Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Практическая работа №4

«Исследование ассиметричных алгоритмов шифрования. Создание ключей PGP, передача подписанных и защищенных сообщений»

|  |  |
| --- | --- |
| Выполнил: ст. гр. 910901 | Нехайчик И. И. |
| Проверил: | Давыдович К. И. |

Минск 2022

**Задание**

Написать программу на любом языке программирования:

1. Перевидите число 343 в двоичную систему счисления.
2. Пусть каждая из 16 первых букв английского алфавита (ABCDEFGKLMNOPQRS) имеет четырёхразрядный двоичный код, соответствующий её номеру от 0 до 15, т.е.

A – 00002;

B – 00012; …;

S – 11112.

1. Составить из этих букв произвольное сообщение, состоящее из 32 символов.
2. Разбить полученное сообщение на блоки длиной 64 бита.
3. Значения полученных блоков записать в десятичной системе счисления.
4. Найти состояние 28-разрядного двоичного регистра сдвига после циклического сдвига влево на 5, числа 17911733310 (вар. 14), предварительно записанного в регистр.
5. Найти сумму по модулю 2 двух чисел 224489930110 и 17911733310 (вар. 14).

**Листинг кода**

const DECIMAL\_NUMBER = 3n \*\* 43n;

const OPEN\_TEXT = 'bad eagle feel mama fill cake pink milk';

const REGISTR\_NUMBER = 179117333n;

const SHIFT\_LEFT = 5;

function convertDecimalBigIntToBinary(number) {

  if (number > 0) {

    return convertDecimalBigIntToBinary(number / 2n) + (number % 2n);

  }

  return '';

}

function convertDecimalToBinary(number) {

  if (number > 0) {

    return convertDecimalToBinary(parseInt(number / 2)) + (number % 2);

  }

  return '';

}

function convertBinaryToDecimalBigInt(number) {

  result = 0n;

  for (let i = 0; i < number.length; i++) {

    result += BigInt(number[i]) \* 2n \*\* BigInt(number.length - 1 - i);

  }

  return result;

}

function task2(openText) {

  const map = new Map();

  let startCodeChar = 65;

  const endCodeChar = 80;

  let count = 0;

  for (let i = startCodeChar; i <= endCodeChar; i++) {

    let binary = convertDecimalToBinary(count);

    count++;

    if (binary.length < 4) {

      while (binary.length !== 4) {

        binary = 0 + binary;

      }

    }

    map.set(i, binary);

  }

  console.log(map, '\n');

  console.log(openText + '\n');

  let str = openText.toUpperCase().split(' ').join('');

  let binaryStr = '';

  for (let i = 0; i < str.length; i++) {

    binaryStr += map.get(str.charCodeAt(i));

  }

  let binaryResult = '';

  for (let i = 0; i < parseInt(binaryStr.length / 64); i++) {

    binaryResult +=

      binaryStr

        .split('')

        .slice(i \* 64, i \* 64 + 64)

        .join('') + ' ';

  }

  binaryResult = binaryResult.trim().split(' ');

  console.log(binaryResult, '\n');

  let result = '';

  for (let i = 0; i < binaryResult.length; i++) {

    result += convertBinaryToDecimalBigInt(binaryResult[i]) + ' ';

  }

  return result.trim();

}

function task3(number, shiftLeft) {

  const binaryNumber = convertDecimalBigIntToBinary(number).split('');

  for (let i = 0; i < shiftLeft; i++) {

    const popBit = binaryNumber.pop();

    binaryNumber.unshift(popBit);

  }

  console.log(`Input decimal number: ${number}`);

  console.log(

    `Converting input number to binary: ${convertDecimalBigIntToBinary(

      number,

    )}`,

  );

  return binaryNumber.join('');

}

function task4(number) {

  const number1 = 2244899301n;

  let binaryNumber1 = convertDecimalBigIntToBinary(number1);

  let binaryNumber = convertDecimalBigIntToBinary(number);

  let result = '';

  if (binaryNumber.length !== binaryNumber1.length) {

    if (binaryNumber.length > binaryNumber1.length) {

      const binaryNumber1Arr = binaryNumber1.split('');

      while (binaryNumber1Arr.length !== binaryNumber.length) {

        binaryNumber1Arr.unshift('0');

      }

      binaryNumber1 = binaryNumber1Arr.join('');

    } else {

      const binaryNumberArr = binaryNumber.split('');

      while (binaryNumberArr.length !== binaryNumber1.length) {

        binaryNumberArr.unshift('0');

      }

      binaryNumber = binaryNumberArr.join('');

    }

  }

  for (let i = 0; i < binaryNumber.length; i++) {

    if (binaryNumber[i] === binaryNumber1[i]) {

      result += '0';

    } else {

      result += '1';

    }

  }

  console.log(`Number 1: ${binaryNumber}`);

  console.log(`Number 2: ${binaryNumber1}`);

  return result;

}

console.log('--- DECIMAL TO BINARY ---\n');

console.log(`Decimal number (3^43): ${DECIMAL\_NUMBER}`);

console.log(`Binary number: ${convertDecimalBigIntToBinary(DECIMAL\_NUMBER)}`);

console.log('\n--- TASK 2 ---\n');

console.log(`Result decimal numbers: ${task2(OPEN\_TEXT)}`);

console.log('\n--- TASK 3 (14) ---\n');

console.log(`Result: ${task3(REGISTR\_NUMBER, SHIFT\_LEFT)}`);

console.log('\n--- TASK 4 (14) ---\n');

console.log(`Result: ${task4(REGISTR\_NUMBER)}`);

**Вывод в консоли**

--- DECIMAL TO BINARY ---

Decimal number (3^43): 328256967394537077627

Binary number: 100011100101101111011000001001010101001010110010111010111101101111011

--- TASK 2 ---

Map(16) {

  65 => '0000',

  66 => '0001',

  67 => '0010',

  68 => '0011',

  69 => '0100',

  70 => '0101',

  71 => '0110',

  72 => '0111',

  73 => '1000',

  74 => '1001',

  75 => '1010',

  76 => '1011',

  77 => '1100',

  78 => '1101',

  79 => '1110',

  80 => '1111'

}

bad eagle feel mama fill cake pink milk

[

  '0001000000110100000001101011010001010100010010111100000011000000',

  '0101100010111011001000001010010011111000110110101100100010111010'

]

Result decimal numbers: 1167565574973931712 6393739988904364218

--- TASK 3 (14) ---

Input decimal number: 179117333

Converting input number to binary: 1010101011010001110100010101

Result: 1010110101010110100011101000

--- TASK 4 (14) ---

Number 1: 00001010101011010001110100010101

Number 2: 10000101110011100111000111100101

Result: 10001111011000110110110011110000