Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

«ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ»

Студент: Белицкий В.Д.

ФИТ 3 курс 5 группа

Преподаватель:

Савельева Маргарита Геннадьевна

Минск 2023

## 1. Описание приложения

Приложение позволяет выполнить следующие задачи:

* найти точки ЭК для значений *х*;
* выполнить операции над точками кривых;

## 2. Методика выполнения поставленных задач

Эллиптическая кривая (ЭК) определяется уравнением вида:

где *a* и *b* - параметры кривой, которые определяют ее форму и положение.

Определения:

* Бесконечно удаленная точка *O*: это особая точка на кривой, которая представляет собой "бесконечность". Она служит нейтральным элементом для операции сложения точек.
* Аффинные координаты точки: для представления точки (*x*, *y*) на эллиптической кривой используются аффинные координаты. *x* и *y* — это обычные координаты точки.
* Групповая операция сложения точек: на эллиптической кривой определена операция сложения точек, которая позволяет складывать две точки и получать третью точку на кривой. Операция сложения обладает свойствами ассоциативности, коммутативности и наличия нейтрального элемента.
* Удвоение точки: Удвоение точки *P* на эллиптической кривой означает сложение точки *P* с самой собой.
* Умножение точки на скаляр: Умножение точки *P* на скаляр *k* — это повторное применение операции сложения точек: *P* + *P* + *P* + ... + *P* (*k* раз).
* Порядок точки: Порядок точки *P* — это количество точек на эллиптической кривой, которые могут быть получены путем умножения точки *P* на скаляр.
* Циклическая подгруппа: Подгруппа на эллиптической кривой, которая состоит из всех точек, полученных путем умножения определенной точки на скаляры. Циклическая подгруппа имеет свой порядок, который является делителем порядка кривой.

Операции на ЭК, представлены на листинге 2.1.

public static int[] CalculateSum(int[] P, int[] Q, int p)

{

int lambda = Lambda(P, Q, p);

int x = NOD.Mod(lambda \* lambda - P[0] - Q[0], p);

int y = NOD.Mod(lambda \* (P[0] - x) - P[1], p);

return new int[] { x, y };

}

public static int[] CalculateSum(int[] P, int a, int p)

{

int lambda = Lambda(P, a, p);

int x = NOD.Mod(lambda \* lambda - P[0] - P[0], p);

int y = NOD.Mod(lambda \* (P[0] - x) - P[1], p);

return new int[] { x, y };

}

public static int[] scalarMultiple(int k, int[] P, int a, int p)

{

int[] scalarMultiple = P;

for (int i = 0; i < (int)Math.Log(k, 2); i++)

scalarMultiple = CalculateSum(scalarMultiple, a, p);

k = k - (int)Math.Pow(2, (int)Math.Log(k, 2));

while (k > 1)

{

for (int i = 0; i < (int)Math.Log(k, 2); i++)

scalarMultiple = CalculateSum(scalarMultiple, CalculateSum(P, a, p), p);

k = k - (int)Math.Pow(2, (int)Math.Log(k, 2));

}

if (k == 1) scalarMultiple = CalculateSum(scalarMultiple, P, p);

return scalarMultiple;

Листинг 2.1 – Операции над ЭК

Втрое задание состояло в том, чтобы написать оконное приложение, которое позволяет зашифровать/расшифровать собственную фамилию (или имя на ваш выбор) с использованием эллиптической кривой (ЭК), указанной в задании 1. Генерирующая точка *G* = (0, 1). Тайный ключ выбирается в соответствии с вариантом.

Вычислить значение открытого ключа *Q* самостоятельно. Для этого следует использовать основную формулу и соотношения для случая *P = Q*. Необходимо учесть, что все вычисления производятся по модулю 751.

Методы шифрования и расшифрования представлены в листинге 2.2

public static int[,] Encrypt(string text, int[] G, int a, int p, int d)

{

int[] Q = scalarMultiple(d, G, a, p), P;

int[,] encrText = new int[text.Length, 4];

int k;

Console.WriteLine($"G = ({G[0]}, {G[1]}), d = {d}, Q = ({Q[0]}, {Q[1]})");

for (int i = 0; i < text.Length; i++)

{

k = random.Next(2, d);

P = Enumerable.Range(0, points.GetLength(1)).Select(x => points[alphabeth.IndexOf(text[i]), x]).ToArray();

int[] C1 = scalarMultiple(k, G, a, p), kQ = scalarMultiple(k, Q, a, p), C2;

C2 = CalculateSum(P, kQ, p);

encrText[i, 0] = C1[0]; encrText[i, 1] = C1[1];

encrText[i, 2] = C2[0]; encrText[i, 3] = C2[1];

}

return encrText;

}

public static string Decrypt(int[,] encrText, int a, int p, int d)

{

string decryptedText = "";

for (int i = 0; i < encrText.GetUpperBound(0) + 1; i++)

{

int[] C1 = scalarMultiple(d, new int[] { encrText[i, 0], encrText[i, 1] }, a, p), C2 = { encrText[i, 2], encrText[i, 3] };

int[] P = CalculateSum(C2, InversePoint(C1), p);

for (int k = 0; k < points.GetUpperBound(0) + 1; k++)

{

if (points[k, 0] == P[0] && points[k, 1] == P[1])

{

decryptedText += alphabeth[k];

break;

}

}

}

return decryptedText;

}

Листинг 2.2 – Методы зашифрования/расшифрования для работы с ЭК

1. **Результаты работы приложения**

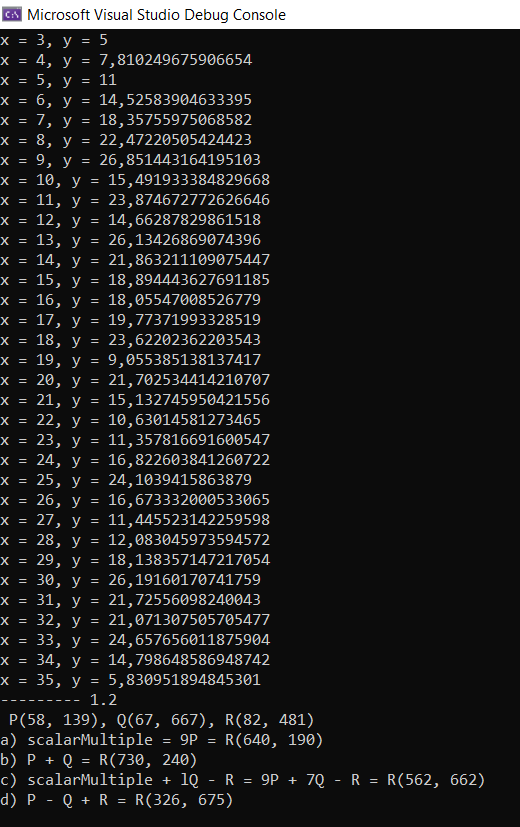
****

Рисунок 3.1 – Нахождение точек ЭК и операции над нимим

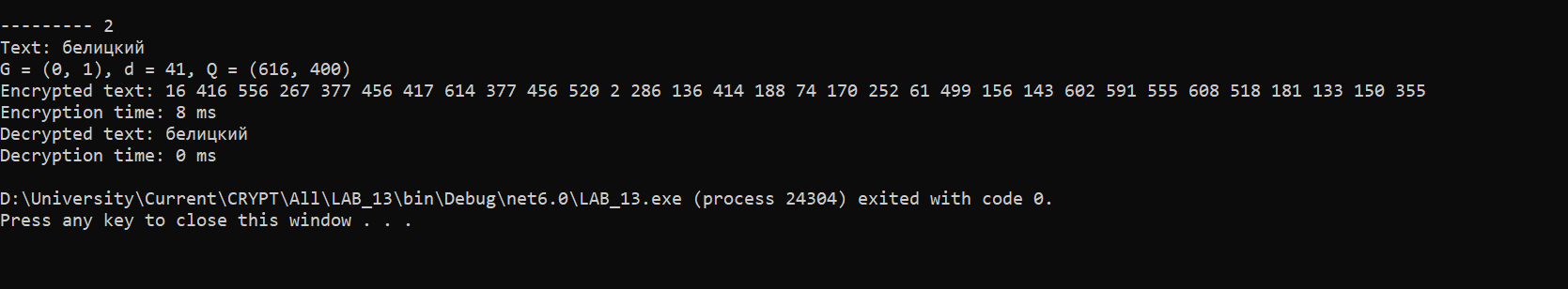


Рисунок 3.2 – Шифрование и Расшифрование с помощью ЭК

## 4. Анализ результатов

Для анализа результатов было принято решение проверки корректности данных через сторонее приложение, рисунок 4.1.

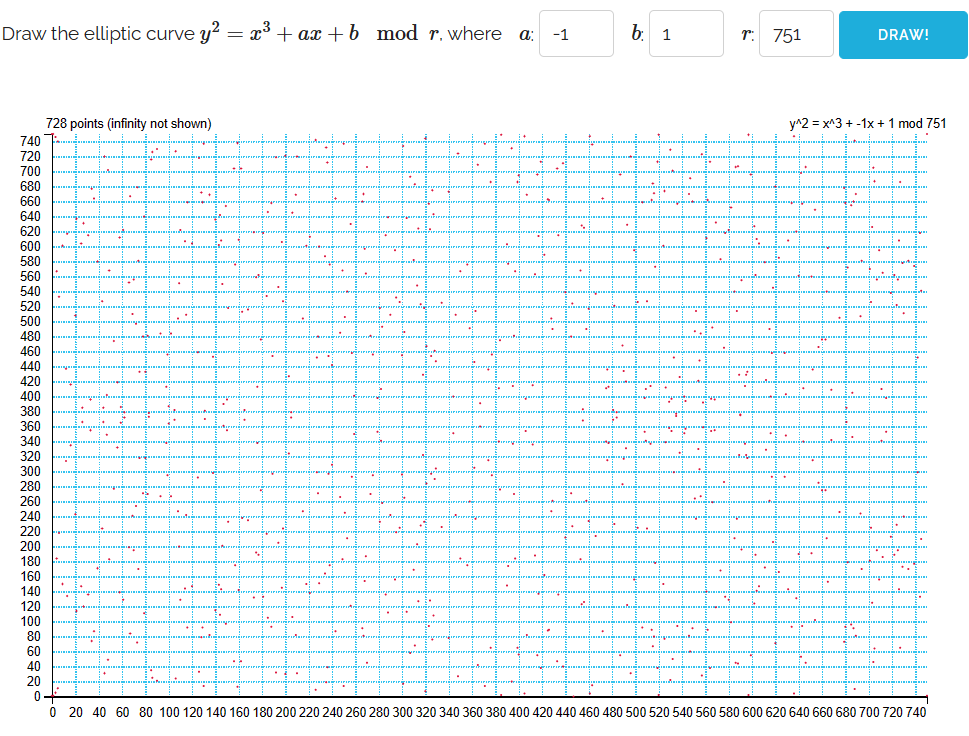


Рисунок 4.1 – Проверка корректности данных через онлайн приложение  
(источник : [Elliptic Curves over Finite Fields (graui.de)](https://graui.de/code/elliptic2/))

Проанализировав график, можно сделать вывод, что точки,которые были найдены в разработанном приложении выявлены корректно (часть точек на графике не отрисовывает ввиду того что их очень много).

Анализ результатов проводился путем оценки скорости выполнения зашифрования/расшифрования текста на основе ЭК, рисунок 4.2.

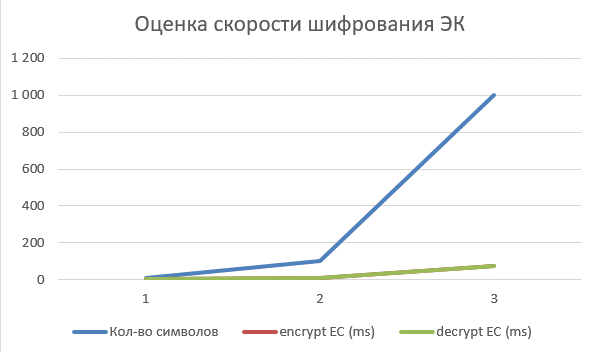


Рисунок 4.2 – Анализ результатов выполнения шифрования данных

На основе графика можно сделать вывод что расшифрование данных на основе ЭК производится очень быстро, и можно смело использовать этот метод шифрования в виду быстрой скорости исчисления.

## Вывод

В ходе лабораторной работы было разработано приложение для определения координат точек, операции над ЭК, а также, реализации зашифрования и расшифрования на основе ЭК. Усвоены навыки работы с ЭК, и с их основными методами.