Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе 8»**

“ИССЛЕДОВАНИЕ ПОТОКОВЫХ ШИФРОВ”

**Выполнила:** студентка 3 курса

4 группы специальности ПОИТ

Сапегина Екатерина Игоревна,

Вариант 11 (1)

Минск 2023

***Цель:*** изучение и приобретение практических навыков разработки и использования приложений для реализации потоковых

шифров.

***Задачи:***

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости потоковых шифров.

2. Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для потокового зашифрования/расшифрования.

3. Выполнить анализ криптостойкости потоковых шифров.

4. Оценить скорость зашифрования/расшифрования реализованных шифров.

5. Результаты выполнения лабораторной работы оформить в

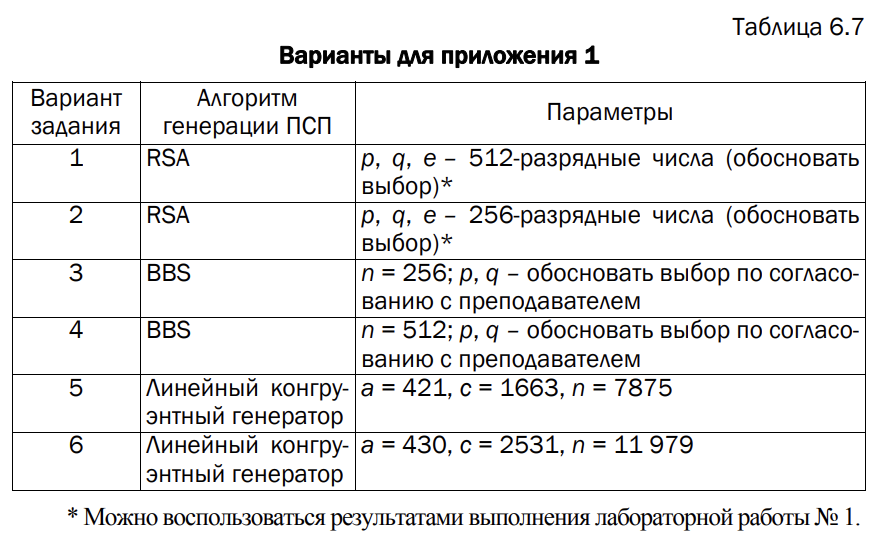
виде описания разработанного приложения, методики выполнения

экспериментов с использованием приложения и результатов эксперимента.

***Практическая часть:***

1. Разработать авторские многооконные приложения в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими заданные алгоритмы.

Приложение 1 должно реализовывать генерацию ПСП в соответствии с вариантом из таблицы 6.7. (Вариант 1)



Приложение 2 должно реализовывать алгоритм RC4 в соответствии с вариантом из табл. 6.8 (Вариант 11), а также дополнительно выполнять оценку скорости выполнения операций генерации ПСП.



В качестве шифруемого сообщения может быть выбран произвольный текст.

Реализация:

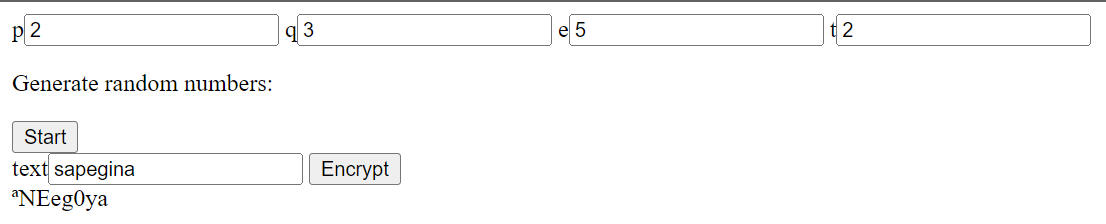
***rsa.js***

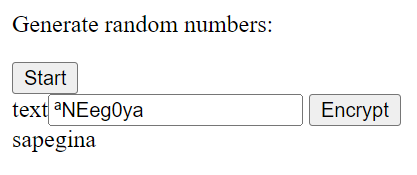
|  |
| --- |
| const p = 13, q = 53;  const e = 23;  const CalculateNOD = (x, y) => {  console.log(x,y);  while (x !== 0 && y !== 0) {  if (x > y) {  x = x % y;  console.log(`x:${x}`);  }  else {  y = y % x;  console.log(`y:${y}`);  }  }  console.log(`x:${x} y:${y}`);  return Math.max(x, y);  }  const IsSimple = (x) => {  if (x === 1)  return true;  for (let i = 2; i <= Math.floor(Math.sqrt(x)); i++)  if (x % i === 0)  return false;  return true;  }  const generateNewNumber = (xPrevious, e, n) => {  console.log('xPrevious ' + xPrevious)  console.log('e ' + e)  console.log('n ' + n)  const a = (xPrevious\*\*e);  console.log('a ' + a);  console.log('a % n ' + a);  return a % n;  }  const generateRandomNumbersRSA = () => {  const p = document.getElementById("p").value.toLowerCase();  const q = document.getElementById("q").value.toLowerCase();  const e = document.getElementById("e").value.toLowerCase();  const t = document.getElementById("t").value.toLowerCase();  const x0 = 19;  const result = document.querySelector('#result');  const n = p \* q;  const fi = (p - 1) \* (q - 1);  if(!IsSimple(p) || !IsSimple(q)){  result.innerHTML = `Info: p and q must be simple p=${p} q=${q}`;  return ;  }  if (CalculateNOD(e, fi) !== 1) {  console.log('HERE');  result.innerHTML = `Info: Number e = ${e} is not mutually simple with Fi(n) = (p-1)\*(q-1) = ${fi}`;  return ;  }  const randomNumbers = [];  randomNumbers.push(generateNewNumber(x0,e,n))  for(let i = 0; i < t; i++){  randomNumbers.push(generateNewNumber(randomNumbers[i],e,n))  }  console.log(randomNumbers);  result.innerHTML = "";  result.innerHTML = randomNumbers.join(' ');  }  function rc4() {  const str = document.getElementById("str").value;  const result = document.querySelector('#result');  console.time('rc4')  const key = [121,14,89,15];  n = 8;  let s = [], j = 0, res = [], resStr = "";  for (let i = 0; i < 2\*\*n; i++) {  s[i] = i;  }  for (let i = 0; i < 2\*\*n; i++) {  j = (j + s[i] + key[i % key.length]) % 2\*\*n;  [s[i],s[j]] = [s[j],s[i]]  }  console.log(s);  let i = 0;  j = 0;  for (let y = 0; y < str.length; y++) {  i = (i + 1) % 2\*\*n;  j = (j + s[i]) % 2\*\*n;  [s[i],s[j]] = [s[j],s[i]]  res.push(s[(s[i] + s[j]) % 2\*\*n]);  console.log(res);  resStr += String.fromCharCode(str.charCodeAt(y) ^ s[(s[i] + s[j]) % 512]);  }  console.timeEnd('rc4')  result.innerHTML = resStr;  return resStr;  } |

***rsa.html***

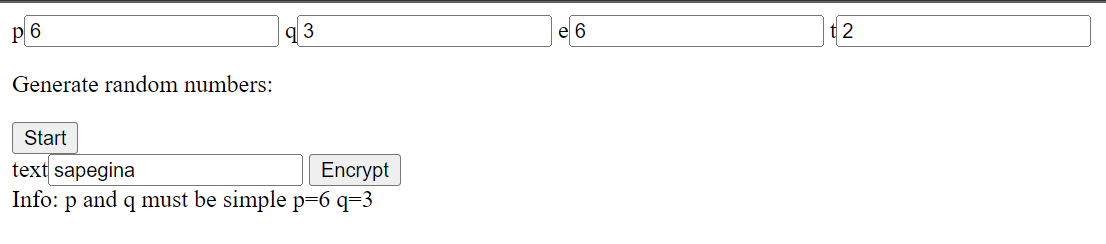
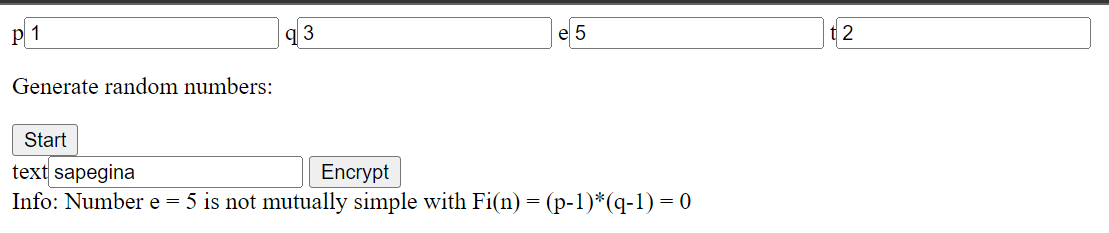
|  |
| --- |
| <html>  <head>  </head>  <body>  <label for="p">p</label><input value="" type="text" id="p">  <label for="q">q</label><input value="" type="text" id="q">  <label for="e">e</label><input value="" type="text" id="e">  <label for="t">t</label><input value="" type="text" id="t">  <p>Generate random numbers:</p>  <button onclick="generateRandomNumbersRSA()">Start</button>  <br>  <label for="str">text</label><input value="1" type="text" id="str">  <button onclick="rc4()">Encrypt</button>  <div id="result"></div>  <script src="rsa.js"></script>  </body>  </html> |

***Результат:***

******

******

***Обоснование выбора значений в коде:***

******