

Отчет по лабораторной работе №7

по предмету Информационная безопасность

Алхимова Дарья Сергеевна

Содержание

Цель работы	4
Задание	5
Теоретическое введение	6
Выполнение лабораторной работы	8
Выводы	13
Список литературы	14

Список иллюстраций

1	Создание файла программы	8
2	Фрагмент программы гаммирования	8
3	Листинг программы 1/7	9
4	Листинг программы 2/7	9
5	Листинг программы 3/7	10
6	Листинг программы 4/7	10
7	Листинг программы 5/7	10
8	Листинг программы 6/7	11
9	Листинг программы 7/7	11
10	Запуск программы по сценарию 1	12
11	Запуск программы по сценарию 2	12

Цель работы

Освоить на практике применение режима однократного гаммирования.

Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Теоретическое введение

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком $+$) между элементами гаммы и элементами подлежащего сокрытию текста. Напомним, как работает операция XOR над битами: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 0$. Такой метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой.

Открытый текст имеет символьный вид, а ключ — шестнадцатеричное представление. Ключ также можно представить в символьном виде, воспользовавшись таблицей ASCII-кодов.

Необходимые и достаточные условия абсолютной стойкости шифра:

- + полная случайность ключа;
- + равенство длин ключа и открытого текста;
- + однократное использование ключа.

Метод гаммирования становится бессильным, если известен фрагмент исход-

ного текста и соответствующая ему шифрограмма. В этом случае простым вычитанием по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Алгоритм гаммирования:

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.

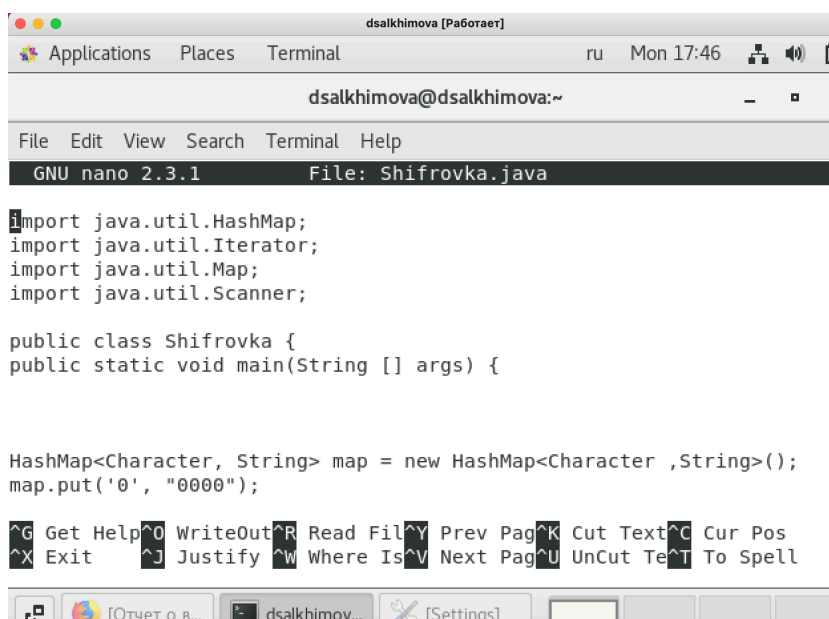
Выполнение лабораторной работы

1. Создала файл программы и открыла его для заполнения кодом. (рис. 1).

```
[dsalkhimova@dsalkhimova ~]$ touch Shifrovka.java  
[dsalkhimova@dsalkhimova ~]$ nano Shifrovka.java  
[dsalkhimova@dsalkhimova ~]$
```

Рис. 1: Создание файла программы

2. Заполнила файл кодом программы гаммирования (рис. 2).



```
dsalkhimova [Работаer]  
Applications Places Terminal ru Mon 17:46  
dsalkhimova@dsalkhimova:~  
File Edit View Search Terminal Help  
GNU nano 2.3.1 File: Shifrovka.java  
  
import java.util.HashMap;  
import java.util.Iterator;  
import java.util.Map;  
import java.util.Scanner;  
  
public class Shifrovka {  
public static void main(String [] args) {  
  
  
  
  
HashMap<Character, String> map = new HashMap<Character ,String>();  
map.put('0', "0000");  
  
^G Get Help ^O WriteOut ^R Read Fil ^Y Prev Pag ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Pag ^U UnCut Te ^T To Spell  
[Очет о в... dsalkhimov... [Settings]
```

Рис. 2: Фрагмент программы гаммирования

Полный текст программы (рис. 3, рис. 4, рис. 5, рис. 6, рис. 7, рис. 8, рис. 9):


```

1 import java.util.HashMap;
2 import java.util.Iterator;
3 import java.util.Map;
4 import java.util.Scanner;
5
6 public class Shifrovka {
7     public static void main(String [] args) {
8
9
10
11     HashMap<Character, String> map = new HashMap<Character, String>();
12     map.put('0', "0000");
13     map.put('1', "0001");
14     map.put('2', "0010");
15     map.put('3', "0011");
16     map.put('4', "0100");
17     map.put('5', "0101");
18     map.put('6', "0110");
19     map.put('7', "0111");
20     map.put('8', "1000");
21     map.put('9', "1001");
22     map.put('A', "1010");
23     map.put('B', "1011");
24     map.put('C', "1100");
25     map.put('D', "1101");

```

Рис. 3: Листинг программы 1/7

```

24 map.put('C', "1100");
25 map.put('D', "1101");
26 map.put('E', "1110");
27 map.put('F', "1111");
28
29 String text="";
30 String cipher;
31 String cipher2;
32 Scanner in = new Scanner(System.in);
33 System.out.println("введите '1' если хотите определить шифротекст по ключу и открытому тексту \n или
    '2' если хотите определить ключ по открытому тексту и шифротексту: ");
34 int input = in.nextInt();
35 if(input==1) {
36     Scanner in2 = new Scanner(System.in);
37     System.out.println("введите ключ шифрования (ключ должен быть в шестнадцатеричной системе счисления и
    должен быть разделен пробелами):");
38     cipher = in2.nextLine();
39     System.out.println("введите открытый текст (размерность текста должна совпадать с размерностью
    ключа):");
40     cipher2 = in2.nextLine();
41     cipher2 = characterTo16(cipher2, map);
42 } else {
43     Scanner in2 = new Scanner(System.in);
44     System.out.println("введите шифротекст : ");
45     cipher = in2.nextLine();
46

```

Рис. 4: Листинг программы 2/7

```

47 System.out.println("введите открытый текст(размерность текста должна совпадать с размерностью
шифротекста) :");
48
49 cipher2= in2.nextLine();
50 cipher2= characterto16(cipher2,map);
51 }
52
53 String shifr = shifrovanie(cipher,cipher2,map);
54
55 if(input==1) {
56 System.out.println("шифротекст : "+shifr);
57 }else {
58 System.out.println("ключ : "+shifr);
59 }
60
61 }
62
63 public static String characterto16 (String cipher,HashMap<Character, String> map)
64 {
65 char[] chararray = cipher.toCharArray();
66 String finalcode="";
67 for(int i=0;i<chararray.length;i++) {
68 char character = chararray[i];
69 int ascii = (int) character;
70 String code = Integer.toString(ascii,2);

```

Рис. 5: Листинг программы 3/7

```

71 String curcode=code;
72 for(int j=0;j<8-code.length();j++) {
73 curcode="0"+curcode;
74 }
75 code= curcode;
76 String val = code.substring(0, 4);
77 String val2= code.substring(4);
78 char nval=' ';
79 char nval2=' ';
80 Iterator it = map.entrySet().iterator();
81
82 while (it.hasNext()) {
83 Map.Entry pair = (Map.Entry)it.next();
84 if(pair.getValue().equals(val)) {
85 nval=(char)pair.getKey();
86 }
87
88 if(pair.getValue().equals(val2)) {
89 nval2=(char)pair.getKey();
90 }
91 }
92
93 String v = String.valueOf(nval)+String.valueOf(nval2);
94 finalcode=finalcode+v+" ";
95
--

```

Рис. 6: Листинг программы 4/7

```

96 }
97
98 return finalcode;
99
100 }
101 public static String shifrovanie(String cipher, String cipher2,HashMap<Character,String> map) {
102
103
104 String[] splt = cipher.split("\\s+");
105 String[] splt2 = cipher2.split("\\s+");
106
107 String finalcode="";
108 for(int i=0;i<splt.length;i++) {
109
110 char[] symbols = splt[i].toCharArray();
111 String symbol = map.get(symbols[0])+map.get(symbols[1]);
112
113 char[] symbols2 = splt2[i].toCharArray();
114 String symbol2 = map.get(symbols2[0])+map.get(symbols2[1]);
115
116 String newsymbol="";
117 for(int j=0;j<symbol2.length();j++) {
118
119 int number = Character.digit(symbol2.charAt(j), 10);
120 int number2 = Character.digit(symbol.charAt(i), 10);

```

Рис. 7: Листинг программы 5/7

```

119 int number = Character.digit(symbol.charAt(j), 10);
120 int number2 = Character.digit(symbol.charAt(j), 10);
121
122 newsymbol+=number^number2;
123
124
125 }
126
127 String val = newsymbol.substring(0, 4);
128 String val2 = newsymbol.substring(4);
129 char nval = ' ';
130 char nval2 = ' ';
131 Iterator it = map.entrySet().iterator();
132
133 while (it.hasNext()) {
134 Map.Entry pair = (Map.Entry)it.next();
135 if(pair.getValue().equals(val)) {
136 nval=(char)pair.getKey();
137 }
138
139 if(pair.getValue().equals(val2)) {
140 nval2=(char)pair.getKey();
141 }
142
143 }
144 |

```

Рис. 8: Листинг программы 6/7

```

144
145 String v = String.valueOf(nval)+String.valueOf(nval2);
146 finalcode=finalcode+v+" ";
147
148
149 }
150
151 return finalcode;
152 }
153
154 |

```

Рис. 9: Листинг программы 7/7

3. Проверила правильность выполнения программы гаммирования по первому сценарию (см. раздел Задание) (рис. 10).

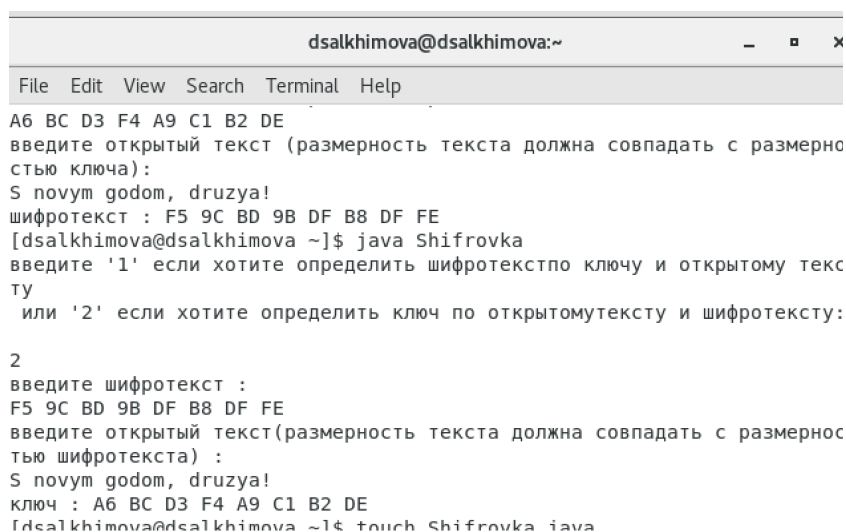
```

[dsalkhimova@dsalkhimova ~]$ javac Shifrovka.java
[dsalkhimova@dsalkhimova ~]$ java Shifrovka
введите '1' если хотите определить шифротекст по ключу и открытому тексту
или '2' если хотите определить ключ по открытому тексту и шифротексту:
1
введите ключ шифрования (ключ должен быть в шестнадцатеричной системе
счисления и должен быть разделен пробелами):
A6 BC D3 F4 A9 C1 B2 DE
введите открытый текст (размерность текста должна совпадать с размерно-
стью ключа):
S novym godom, druzya!
шифротекст : F5 9C BD 9B DF B8 DF FE

```

Рис. 10: Запуск программы по сценарию 1

4. Проверила правильность выполнения программы гаммирования по второму сценарию (см. раздел Задание) (рис. 11).



```

dsalkhimova@dsalkhimova:~
File Edit View Search Terminal Help
A6 BC D3 F4 A9 C1 B2 DE
введите открытый текст (размерность текста должна совпадать с размерно-
стью ключа):
S novym godom, druzya!
шифротекст : F5 9C BD 9B DF B8 DF FE
[dsalkhimova@dsalkhimova ~]$ java Shifrovka
введите '1' если хотите определить шифротекст по ключу и открытому тексту
или '2' если хотите определить ключ по открытому тексту и шифротексту:
2
введите шифротекст :
F5 9C BD 9B DF B8 DF FE
введите открытый текст(размерность текста должна совпадать с размернос-
тью шифротекста) :
S novym godom, druzya!
ключ : A6 BC D3 F4 A9 C1 B2 DE
[dsalkhimova@dsalkhimova ~]$ touch Shifrovka.java

```

Рис. 11: Запуск программы по сценарию 2

Выводы

В процессе выполнения данной лабораторной работы я приобрела навыки применения режима однократного гаммирования.

Список литературы

1. Описание лабораторной работы 7 - URL: https://esystem.rudn.ru/pluginfile.php/1652175/mod_resource/content/2/007-lab_crypto-gamma.pdf
2. Установка javac - URL: <https://stackoverflow.com/questions/5407703/javac-command-not-found>