МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил\_\_:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Брест – 2023

**Ход работы**

***Цель работы:*** *изучить алгоритмы обмена ключами. Практически реализовать алгоритмы обмена ключами.*

***Вариант:*** *RC2 + EKE на основе RSA*

***Код программы: Алгоритм RC2***

class RC4 {

shuffle(key) {

const s = Array.from({ length: 256 }, (\_, index) => index);

let j = 0;

const l = key.length;

for (let i = 0; i < 256; i++) {

j = (j + s[i] + key[i % l]) % 256;

[s[i], s[j]] = [s[j], s[i]];

}

return s;

}

\*pseudoRandomByte(s) {

let i = 0;

let j = 0;

while (true) {

i = (i + 1) % 256;

j = (j + s[i]) % 256;

[s[i], s[j]] = [s[j], s[i]];

const t = (s[i] + s[j]) % 256;

yield s[t];

}

}

crypt(str, key) {

key = String(key).split('').map((item) => +item);

const s = this.shuffle(key);

const message = str.split('').map((item) => item.charCodeAt(0));

const rand = this.pseudoRandomByte(s);

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

message[i] ^= rand.next().value;

}

return message.map((item) => String.fromCharCode(item)).join('');

}

}

***Код программы: Алгоритм RSA***

class RSA {

constructor() {

this.p1 = this.getPrime(this.getRamdomInt(0, 100));

this.p2 = this.getPrime(this.getRamdomInt(0, 100));

this.e = this.getPrime(this.getRamdomInt(0, 3));

this.n = this.p1 \* this.p2;

this.f = (this.p1 - 1) \* (this.p2 - 1);

this.k = 2;

this.d = (this.k \* this.f + 1) / this.e;

}

getRamdomInt(a, b) {

return Math.round(a + Math.random() \* (b - a));

}

getPrime(n) {

const primes = [

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37,

41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89,

97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151,

157, 163, 167, 173, 179, 181, 191, 193, 197, 199

];

return primes[n % primes.length];

}

getPublicKey() {

return { n: this.n, e: this.e };

}

encrypt(message, n, e) {

return (+message) \*\* e % n + '';

}

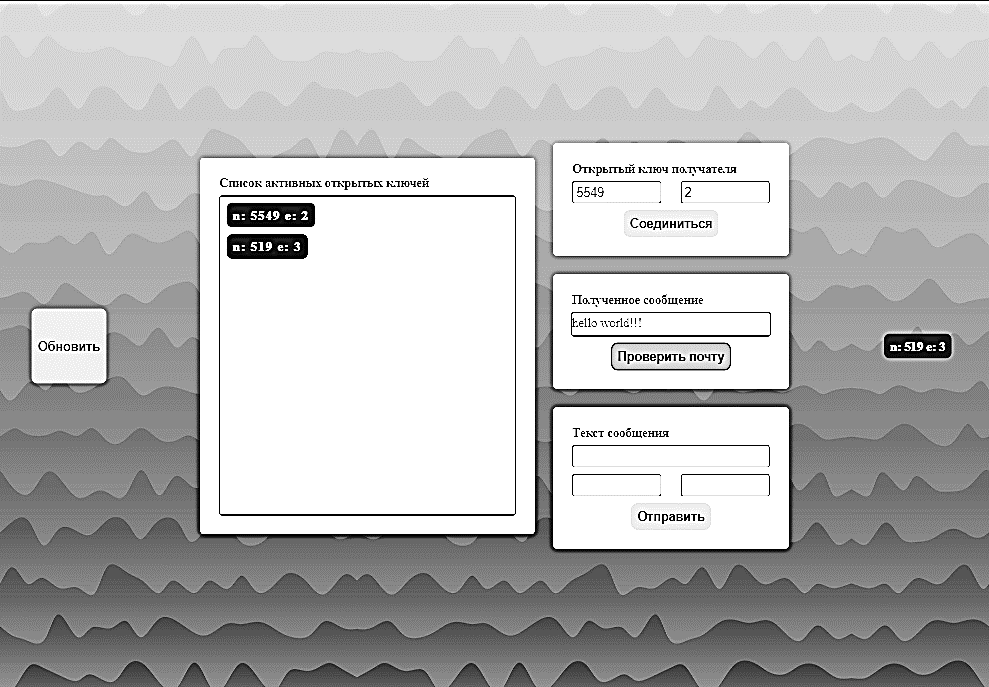
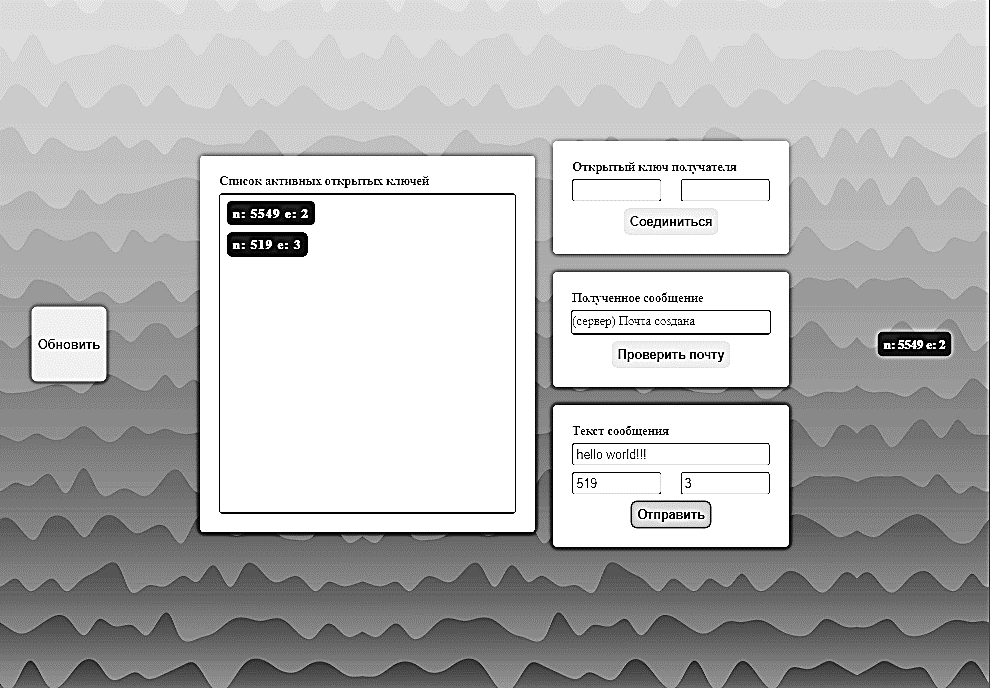
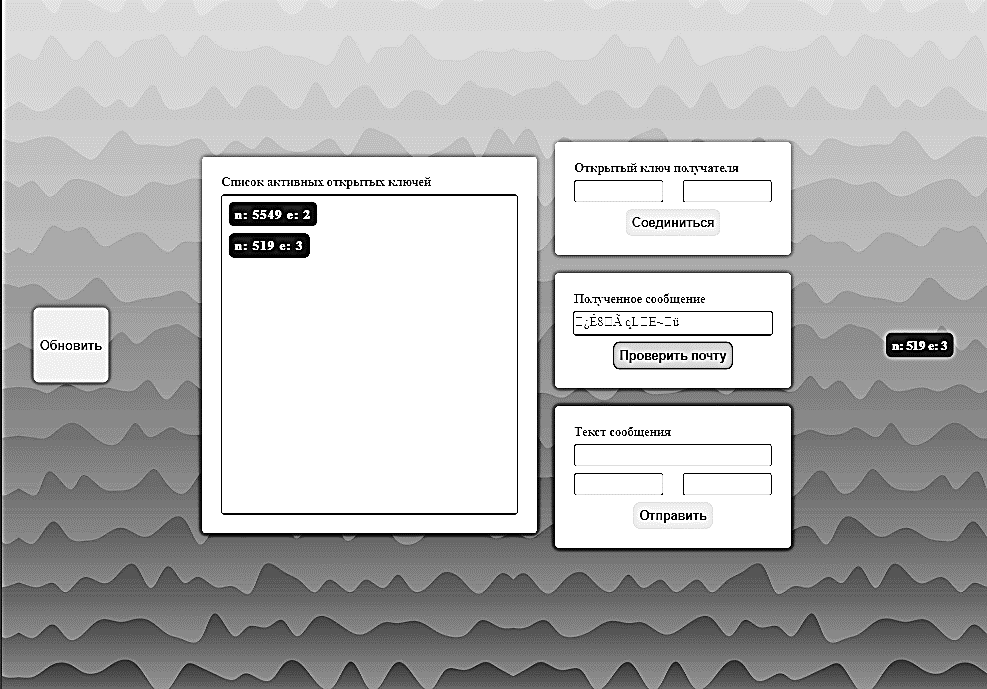
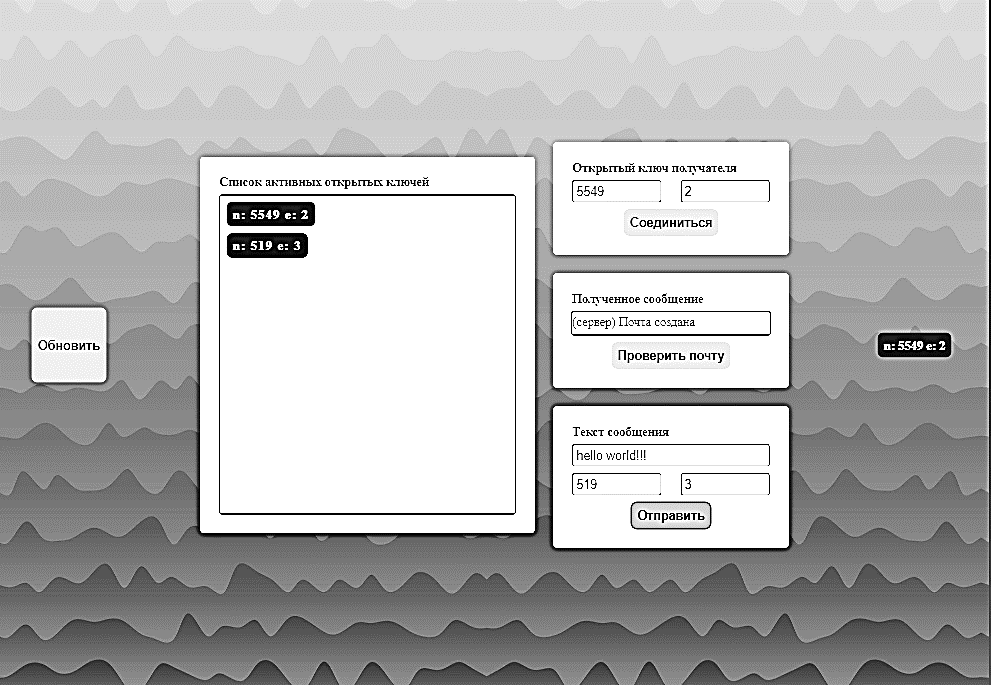
decrypt(message) {

return (+message) \*\* this.d % this.n + '';

}

}

***Результаты работы программы:***

******