Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет**

«Исследование криптографических алгоритмов

на основе эллиптических кривых»

Выполнил:

студент 3 курса 4 группы

специальности ПОИТ

Чернявский А. Л.

Минск 2020

**Цель**: изучение и приобретение практических навыков разработки и использования приложений для реализации криптографических алгоритмов на основе эллиптических кривых (содержит 3 самостоятельных задания, каждое из которых рассчитано на 2 часа аудиторных занятий).

**Задачи**:

1. Закрепить теоретические знания по алгебраическому описанию и геометрическому представлению операций над эллиптическими кривыми (ЭК):

- по алгоритмам согласования ключевой информации на основе ЭК,

- алгоритмам зашифрования/расшифрования информации на основе

асимметричной криптонафии и ЭК,

- алгоритмам генерации и верификации электронной цифровой подписи

на основе асимметричной криптографии и ЭК,

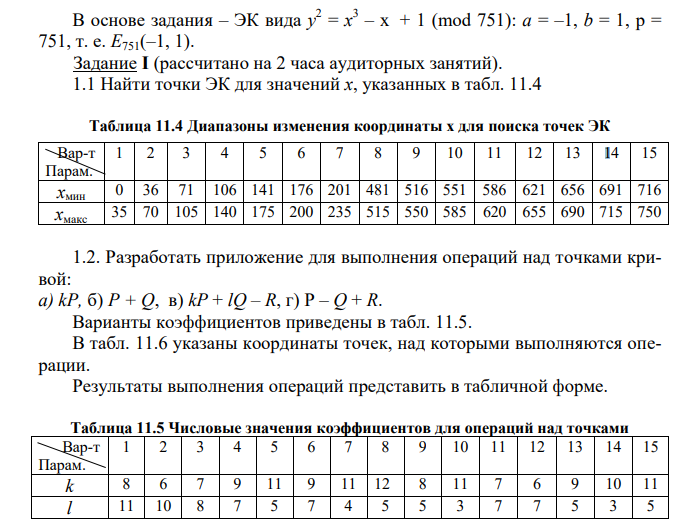
- оценке криптостойкости систем на основе ЭК.

2. Разработать приложение для реализации указанных преподавателем

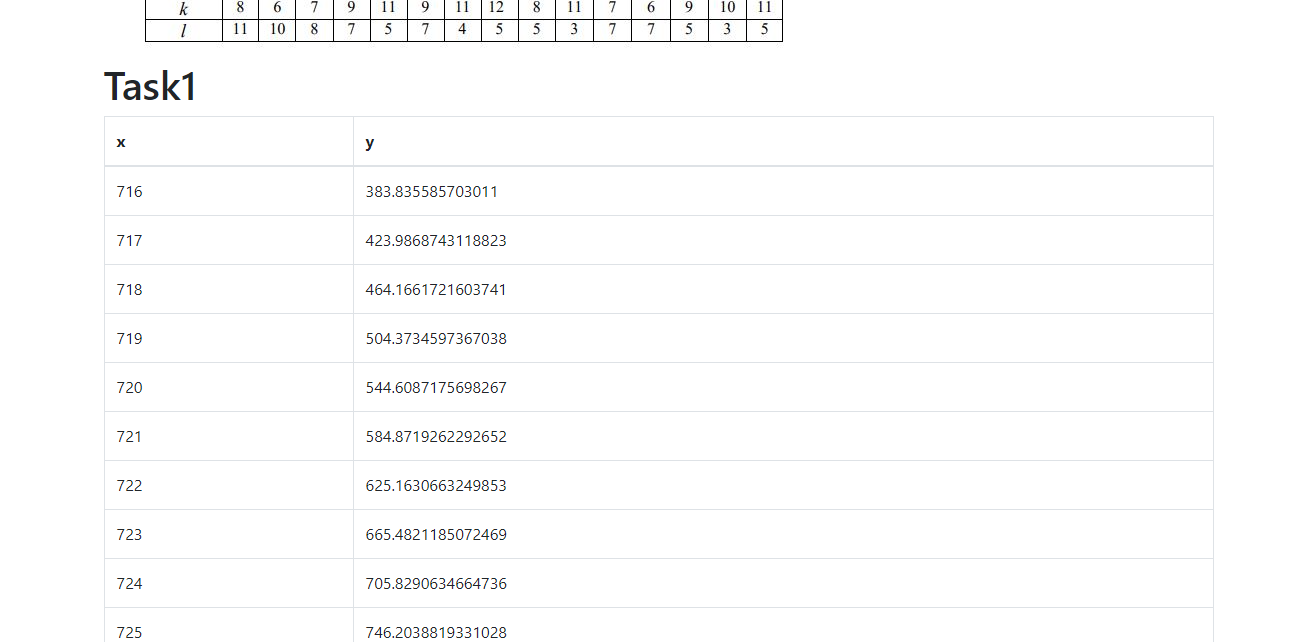
методов криптопреобразования на основе ЭК.

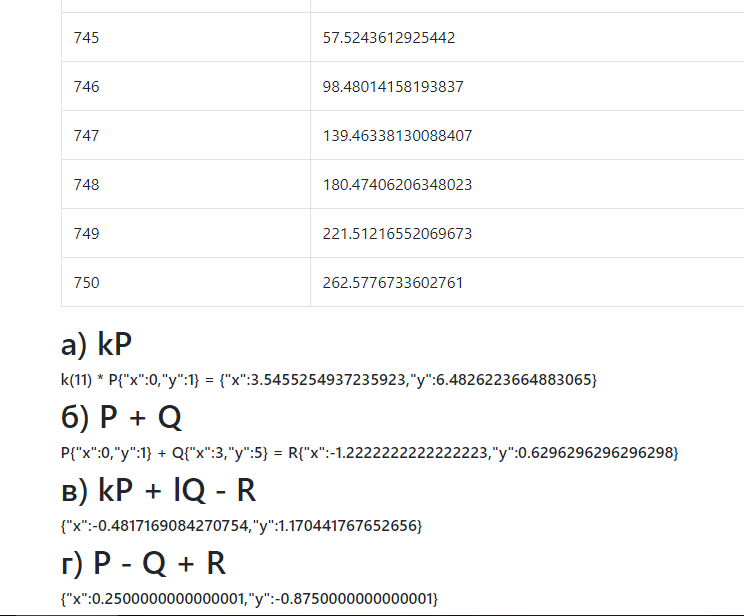
3. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

**Практическое задание**:



**Результат выполнения работы приложения:**





**Листинги**

import React from 'react'

export const Task1 = ({ xMin, xMax, p, a, b, k, l }) => {

    const xArr = []

    const P = { x: 0, y: 1 }

    const Q = { x: 3, y: 5 }

    for (let i = xMin; i <= xMax; i++) {

        xArr.push(i)

    }

    let points = []

    // y^2 = x^3 - x + 1 (mod 751)

    xArr.forEach(x => {

        points.push({

            x,

            y: ((x \*\* 3 - x + 1) \*\* 0.5) % p

        })

    })

    const inverse = (P) => {

        return { x: P.x, y: -P.y }

    }

    const lambda = (P, Q) => {

        if (P.x === Q.x && P.y === Q.y) {

            return (3 \* P.x \* P.x + a) / (2 \* P.y)

        }

        return (Q.y - P.y) / (Q.x - P.x)

    }

    const sum = (P, Q) => {

        const x = lambda(P, Q) \*\* 2 - P.x - Q.x

        const y = lambda(P, Q) \* (P.x - x) - P.y

        return { x, y }

    }

    const mul = (k, P) => {

        let result = { ...P }

        for (let i = 0; i < k; i++) {

            // console.log(result)

            result = sum(result, P)

        }

        return result

    }

    return (

        <div>

            <h1>Task1</h1>

            <table className="table table-bordered">

                <thead>

                    <tr>

                        <th>x</th>

                        <th>y</th>

                    </tr>

                </thead>

                <tbody>

                    {

                        points.map((point, index) => (

                            <tr key={index}>

                                <td>{point.x}</td>

                                <td>{point.y}</td>

                            </tr>)

                        )

                    }

                </tbody>

            </table>

            <h2>а) kP</h2>

            <h6>k({k}) \* P{JSON.stringify(P)} = {JSON.stringify(mul(k, P))}</h6>

            <h2>б) P + Q</h2>

            <h6>P{JSON.stringify(P)} + Q{JSON.stringify(Q)} = R{JSON.stringify(sum(P, Q))}</h6>

            <h2>в) kP + lQ - R</h2>

            <h6>{JSON.stringify(sum(sum(mul(k, P), mul(l, Q)), inverse(sum(P, Q))))}</h6>

            <h2>г) P - Q + R</h2>

            <h6>{JSON.stringify(sum(sum(P, inverse(Q)), sum(P, Q)))}</h6>

        </div>

    )

}