МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«**ИССЛЕДОВАНИЕ АСИММЕТРИЧНЫХ ШИФРОВ**»**

Студент: Бутурля Р.А.

ФИТ 3 курс 5 группа

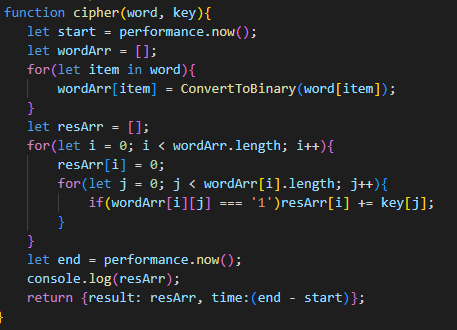
Вариант 3

Преподаватель: Савельева М. Г.

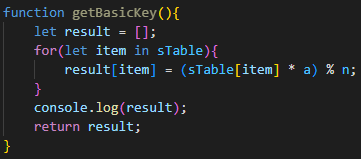
Минск 2023

1. Разработать авторские оконное приложения в соответствии с целью лабораторной работы. Приложение должно реализовывать следующие операции:
   * генерация сверхвозрастающей последовательности;
   * вычисление нормальной последовательности;
   * зашифрование сообщения, состоящего из собственных фамилии, имени и отчества;
   * расшифрование сообщения;
   * оценка времени выполнения операций зашифрования и расшифрования.

Для выполнения шифрования был разработан метод cipher(word, key). Параметр word – шифруемое слово, key – открытый ключ, генерируемый на основе закрытого методом getBasikKey().



Листинг 1 – Метод cipher(word, key)



Листинг 2 – Метод getBasilKey()

Для расшифроки сообщения был разработан метод decipher(word). На основе закрытого ключа и зашифрованного сообщения формируется последовательность 1 и 0 – битов зашифрованного символаю

|  |
| --- |
| function decipher(word){      let start = performance.now();      let wordArr = [];      let resultPer = [];      let result = [];      for(let i in word){          wordArr[i] = (word[i] \* a1) % n;      }      let arr = [];      for(let item in wordArr){          arr[item] = getArr(wordArr[item], sTable);      }      for(let i = 0; i < arr.length; i++){          resultPer[i] = [];          for(let j = 0; j < arr[i].length; j++){              resultPer[i][j] = sTable.indexOf(arr[i][j]);          }      }      for(let i = 0; i < resultPer.length; i++){          result[i] = ['0', '0', '0', '0', '0', '0', '0', '0'];          for(let j = 0; j < resultPer[i].length; j++){              result[i][resultPer[i][j]] = '1';          }          result[i] = result[i].join('');      }      for(let item in result){          result[item] = String.fromCharCode(parseInt(result[item], 2));      }      console.log(result);      let end = performance.now();      return {result: result.join(''), time:(end - start)};  } |

Листинг 3 – Метод decipher(word)

Метод decipher вызывает другой пользовательский метод – getArr() для получения правильной последовательности расположения битов в ресшифрованном сообщении.

|  |
| --- |
| function getArr(n, table){      for(let i = 0; i < table.length; i++){          if(table[i] == n) {console.log(table[i], 999);return;}      }      let y;      for(let i = 0; i < table.length; i++){          y=ju(n, [table[i]], table)          if(y && y.length!=0){              return y;          }      }      return false;  } |

Листинг 4 – Метод getArr()

Результаты выполнения шифрования и расшифрования представоены на рисунках 1 и 2 соответственно. Входное сообщение – “buturliaromanandreevich”.

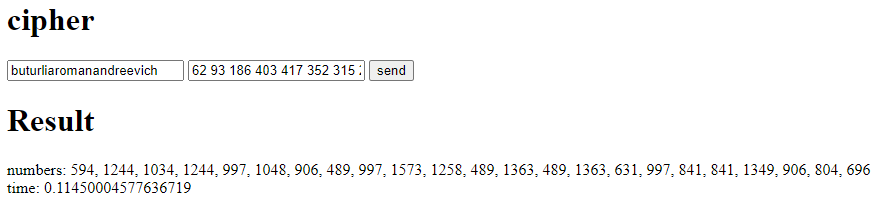


Рисунок 1 – Результат шифрования



Рисунок 2 – Результат расшифрования

1. Проанализировать время выполнения операций зашифрования/расшифрования при увеличении числа членов ключевой последовательности

Таблица 1 – Зависимость времени выполнения операция от количества членов ключевой последовательности

|  |  |  |  |
| --- | --- | --- | --- |
| Операция\кол-во членов ключевой последовательности | 8 | 10 | 12 |
| Шифрование | 0.09 | 0.53 | 0.26 |
| Расшифровка | 4.85 | 9.19 | 9.89 |

Рисунок 3 – График зависимости времени от количества членов в ключевой последовательности

**Вывод**

Применил практические навыки разработки и использования приложений для реализации асимметричных шифров.