Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования
Кафедра инженерной психологии и эргономики

КРИПТОГРАФИЧЕСКИЕ ТЕХНОЛОГИИ

Практическая работа №3

Шифрование, дешифрование информации с применением криптографических алгоритмов гаммирования

Выполнил:
Глик А. Г.

Проверил:
Давыдович К.И.

Минск 2023

Цель работы:

Основная часть занятия состоит в закреплении теоретического материала курса, приобретении навыков выполнения задач по основам криптографических технологий, анализа результатов, грамотного оформления отчетов, в частности: шифрование, дешифрование информации с применением криптографических алгоритмов гаммирования, примеры шифрования и дешифрования.

Ход работы:

В ходе работы были изучены:

1. Гаммирование, как метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст.
2. Доказательства абсолютной стойкости гаммирования, описание Шенноном
3. Требования к гамме

Задание

Написать программу генерации шифра для заданных , a и s по формуле:

$$Ci = (a\ Pi + s) \bmod N,$$

где

P – порядковый номер символа открытого текста ($0 \leq Pi \leq N1$);

C – порядковый номер символа зашифрованного текста ($0 \leq Ci \leq N - 1$);

N – размер алфавита;

a – десятичный коэффициент;

s – коэффициент сдвига.

Напишите программы шифровки и расшифровки для метода моноалфавитной подстановки по заданному шифру, подходящую для работы с русским языком.

Вывод:

В ходе работы произошло ознакомление с методическим материалом и дополнительной информацией, связанной с предоставленной темой; было разработано приложение, полностью покрывающее задачу, изложенную в вышепредставленных требованиях задания для практического занятия. В приложениях А и Б предоставлены код программы и результат ее работы соответственно.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг кода программы

```
function encrypt(text, a, s, N) {
  let encryptedText = '';

  for (let i = 0; i < text.length; i++) {
    if (text[i] === ' ') {
      encryptedText += ' ';
    } else {
      const Pi = getRussianAlphabetIndex(text[i]);
      const Ci = (a * Pi + s) % N;
      const encryptedChar = getRussianAlphabetChar(Ci);
      encryptedText += encryptedChar;
    }
  }

  return encryptedText;
}

function decrypt(encryptedText, a, s, N) {
  let decryptedText = '';

  for (let i = 0; i < encryptedText.length; i++) {
    if (encryptedText[i] === ' ') {
      decryptedText += ' ';
    } else {
      const Ci = getRussianAlphabetIndex(encryptedText[i]);
      const aInverse = findModularInverse(a, N);
      const Pi = (aInverse * (Ci - s + N)) % N;
      const decryptedChar = getRussianAlphabetChar(Pi);
      decryptedText += decryptedChar;
    }
  }

  return decryptedText;
}
```

```
function findModularInverse(a, N) {
  for (let x = 1; x < N; x++) {
    if ((a * x) % N === 1) {
      return x;
    }
  }
  return null;
}

function getRussianAlphabetIndex(char) {
  const alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
  return alphabet.indexOf(char);
}

function getRussianAlphabetChar(index) {
  const alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ ";
  return alphabet[index];
}

const a = 7;
const s = 3;
const N = 33;

const plaintext = "великий и могучий русский язык".toUpperCase();

const encryptedText = encrypt(plaintext, a, s, N);
console.log("Зашифрованный текст:", encryptedText);

const decryptedText = decrypt(encryptedText, a, s, N);
console.log("Расшифрованный текст:", decryptedText);
```

ПРИЛОЖЕНИЕ Б

(обязательное)

Результаты работы программы

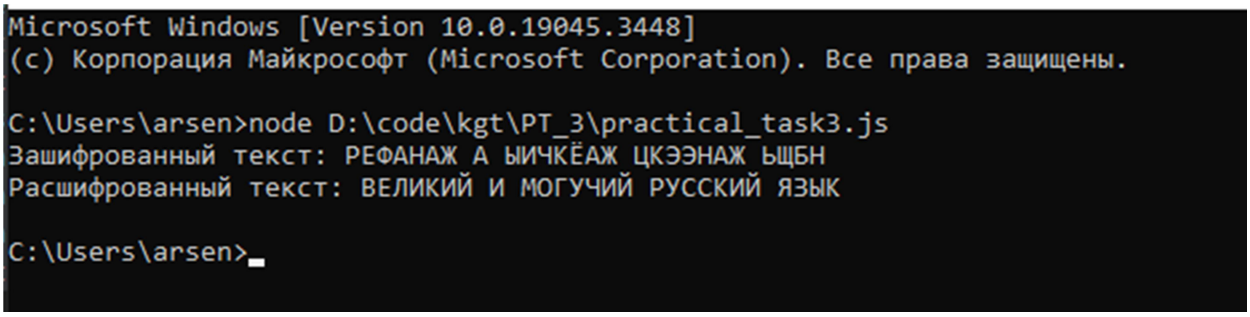

Рисунок Б – Результат работы приложение