МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«**ИССЛЕДОВАНИЕ АСИММЕТРИЧНЫХ ШИФРОВ**»**

Студент: Бутурля Р.А.

ФИТ 3 курс 5 группа

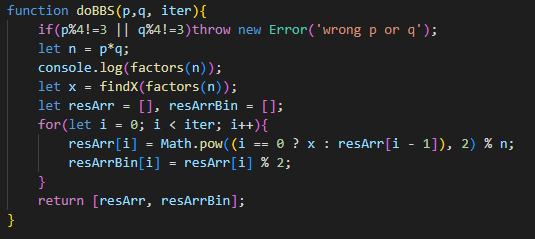
Вариант 3

Преподаватель: Савельева М. Г.

Минск 2023

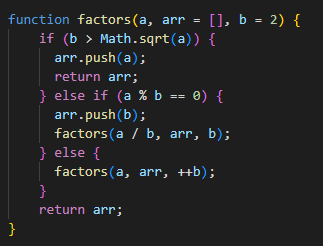
1. Разработать авторские многооконные приложения в соответствии с целью лабораторной работы. При этом можно воспользоваться готовыми библиотеками либо программными кодами, реализующими заданные алгоритмы.
2. Приложение 1 должно реализовывать генерацию ПСП(BBS, 256)

Для выполнения генерации ПСП был разработан метод doBBS(p, q, iter). Параметры p и q – простые числа, сравнимые с числом 3 по модулю 4, параметр iter – количество символов, которые должен вернуть нам генератор.



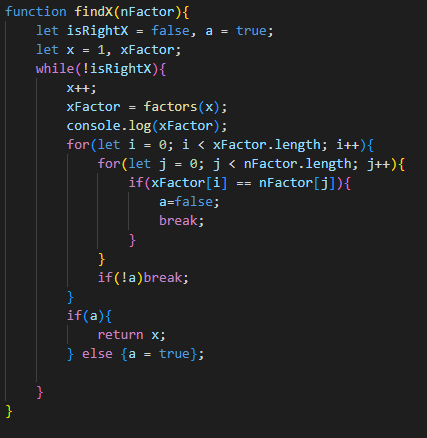
Листинг 1 – Метод doBBS(p, q, iter)

Метод doBBS(p, q, iter) вызывает во время выполнения метод factors(n). Метод factors(n) расклалывает поступившее в качестве параметра n число на простые множители и возвращает их.



Листинг 2 – Метод factors(n)

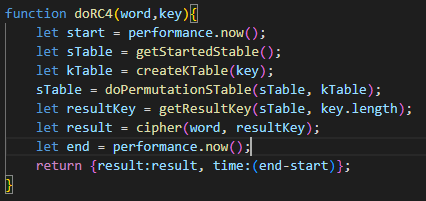
Также метод doBBS(p, q, iter) вызывает метод findX(nFactor). Данный метод определяет некоторое значение x, которое должно быть взаимно простым с числом n. В параметрe nFactor хранится разложение n на простые числа и на основе этого параметра находится x.



Листинг 3 – findX(nFactor)

1. Приложение 2 должно реализовывать алгоритм RC4

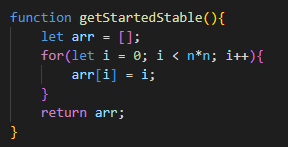
Для реализации алгоритма RC4 был разработан метод doRC4(word, key). Параметр word – шифруемое слово, key – ключ, состоящий из некоторого количества чисел.



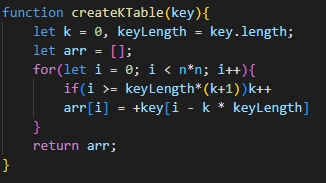
Листинг 4 – Метод doRC4(word, key)

Метод doRC4(word, key) поочередно вызывает следующие методы:

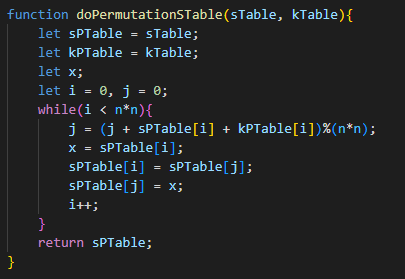
* getStartedStable() – генерация стартовой таблицы S;
* createKTable(key) – генерирует таблицу K, записывая в нее последовательность ключей необходимое количество раз;
* doPermutationSTable(sTable, kTable) и getResultKey(sTable, length) – на основе таблиц, полученных из методов пункта 1 и 2, методы подготавливают таблицу S к основному этапу и выполняют его;
* cipher(word, key) – метод выполняет конечное преобразования слова с помощью битовой операции.



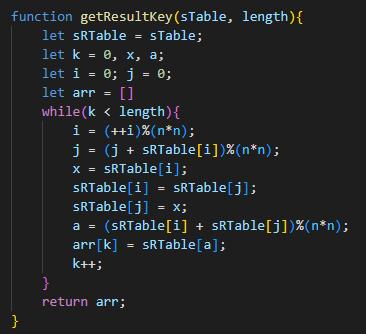
Листинг 5 – Метод getStartedStable()



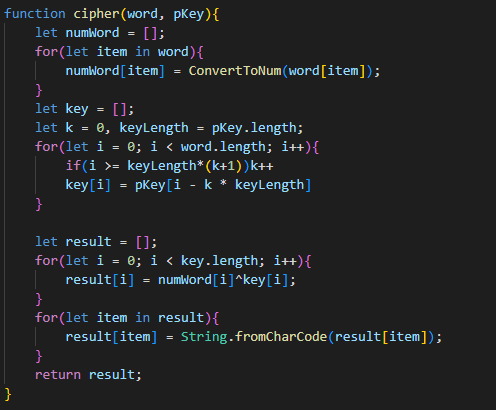
Листинг 6 – Метод createKTable(key)



Листинг 7 – Метод doPermutationSTable(sTable, kTable)



Листинг 8 – Метод getResultKey(sTable, length)



Листинг 9 – Метод cipher(word, pKey)

Для написания оконного приложения был выбран JavaScript(Node.js).

|  |
| --- |
| const express = require('express');  const {doBBS} = require('./bbs');  const {doRC4} = require('./rc4');  const fs = require('fs');  const app = express();  app.use(express.json());  app.get('/', (req, res)=>{      res.end(fs.readFileSync('first.html'));  });  app.post('/',(req, res)=>{      console.log(req.body);      let {p,q} = req.body;      console.log(p,q);      let arr = doBBS(+p,+q,10);      res.end(JSON.stringify(arr));  });  app.get('/rc4', (req, res)=>{      res.end(fs.readFileSync('second.html'));  });  app.post('/rc4', (req, res)=>{      let {word,key} = req.body;      key = key.split(' ');      console.log(word, key);      let obj = doRC4(word,key);      res.end(JSON.stringify(obj));  })  app.listen(3000); |

Листинг 10 – Приложение для демонстрации работы потоковых шифров

Результаты выполнения приложения:

* BBS – входные данные: 11 и 19. Результат выполнения представлен на рисунке 1.
* RC4 – входные данные: “roman” и “1 2 3”. Результат выполнения представлен на рисунке 1.

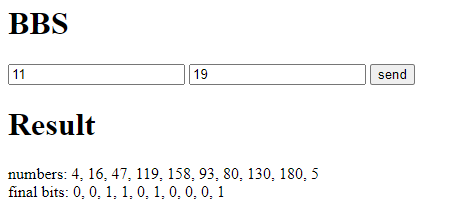


Рисунок 1 – Результат работы BBS

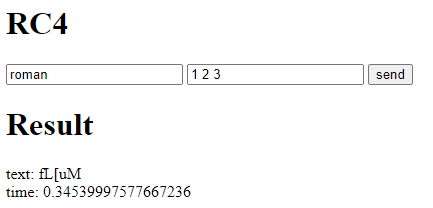


Рисунок 2 – Результат работы RC4

**Вывод**

Применил практические навыки разработки и использования приложений для реализации потоковых шифров.