МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«**ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ**»**

Студент: Бутурля Р.А.

ФИТ 3 курс 5 группа

Вариант 3

Преподаватель: Савельева М. Г.

Минск 2023

1. Найти точки ЭК для значений х. Разработать приложение для выполнения операций над точками кривой.

Для выполнения этой задачи был разработан ряд методов:

* Первый метод – находит точки ЭК
* Второй – производит вычисления операций над точками кривой

|  |
| --- |
| int xmin = 71, xmax = 105, a = -1, b = 1, p = 751;  for (int x = xmin; x <= xmax; x++)  {  Console.WriteLine($"x = {x}, y = {Math.Sqrt((x \* x \* x - x + b) % p)}");  } |

Листинг 1 – Метод расчета точек ЭК

|  |
| --- |
| Console.WriteLine("Задание 1.2");  int[] P = { 86, 25 }, Q = { 90, 21 }, R = { 83, 14 };  Console.WriteLine($"P({P[0]}, {P[1]}), Q({Q[0]}, {Q[1]}), R({R[0]}, {R[1]})");  int[] kP = EllipticCurves.kP(7, P, a, p);  int[] lQ = EllipticCurves.kP(8, Q, a, p);  Console.WriteLine($"kP = 7P = {kP.Select(el => el.ToString()).Aggregate((prev, current) => "R(" + prev + ", " + current + ")")}");  Console.WriteLine($"P + Q = {EllipticCurves.CalculateSum(P, Q, p).Select(el => el.ToString()).Aggregate((prev, current) => "R(" + prev + ", " + current + ")")}");  Console.WriteLine($"kP + lQ - R = 7P + 8Q - R = {EllipticCurves.CalculateSum(EllipticCurves.CalculateSum(kP, lQ, p), EllipticCurves.InversePoint(R), p).Select(el => el.ToString()).Aggregate((prev, current) => "R(" + prev + ", " + current + ")")}");  Console.WriteLine($"P - Q + R = {EllipticCurves.CalculateSum(EllipticCurves.CalculateSum(P, EllipticCurves.InversePoint(Q), p), R, p).Select(el => el.ToString()).Aggregate((prev, current) => "R(" + prev + ", " + current + ")")}");  Console.WriteLine(); |

Листинг 2 – Метод выполнения операций над точками кривой

1. Создать оконное приложение для зашифрования/расшифрования собственной фамилии (или имени – по выбору) на основе ЭК, указанной в задании 1, для генерирующей точки G = (0, 1). Вычислить самостоятельно значение открытого ключа Q.

Для выполнения поставленной задачи был разработан код, представенный на листинге 3.

|  |
| --- |
| Console.WriteLine("Задание 2");  string text = "романбутурля";  Console.WriteLine($"Текст для шифрования: {text}");  var stopwatch = Stopwatch.StartNew();  int[,] encryptedText = EllipticCurves.Encrypt(text, new int[] { 0, 1 }, a, p, 25);  stopwatch.Stop();  Console.WriteLine($"Зашифр текст: {string.Join(" ", encryptedText.Cast<int>())}");  Console.WriteLine($"Зашифр время: {stopwatch.ElapsedTicks} ticks");  stopwatch.Restart();  Console.WriteLine($"Расшифр текст: {EllipticCurves.Decrypt(encryptedText, a, p, 25)}");  stopwatch.Stop();  Console.WriteLine($"Расшифр время: {stopwatch.ElapsedTicks} ticks"); |

Листинг 3 – Метод шифрования и расшифрования сообщения

1. Создать оконное приложение для генерации/верификации ЭЦП на основе алгоритма ЕСDSA: ЭК Е751(–1, 1) c генерирующей точкой G = (416, 55); порядок точки q = 13.

Листинг разработанного метода представлени на листинге 4.

|  |
| --- |
| stopwatch.Restart();  int[] digitalSign = EllipticCurves.CreateDigitalSign(new int[] { 416, 55 }, 13, 7, a, p);  stopwatch.Stop();  Console.WriteLine($"Цифр подпись: {digitalSign.Select(el => el.ToString()).Aggregate((prev, current) => prev + " " + current)}");  Console.WriteLine($"Время создания цифр подписи: {stopwatch.ElapsedTicks}");  stopwatch.Restart();  Console.WriteLine($"Проверка валидности подписи: {EllipticCurves.VerifyDigitalSign(digitalSign, new int[] { 416, 55 }, 13, 7, a, p)}");  stopwatch.Stop();  Console.WriteLine($"Время проверки подписи: {stopwatch.ElapsedTicks}"); |

Листинг 4 – Метод генерации и верификации ЭЦП

График зависимости вермени выполнения шифрования данных от алгоритма представлени ниже.

**Вывод**

Изучил и приобрел практические навыки разработки и использования приложений для реализации криптографических алгоритмов на основе эллиптических кривых.