Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра инженерной психологии и эргономики

Практическая работа №3

«Шифрование, дешифрование информации с применением криптографических алгоритмов гаммирования»

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

Минск 2022

**Задание**

Написать программу генерации шифра для заданных *a* и *s* по формуле: *Ci = (a Pi + s) mod N*, где *P* – порядковый номер символа открытого текста *(0≤ Pi ≤ N-1)*; *C* – порядковый номер символа зашифрованного текста *(0≤ Ci ≤ N-1)*; *N* – размер алфавита; *a* – десятичный коэффициент; *s* – коэффициент сдвига.

Написать программы шифровки и дешифровки для метода моноалфавитной подстановки по заданному шифру.

**Листинг кода**

const OPEN\_TEXT = 'Nekhaychik Irina Igorevna';

const ALPHABETH\_SIZE = 26;

const A = 3.13; // A и ALPHABET\_SIZE не должны иметь общих делителей

const S = 7;

const START\_ASCII\_CODE = 65;

const END\_ASCII\_CODE = 90;

const SPACE\_ASCII\_CODE = 32;

function getByValue(map, searchValue) {

  for (let [key, value] of map.entries()) {

    if (value === searchValue) {

      return key;

    }

  }

}

function createCipherAlphabet() {

  const cipherAlphabet = new Map();

  for (let P = START\_ASCII\_CODE; P <= END\_ASCII\_CODE; P++) {

    let value =

      Math.round((A \* P + S) % ALPHABETH\_SIZE) + (START\_ASCII\_CODE - 1);

    cipherAlphabet.set(P, value);

  }

  return cipherAlphabet;

}

// ASCII - 65-90, SPACE-->32

function cryptWithXORCipher(str) {

  const openText = str.toUpperCase();

  const cipherAlphabet = createCipherAlphabet();

  let cipherText = '';

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

    const openCharCode = openText.charCodeAt(i);

    let cipherCharCode = openCharCode;

    if (openCharCode !== SPACE\_ASCII\_CODE) {

      cipherCharCode = cipherAlphabet.get(openCharCode);

    }

    cipherText += String.fromCharCode(cipherCharCode);

  }

  return cipherText;

}

function encryptXORCipher(str) {

  const cipherAlphabet = createCipherAlphabet();

  let encryptedText = '';

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

    const cipherCharCode = str.charCodeAt(i);

    let encryptedCharCode = cipherCharCode;

    if (cipherCharCode !== SPACE\_ASCII\_CODE) {

      encryptedCharCode = getByValue(cipherAlphabet, cipherCharCode);

    }

    encryptedText += String.fromCharCode(encryptedCharCode);

  }

  return encryptedText;

}

console.log('Cipher alphabet through char code: ');

console.log(createCipherAlphabet());

console.log('\n\n--- CRYPT ---\n');

console.log(`Open text: ${OPEN\_TEXT}`);

console.log(`Crypted text: ${cryptWithXORCipher(OPEN\_TEXT)}`);

console.log('\n\n--- ENCRYPT ---\n');

console.log(`Crypted text: ${cryptWithXORCipher(OPEN\_TEXT)}`);

console.log(

  `Encrypted text: ${encryptXORCipher(cryptWithXORCipher(OPEN\_TEXT))}`,

);

**Вывод в консоли**

Cipher alphabet through char code:

Map(26) {

  65 => 66,

  66 => 70,

  67 => 73,

  68 => 76,

  69 => 79,

  70 => 82,

  71 => 85,

  72 => 88,

  73 => 65,

  74 => 69,

  75 => 72,

  76 => 75,

  77 => 78,

  78 => 81,

  79 => 84,

  80 => 87,

  81 => 65,

  82 => 68,

  83 => 71,

  84 => 74,

  85 => 77,

  86 => 80,

  87 => 83,

  88 => 86,

  89 => 90,

  90 => 67

}

--- CRYPT ---

Open text: Nekhaychik Irina Igorevna

Crypted text: QOHXBZIXAH ADAQB AUTDOPQB

--- ENCRYPT ---

Crypted text: QOHXBZIXAH ADAQB AUTDOPQB

Encrypted text: NEKHAYCHIK IRINA IGOREVNA