Отчёт по лабораторной работе №7

Шифр гаммирования

Таубер Кирилл Олегович НПИбд-02-19

Содержание

[Цель работы 1](#_Toc117373403)

[Теоретические сведения 1](#_Toc117373404)

[Шифр гаммирования 1](#_Toc117373405)

[Выполнение работы 2](#_Toc117373406)

[Реализация шифратора и дешифратора Python 2](#_Toc117373407)

[Контрольный пример 3](#_Toc117373408)

[Выводы 3](#_Toc117373409)

[Список литературы 3](#_Toc117373410)

# Цель работы

Изучение алгоритма шифрования гаммированием

# Теоретические сведения

## 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.

# Выполнение работы

## Реализация шифратора и дешифратора Python

alphabeth = "abcdefghijklmnopqrstuvwxyz"  
def encrypt(text, gamma):  
 textLen = len(text)  
 gammaLen = len(gamma)  
 keyText = []  
 for i in range(textLen // gammaLen):  
 for symb in gamma:  
 keyText.append(symb)  
 for i in range(textLen % gammaLen):  
 keyText.append(gamma[i])  
 code = []  
 for i in range(textLen):  
 code.append(alphabeth[(alphabeth.index(text[i]) + alphabeth.index(keyText[i])) % 26])  
 return code  
def decrypt(code, gamma):  
 codeLen = len(code)  
 gammaLen = len(gamma)  
 keyText = []  
 for i in range(codeLen // gammaLen):  
 for symb in gamma:  
 keyText.append(symb)  
 for i in range(codeLen % gammaLen):  
 keyText.append(gamma[i])  
 text = []  
 for i in range(codeLen):  
 text.append(alphabeth[(alphabeth.index(code[i]) - alphabeth.index(keyText[i]) + 26) % 26])   
 return text

## Контрольный пример

Работа алгоритма гаммирования

Работа алгоритма гаммирования

# Выводы

Изучили алгоритмы шифрования на основе гаммирования

# 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)