Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №4»**

«Исследование криптографических шифров на основе подстановки (замены) символов»

**Выполнил:** студент 3 курса

4 группы специальности ПОИТ

Супрунюк Евгений Андреевич

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2023

**Выполнять зашифрование/расшифрование текстовых документов (объемом не менее 5 тысяч знаков**) **аффинным** **шифром** **Цезаря:**

Реализация на языке JavaScript:

|  |
| --- |
| const cipher = (char) => {  const index = alphabet.indexOf(char);  if (index === -1) { return char; }  const newIndex = (a \* index + b) % alphabet.length;  const newChar = alphabet[newIndex]; return newChar;  }; |

|  |
| --- |
| const decipher = (char) => {  const index = alphabet.indexOf(char);  if (index === -1) { return char; } let newIndex = index - b;  while (newIndex < 0) { newIndex += alphabet.length; }  newIndex = (newIndex \* modInverse(a, alphabet.length)) % alphabet.length;  const newChar = alphabet[newIndex]; return newChar;  }; |

**Выполнять зашифрование/расшифрование текстовых документов (объемом не менее 5 тысяч знаков**) **аффинным** **шифром** **Трисемуса:**

|  |
| --- |
| function encryptTrisemus(message) {  let encryptedMessage = '';  for (let i = 0; i < message.length; i++) {  let char = message[i];  let row = -1;  let col = -1;  for (let j = 0; j < trisemusTable.length; j++) {  for (let k = 0; k < trisemusTable[j].length; k++) {  if (trisemusTable[j][k] === char) {  row = j; col = k; break;  }  }  if (row !== -1 && col !== -1) { break; }  }  if (row === -1 || col === -1) {  encryptedMessage += char;  }  else {  let newRow = row;  let newCol = col;  if (newRow === 5) {  newRow = 0;  }  else {  newRow++;  }  encryptedMessage += trisemusTable[newRow][newCol];  } }  return encryptedMessage; } |

|  |
| --- |
| function encryptTrisemus(message) {  let encryptedMessage = '';  for (let i = 0; i < message.length; i++) {  let char = message[i];  let row = -1;  let col = -1;  for (let j = 0; j < trisemusTable.length; j++) {  for (let k = 0; k < trisemusTable[j].length; k++) {  if (trisemusTable[j][k] === char) {  row = j; col = k; break;  }  }  if (row !== -1 && col !== -1) { break; }  }  if (row === -1 || col === -1) {  encryptedMessage += char;  }  else {  let newRow = row;  let newCol = col;  if (newRow === 0) {  newRow = 5;  }  else {  newRow--;  }  encryptedMessage += trisemusTable[newRow][newCol];  } }  return encryptedMessage; } |

**Результат работы программы, выполняющей шифрование на основе аффинной системы подстановок Цезаря:**

|  |
| --- |
| Encrypt Caesar: 11.707ms  Successfully ciphered data and saved to outputCaesar.txt |

**Результат работы программы, выполняющей расщифрование на основе аффинной системы подстановок Цезаря:**

|  |
| --- |
| Decrypt Caesar: 9.042ms  Successfully deciphered data and saved to decryptCaesar.txt |

**Результат работы программы, выполняющей шифрование на основе таблицы Трисемуса:**

|  |
| --- |
| Encrypt Trisemus: 19.145ms  Successfully ciphered data and saved to outputTrisemus.txt |

**Результат работы программы, выполняющей расщифрование на основе таблицы Трисемуса:**

|  |
| --- |
| Decrypt Trisemus: 11.281ms  Successfully ciphered data and saved to decryptTrisemus.txt |