

Лабораторная работа №3: отчет.

Шифры простой замены

Евдокимов Максим Михайлович. Группа - НФИмд-01-24.

Содержание

Цели и задачи работы	4
Цель лабораторной работы	4
Задание	4
Теоретическое введение	5
Справка по способам	5
Алгоритм Гаммирования: Бесконечный случай	6
Алгоритм Гаммирования: Конечная гамма	6
Ход работы	7
Вариант 1	7
Результат 1	9
Вариант 2	9
Результат 2	12
Выводы по проделанной работе	13
Вывод	13
Список литературы	14

Список иллюстраций

1	Первый вариант гаммирования	8
2	Результат первого варианта	9
3	Список используемых символов	10
4	Второй вариант гаммирования	11
5	Результат второго варианта	12

Цели и задачи работы

Цель лабораторной работы

Изучить понятие Гаммирование и его особенности и типы.

Задание

Реализовать алгоритм шифрования гаммированием конечной гаммой.

Теоретическое введение

Гаммирование — это метод шифрования, при котором каждый символ открытого текста складывается с соответствующим символом гаммы (ключа) для получения зашифрованного текста. Гамма — это последовательность символов, которая используется для шифрования.

Справка по способам

Процедура наложения гаммы на исходный текст может быть различной. Например, символы исходного текста и гаммы заменяются цифровыми эквивалентами, которые затем складываются или вычитаются. Или символы исходного текста и гаммы представляются в виде двоичного кода, затем соответствующие разряды складываются по модулю 2. Также можно использовать преобразование по правилу логической эквивалентности (неэквивалентности) и другие логические операции.

В качестве гаммы может быть использована любая последовательность случайных символов, например, последовательность цифр числа пи (3,14...). При ручном шифровании для формирования случайной цифровой последовательности любой длины можно использовать фортунку-рулетку, раскручивая стрелку. Шкала вертушки разделена на 10 равных секторов, которые помечены цифрами от 0 до 9.

Алгоритм Гаммирования: Бесконечный случай

- Гамма: Бесконечная псевдослучайная последовательность символов.

Шифрование:

1. Для каждого символа открытого текста $P[i]$ и соответствующего символа гаммы $G[i]$.
2. Вычислить зашифрованный символ $C[i] = P[i] \boxtimes G[i]$ (побитовое XOR).

Дешифрование:

1. Для каждого символа зашифрованного текста $C[i]$ и соответствующего символа гаммы $G[i]$.
2. Вычислить открытый символ $P[i] = C[i] \boxtimes G[i]$ (побитовое XOR).

Алгоритм Гаммирования: Конечная гамма

- Гамма: Конечная последовательность символов, которая повторяется, если длина открытого текста превышает длину гаммы.

Шифрование:

1. Для каждого символа открытого текста $P[i]$ и соответствующего символа гаммы $G[i \% \text{len}(G)]$.
2. Вычислить зашифрованный символ $C[i] = P[i] \boxtimes G[i \% \text{len}(G)]$ (побитовое XOR).

Дешифрование:

1. Для каждого символа зашифрованного текста $C[i]$ и соответствующего символа гаммы $G[i \% \text{len}(G)]$.
2. Вычислить открытый символ $P[i] = C[i] \boxtimes G[i \% \text{len}(G)]$ (побитовое XOR).

Ход работы

В ходе выполнения задания было создано 2 вариант простейшего гаммирования на основе логической операции XOR.

Вариант 1

В первом варианте реализации метода Гаммирования я использовал возможности Julia для использования всего диапазона всех ASCII/Unicode символов.

```

1 # функция Гаммирования с любыми символами из ASCII/Unicode
2 function gamming_encryptV1(t::String, k::String)
3     text_b = map{Int, collect(t)}
4     key_b = map{Int, collect(k)}
5     lenT, lenK = length(text_b), length(key_b)
6
7     encrypted_b = [xor(text_b[i], key_b[1+(i-1)%lenK]) for i in 1:lenT]
8     result = map{Char, encrypted_b}
9
10    return join(result)
11 end
12
13 text = "ПРИКАЗ"
14 key = "ГАММА"
15
16 encrypted_text = gamming_encryptV1(text, key)
17 println("Зашифрованный текст: ", encrypted_text)
18
19 decrypted_text = gamming_encryptV1(encrypted_text, key)
20 println("Расшифрованный текст: ", decrypted_text)

```

Рис. 1: Первый вариант гаммирования

Результат 1

```
13 text = "ПРИКАЗ"
14 key = "ГАММА"
15
16 encrypted_text = gamming_encryptV1(text, key)
17 println("Зашифрованный текст: ", encrypted_text)
18
19 decrypted_text = gamming_encryptV1(encrypted_text, key)
20 println("Расшифрованный текст: ", decrypted_text)
```

Зашифрованный текст: 0000

Расшифрованный текст: ПРИКАЗ

Рис. 2: Результат первого варианта

Вариант 2

Для второго варианта реализации метода Гаммирования уже был создан список символов состоящих из обычных и заглавных букв русского и английского

алфавита.

```
1 matr = vcat('a','z', 'A','Z', 'a','a', 'A','A')
2 for i in 1:length(matr)-1
3     println(i, ", ", matr[i], " | ", i+1, ", ", matr[i+1], " | ", i+2, ", ", matr[i+2], " | ", i+3, ", ", matr[i+3], " | ", i+4, ", ", matr[i+4], " | ")
4 end
```

1 a	2 b	3 c	4 d	5 e
6 f	7 g	8 h	9 i	10 j
11 k	12 l	13 m	14 n	15 o
16 p	17 q	18 r	19 s	20 t
21 u	22 v	23 w	24 x	25 y
26 z	27 A	28 B	29 C	30 D
31 E	32 F	33 G	34 H	35 I
36 J	37 K	38 L	39 M	40 N
41 O	42 P	43 Q	44 R	45 S
46 T	47 U	48 V	49 W	50 X
51 Y	52 Z	53 a	54 b	55 v
56 r	57 d	58 e	59 ж	60 э
61 и	62 й	63 к	64 л	65 н
66 н	67 о	68 п	69 р	70 с
71 т	72 у	73 ф	74 х	75 ц
76 ч	77 ш	78 щ	79 ь	80 ы
81 ь	82 э	83 ю	84 я	85 A
86 Б	87 В	88 Г	89 Д	90 Е
91 Ж	92 З	93 И	94 Й	95 К
96 Л	97 М	98 Н	99 О	100 П
101 Р	102 С	103 Т	104 У	105 Ф
106 Х	107 Ц	108 Ч	109 Ш	110 Щ
111 Ъ	112 Ы	113 Ь	114 Э	115 Ю

Рис. 3: Список используемых символов

```

1 # функция Гаммирования с символами из диапазона
2 function gamming_encryptV2(t::String, k::String)
3     alf = vcat('_', 'a':'z', 'A':'Z', 'а':'я', 'А':'Я')
4     text_b = [findfirst(x -> x == elem, alf)-1 for elem in collect(t)]
5     key_b = [findfirst(x -> x == elem, alf)-1 for elem in collect(k)]
6     lenT, lenK = length(text_b), length(key_b)
7
8     encrypted_b = [xor(text_b[i], key_b[1+(i-1)%lenK])+1 for i in 1:lenT]
9     result = alf[encrypted_b]
10    return join(result)
11 end
12
13 text = "ПРИКАЗ"
14 key = "ГАММА"
15
16 encrypted_text = gamming_encryptV2(text, key)
17 println("Зашифрованный текст: ", encrypted_text)
18
19 decrypted_text = gamming_encryptV2(encrypted_text, key)
20 println("Расшифрованный текст: ", decrypted_text)

```

Рис. 4: Второй вариант гаммирования

Результат 2

```
13 text = "ПРИКАЗ"
14 key = "ГАММА"
15
16 encrypted_text = gamming_encryptV2(text, key)
17 println("Зашифрованный текст: ", encrypted_text)
18
19 decrypted_text = gamming_encryptV2(encrypted_text, key)
20 println("Расшифрованный текст: ", decrypted_text)
```

Зашифрованный текст: зVзй_d

Расшифрованный текст: ПРИКАЗ

Рис. 5: Результат второго варианта

Выводы по проделанной работе

Вывод

В ходе выполнения лабораторной работы было изучено понятие гаммирования и его принципах работы. Применены некоторые способы его реализации и рассмотрены разные его типы.

Список литературы

1. Шифрование методом гаммирования
2. МАТЕМАТИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ Рацеева С.М. (стр. 292)
3. 88-97, 174. метод гаммирования Евгений Александрович Лыгин