

Отчёт по лабораторной работе №7

Шифр гаммирования

Цэдулма Цыденова НБИ-01-19

Содержание

1	Цель работы	4
2	Теоретические сведения	5
2.1	Шифр гаммирования	5
3	Выполнение работы	7
3.1	Реализация шифратора и дешифратора Java	7
3.2	Контрольный пример	8
4	Выводы	9
	Список литературы	10

List of Figures

3.1 Работа алгоритма гаммирования 8

1 Цель работы

Изучение алгоритма шифрования гаммированием

2 Теоретические сведения

2.1 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 представить 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.

3 Выполнение работы

3.1 Реализация шифратора и дешифратора Java

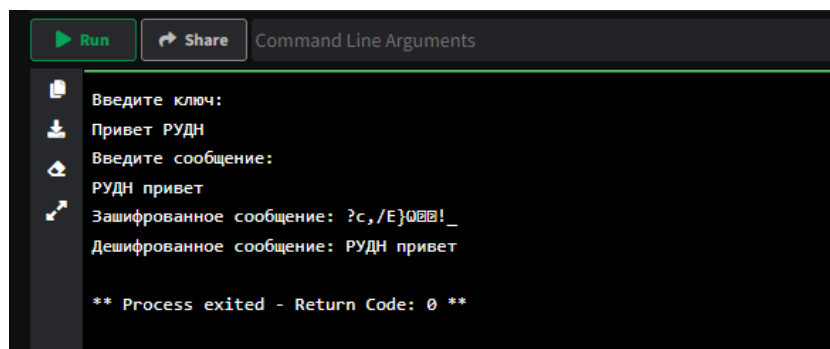
```
import java.util.Scanner;

public class Main {
    private static Scanner sc = new Scanner(System.in);

    public static String enc(String message, String key) {
        String output = "";
        for (int i = 0; i < message.length(); i++) {
            output += String.valueOf((char)(message.charAt(i) ^ key.charAt(i % key.length())));
        }
        return output;
    }

    public static void main(String[] args) {
        System.out.print("Введите ключ: ");
        String key = sc.nextLine();
        System.out.print("Введите сообщение: ");
        String input = sc.nextLine();
        System.out.printf("Зашифрованное сообщение: %s", enc(input, key));
        System.out.printf("\nДешифрованное сообщение: %s", enc(enc(input, key), key));
    }
}
```

3.2 Контрольный пример



```
Run Share Command Line Arguments
Введите ключ:
Привет РУДН
Введите сообщение:
РУДН привет
Зашифрованное сообщение: ?с,/Е}000!_
Дешифрованное сообщение: РУДН привет

** Process exited - Return Code: 0 **
```

Figure 3.1: Работа алгоритма гаммирования

4 Выводы

Изучили алгоритмы шифрования на основе гаммирования

Список литературы

1. Шифрование методом гаммирования
2. Режим гаммирования в блочном алгоритме шифрования