

Отчёт по лабораторной работе 3

МОЗИИБ

Папикян Гагик Тигранович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Шифрование гаммированием	7
3.2	Шифрование конечной гаммой	7
4	Выполнение лабораторной работы	8
5	Выводы	10

List of Figures

4.1	Выполнение лабораторной работы	9
-----	--	---

List of Tables

1 Цель работы

Познакомиться с принципом шифрования через гаммирование, посредством реализации алгоритма шифрования с конечной гаммой

2 Задание

- 1) Реализовать алгоритм шифрования конечной гаммой

3 Теоретическое введение

3.1 Шифрование гаммированием

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле.

3.2 Шифрование конечной гаммой

Шифрование конечной гаммой использует определенный ключ маленькой длины, и циклическим его повторением получает гамму, равную по длине входному сообщению

4 Выполнение лабораторной работы

Был написан следующий скрипт на javascript

```
const alphabet = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM !'
const reversedAlphabet = alphabet.split('').reverse().join('')

function gammaCipher(msg, gamma, decode = false){
  const swappedAlphabet = decode ? reversedAlphabet : alphabet
  const fullGamma = gamma.repeat(Math.ceil(msg.length / gamma.length))

  const msgCodes = msg.split('').map(char=>swappedAlphabet.indexOf(char))
  const fullGammaCodes = fullGamma.split('').map(char=>alphabet.indexOf(char))

  const resCodes = msgCodes.map((msgCode, idx)
    => (msgCode+fullGammaCodes[idx])%alphabet.length)
  const res = resCodes.map( code=>swappedAlphabet[code] ).join('')

  return res
}

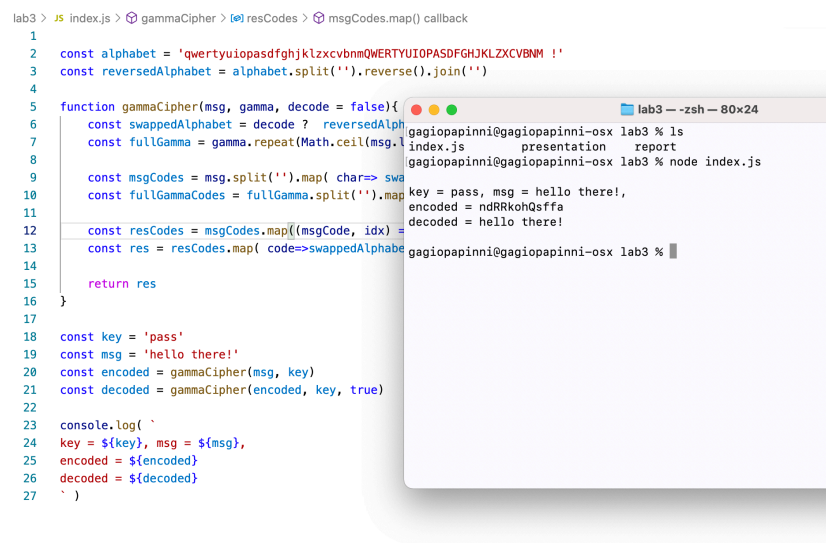
const key = 'pass'
const msg = 'hello there!'
const encoded = gammaCipher(msg, key)
```



```
const decoded = gammaCipher(encoded, key, true)
```

```
console.log( `
key = ${key}, msg = ${msg},
encoded = ${encoded}
decoded = ${decoded}
` )
```

Результат исполнения скрипта приведен на рисунке 1 (рис. 4.1)



```
lab3 > ./index.js > gammaCipher > resCodes > msgCodes.map() callback
1
2 const alphabet = 'qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM !'
3 const reversedAlphabet = alphabet.split('').reverse().join('')
4
5 function gammaCipher(msg, gamma, decode = false){
6   const swappedAlphabet = decode ? reversedAlphabet : alphabet
7   const fullGamma = gamma.repeat(Math.ceil(msg.length / gamma))
8   const msgCodes = msg.split('').map( char=> swappedAlphabet.indexOf(char))
9   const fullGammaCodes = fullGamma.split('').map( char=> char.charCodeAt(0))
10  const resCodes = msgCodes.map((msgCode, idx) => {
11    const code = fullGammaCodes[idx % fullGammaCodes.length]
12    const resCode = (msgCode + code) % swappedAlphabet.length
13    return swappedAlphabet[resCode]
14  })
15  return res
16 }
17
18 const key = 'pass'
19 const msg = 'hello there!'
20 const encoded = gammaCipher(msg, key)
21 const decoded = gammaCipher(encoded, key, true)
22
23 console.log( `
24 key = ${key}, msg = ${msg},
25 encoded = ${encoded}
26 decoded = ${decoded}
27 ` )
```

Figure 4.1: Выполнение лабораторной работы

5 Выводы

Был реализован алгоритм шифрования конечной гаммой

Был использован фиксированный алфавит, состоящий из символов

“qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM !”

На рисунке 4.1 в окне терминала было показано, как текст “hello there!” зашифровывается и расшифровывается реализованным алгоритмом