Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Защита информации и надёжность информационных систем»

**Лабораторная работа №2**

**Тема «ЭЛЕМЕНТЫ ТЕОРИИ ИНФОРМАЦИИ. ИНФОРМАТИВНОСТЬ ДАННЫХ В РАЗЛИЧНЫХ КОДИРОВКАХ»**

Выполнил:

Студент 3 курса 7 группы ФИТ

Тимошенко Д. В.   
 Проверила:   
 асс. Николайчук А. С.

Минск 2023

**Цель работы:** приобретение практических навыков трансформации данных и сопоставление энтропийных свойств используемых при этом алфавитов

**Задание на лабораторную работу**

Создать собственное приложение или воспользоваться Base64-онлайн-кодировщиком, с помощью которого конвертировать произвольный документ (а) на латинице (можно использовать документ из лабораторной работы № 1) в документ (б) формата base64. В качестве входных данных можно использовать указанный преподавателем вариант из списка:

• входные параметры;

• текстовый файл (\*.txt);

• документ Word (\*.doc);

• документ Word (\*.docx);

• документ PowerPoint (\*.ppt, \*.pptx);

• архив (\*.zip);

• текстовая строка;

• случайное число (от 999999);

• PDF-файл;

• архив (\*.rar);

• архив (\*.7z).

2. С помощью приложения, созданного в лабораторной работе № 1, получить распределение частотных свойств алфавитов по документам (а) и (б). Вычислить энтропию Хартли и Шеннона, а также избыточность алфавитов. Объяснить полученный результат.

3. Написать функцию, которая принимает в качестве аргументов два буфера (а и b) одинакового размера и возвращает XOR (собственная фамилия (а) и имя (b); при разной длине меньшую дополнить нулями). Входные аргументы представлять: 1) в кодах ASCII; 2) в кодах base64. Что будет результатом операции аXORbXORb? **Ответ: результатом будет фамилия (а)**

4.При написании не использовать стандартные функции языка программирования. Итоговые данные сравнить с результатами использования стандартных функций языка программирования (если они есть).

Приложение написано на языке программирования JavaScript. Ниже представлен листинг функций, используемых во всех задания, листинг 1.1.

|  |
| --- |
| const fs = require('fs');  const path = require('path');  function countFrequency(text) {  let freqMap = {};  for (let char of text) {  if (char in freqMap) {  freqMap[char]++;  } else {  freqMap[char] = 1;  }  }  return freqMap;  }  function calculateEntropy(freqMap, totalChars) {  let entropy = 0;  for (let char in freqMap) {  let p = freqMap[char] / totalChars;  entropy -= p \* Math.log2(p);  }  return entropy;  }  function calculateHartleyEntropy(alphabetSize) {  return Math.log2(alphabetSize);  }  function calculateRedundancy(hartleyEntropy, shannonEntropy) {  return 1 - (shannonEntropy / hartleyEntropy);  }  function processFile(filePath) {  fs.readFile(filePath, 'utf8', function(err, data) {  console.log(filePath);  if (err) throw err;  let freqMap = countFrequency(data);  let totalChars = data.length;  let alphabetSize = Object.keys(freqMap).length;  let shannonEntropy = calculateEntropy(freqMap, totalChars);  let hartleyEntropy = calculateHartleyEntropy(alphabetSize);  let redundancy = calculateRedundancy(hartleyEntropy, shannonEntropy);  console.log(`Shannon Entropy: ${shannonEntropy}`);  console.log(`Hartley Entropy: ${hartleyEntropy}`);  console.log(`Redundancy: ${redundancy}`);  });  }  function xor(a, b) {  let result = '';  for (let i = 0; i < a.length; i++) {  result += String.fromCharCode(a.charCodeAt(i) ^ b.charCodeAt(i));  }  return result;  }  function padWithZeros(a, b) {  while (a.length < b.length) {  a += '\0';  }  while (b.length < a.length) {  b += '\0';  }  return [a, b];  }  function processInputs(input1, input2, encoding) {  console.log(encoding);  let a = Buffer.from(input1, encoding).toString();  let b = Buffer.from(input2, encoding).toString();  [a, b] = padWithZeros(a, b);  let xorResult = xor(a, b);  console.log(`XOR result: ${xorResult}`);  } |

Листинг 1.1 – используемые функции

А также код выполнения самих заданий лабораторной работы, листинг 1.2.

|  |
| --- |
| let filePath = path.join(\_\_dirname, 'encoded.txt');  processFile(filePath);  filePath = path.join(\_\_dirname, 'sourceFile.docx');  processFile(filePath);  let a = 'Timoshenko';  let b = 'Dmitry';  processInputs(a, b, 'ascii');  a = 'VGltb3NoZW5rbw==';  b = 'RG1pdHJ5';  processInputs(a, b, 'base64'); |

Листинг 1.2 – выполнение задания лабораторной работы

Вывод представленного кода, рисунок 1.1.

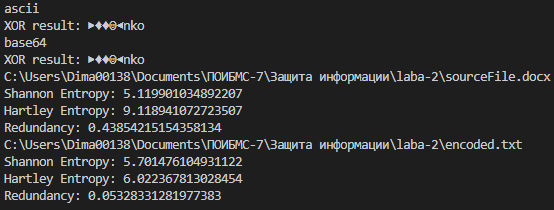


Рисунок 1.1

Процесс кодирования в base64 представляет группу из 24 последовательных битов в форме строки из 4 символов. Обработка выполняется слева направо, а 24-битная исходная группа образуется конкатенацией трех 8-битных групп. Данные 24 бита после этого трактуются как 4 сцепленных группы по 6 битов, каждая из которых транслируется в один символ алфавита base64.

**Вывод:** В ходе выполнения лабораторной работы был преобразован файл .docx в формат base64, была вычислена избыточность base64 и латинского алфавитов, используя энтропии Хартли и Шенона. Также был получен результат выполнения операции XOR над ASCII и base64 строками.