=alphabet[c[i]]
```

```
print('Криптограмма:', encrypted)
```

Контрольный пример

```
# Шифрование гаммированием конечной гаммой
```

```
message = input('Введите строку: ').lower()
gamma = str(input('Введите гамму: ')).lower()
```

```
Введите строку: приказ
Введите гамму: гамма
```

```
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'
```

```
n = len(gamma)
m=len(message)
```

```
k = (m % n) # Количество символов которые нужно дополнить
gamma_len = '' + gamma * (m // n) + gamma[:k]
print(message, gamma_len, sep='\n')
```

```
приказ
гаммаг
```

```
c=[]
```

```
for i in range(m):
    c.append(alphabet.find(message[i])+(alphabet.find(gamma_len[i])+1)%33)
```

```
encrypted = ''
```

```
for i in range(len(c)):
    encrypted+=alphabet[c[i]]
```

```
print('Криптограмма:', encrypted)
```

```
Криптограмма: усхчбл
```

Пример работы алгоритма

Выводы

В ходе выполнения работы удалось изучить алгоритм шифрования гаммированием конечной гаммой, а также реализовать данный алгоритм программно на языке Python.

Список литературы

1. Гаммирование
2. Методы гаммирования
3. Шифрование методом гаммирования
4. Шифрование методом гаммирования