Отчёт по лабораторной работе №3

Дисциплина: Математические основы защиты информации и информационной безопасности

Полиенко Анастасия Николаевна, НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Изучить шифрование гаммированием.

2 Задание

Реализовать алгоритм шифрования гаммированием конечной гаммой.

3 Теоретическое введение

Из всех схем шифрования простейшей и наиболее надежной является схема однократного использования. Формируется m-разрядная случайная двоичная последовательность - ключ шифра. Отправитель производит побитовое сложение по модулю два $(mod 2)$ ключа

$$k = k_1 k_2 \dots k_i \dots k_m$$

и m-разрядной двоичной последовательности

$$p = p_1 p_2 \dots p_i \dots p_m$$

соответствующей посылаемому сообщению:

$$c_i = p_i \oplus k_i, i = \overline{1, m}$$

где p - -й бит исходного текста, $_i$ - -й бит ключа, $\oplus$ - операция побитового сложения (XOR), c_i - -й бит получившейся криптограммы

$$c = c_1 c_2 \dots c_i \dots c_m$$

Операция побитного сложения является обратимой, т.е. $(x \oplus y) \oplus y)$, поэтому дешифрование осуществляется повторным применением операции $\oplus$ к криптограмме:

$$p_i = c_i \oplus k_i, i = \overline{1, m}$$

Основным недостатком такой схемы является равенство объема ключевой информации и суммарного обьема передаваемых сообщений. Данный недостаток можно убрать, использовав ключ в качестве «зародыша», порождающего значительно более длинную ключевую последовательность. Такая схема называется *гаммированием*.

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

$$\gamma_i = a\gamma_{i-1} + bmod(m), i = \overline{1, m}$$

где γ_i - i-й член последовательности псевдослучайных чисел, , γ_0, b - ключевые параметры. Такая последовательность состоит из целых чисел от 0 до $m - 1$. Если элементы γ_i и γ_j совпадут, то совпадут и последующие участки: $\gamma_{i+1} = \gamma_{j+1}, \gamma i + 2 = \gamma_{j+2}$. Таким образом, ПСІП является периодической. Знание периода гаммы существенно облегчает криптоанализ. Максимальная длина периода равна *m*. Для достижения необходимо удовлетворить следующим условиям:

1. *b* и *m* - взаимно простые числа;
2. -1 делится на любой простой делитель числа *m*;
3. -1 кратно 4, если кратно 4.

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

При использовании генератора ПСП получаем бесконечную гамму. Однако, возможен режим шифрования конечной гаммы. В роли конечной гаммы может выступать фраза. Как и ранее, используется алфавитный порядок букв, т.е. буква «а» имеет порядковый номер 1, «б» - 2 и т.д.

Более подробно см. в [1–6].

4 Выполнение лабораторной работы

Реализуем алгоритм шифрования гаммированием конечной гаммой (рис. 4.1).

```
alpha = ' АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
message = input("Введите сообщение:")
key = input("Введите ключ:")

#уравнивание ключа с сообщением
i = 0
while len(message) != len(key):
    key += key[i]
    i += 1

#печать результата
res = ''
for i in range(len(message)):
    res += alpha[(alpha.index(message[i]) + alpha.index(key[i]))
                 % (len(alpha) - 1)]
print(res)
```

```
Введите сообщение:ПРИКАЗ
Введите ключ:ГАММА
УСХЧБЛ
```

Рис. 4.1: Гаммирование

5 Выводы

Изучила шифрование гаммированием.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: https://www.gnu.org/software/bash/manual/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.