Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование криптографических алгоритмов на основе эллиптических кривых**

Студент: Каспер Н.В.

ФИТ 3 курс 5 группа

Преподаватель: Берников В.О.

Минск 2020

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* находить точки эк для значений x;
* выполнять операции над точками кривой;
* шифровать и расшифровывать фамилию на основе ЭК.
* генерировать ЭЦП на основании алгоритма EDSA;
* верифицировать ЭЦП.

1. **Методика выполнения поставленных задач**

Эллиптические кривые – математический объект, который может быть определен над любым полем.

Эллиптическая кривая над вещественными числами – это множество точек, описываемых уравнением ;

Для того, чтобы найти значения точки ЭК из промежутка, необходимо выполнить написанное выше уравнение.

Результат работы приложения представлен на рисунке 2.1.

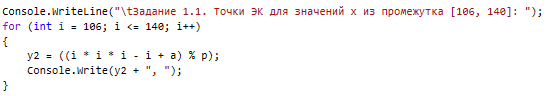


Рисунок 2.1 – Реализация задания

На рисунке 2.2 представлена реализация функции для умножения, которая используется для основных операций над точками ЭК. Важно отметить, что умножение описывается задачей логарифмирования, а также выполняется путем суммирования значения P несколько раз.

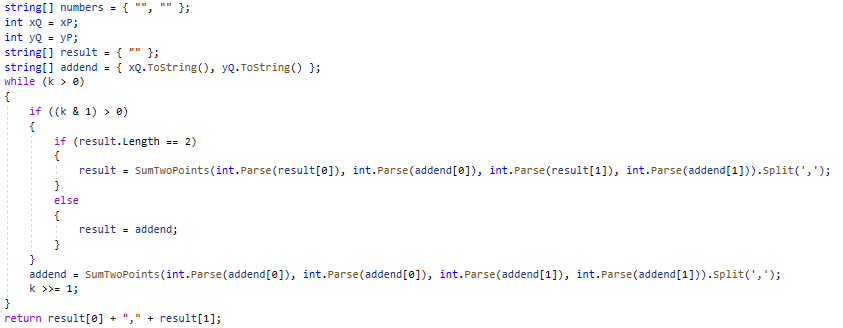


Рисунок 2.2 – Функция Multiply

В данной функции используется метод SumTwoPoint, реализация которого представлена в листинге 1.

Листинг 1. Реализация SumTwoPoint

|  |
| --- |
| public static string SumTwoPoints(int xP, int xQ, int yP, int yQ)  {  int p = 751;  BigInteger lyambda;  int raznX = xQ - xP;  int raznY = yQ - yP;  if (raznX < 0)  {  raznX += p;  }  if (raznY < 0)  {  raznY += p;  }  if (xP == 0 & yP == 0)  {  return xQ.ToString() + ',' + yQ.ToString();  }  if (xQ == 0 & yQ == 0)  {  return xP.ToString() + ',' + yP.ToString();  }  BigInteger xR = 0, yR = 0;  if (xP == xQ && yP != yQ || (yP == 0 && yQ == 0 && xP == xQ))  { }  else  {  if (xP == xQ && yP == yQ)  {  lyambda = (3 \* BigInteger.Pow(xP, 2) - 1) \* (Foo(2 \* yP, p));  }  else  {  lyambda = (raznY) \* Foo(raznX, p);  }  xR = (BigInteger.Pow(lyambda, 2) - xP - xQ);  yR = yP + lyambda \* (xR - xP);  xR = xR % p < 0 ? (xR % p) + p : xR % p;  yR = -yR % p < 0 ? (-yR % p) + p : (-yR % p);  }  string Result = xR.ToString() + ',' + yR.ToString();  return Result;  } |

При использовании ЭК зашифрование предполагает представление сообщения в виде точки Р (или представления каждого блока сообщения в виде разных точек Рi) ЭК с известной точкой G и известным Q. Соответственно шифртекст – это две точки на той же ЭК: С1 и C2 или Сi1 и Ci2.

Реализация функции зашифрования представлен на рисунке 2.3.

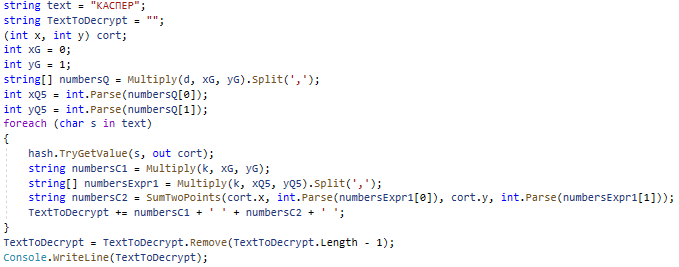


Рисунок 2.3 – Функция зашифрования

Для расшифрования сторона B вычисляется dC1. Реализация алгоритма расшифрования представлена на рисунке 2.4.

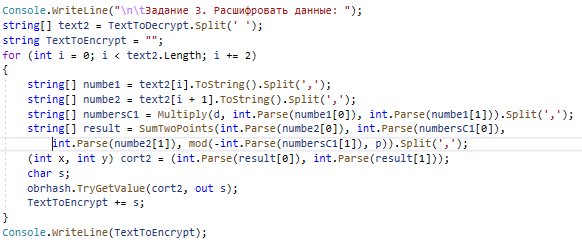


Рисунок 2.4 – Реализация расшифрования

Генерация ЭЦП состоит из нескольких этапов, в результате чего формируются числа k, q, r, t и s.

1. Выбрать число k (1 < k < q), q – порядок точки G.

2. Вычислить точку kG = (х, у), вычислить r = x mod q; при r = 0 изменить k и повторить шаг 2.

3. Вычислить t = k-1mod q (например, на основе расширенного алгоритма Евклида).

4. Вычислить s = (t (H(M) + dr)) mod q; при s = 0 изменить k и повторить алгоритм.

Другой стороне пересылаются сообщение M и ЭЦП (числа r и s).

Реализация генерации ЭЦП представлена на рисунке 2.5.

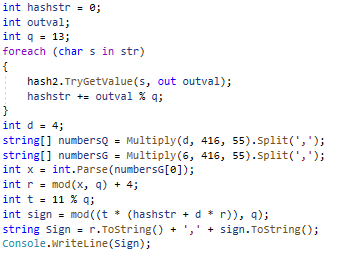


Рисунок 2.5 – Генерация ЭЦП

Верификация ЭЦП состоит из следующих этапов:

1. Проверить выполнение условия: 1 < r, s < q; если условие не выполняется, то легитимность подписи не подтверждается, в противном случае – выполняются дальнейшие шаги.

2. Вычисляются Н(М) и w = s–1 mod q.

3. Вычисляются u1 = w Н(М) (mod q), u2 = wr (mod q).

4. Вычисляются Gu1 + Qu2 = (x', y'), v = x' mod q.

5. Сравниваются v и r; если равенство выполняется, подтверждается легитимность подписи и целостность полученного сообщения.

Проверка ЭЦП представлена на рисунке 2.6.

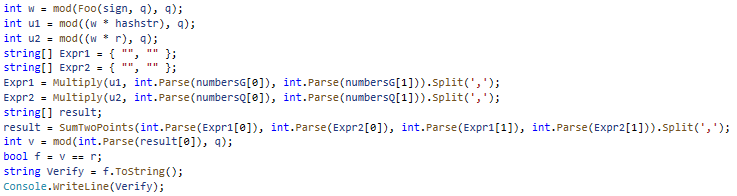


Рисунок 2.6 – Верификация ЭЦП

Таким образом, были реализованы все поставленные задачи.

**Вывод**

В ходе лабораторной работы были приобретены навыки работы с эллиптическими кривыми для зашифрования, расшифрования и генерации ЭЦП, а также верификации.

Также было разработано приложение, на языке программирования C#, для реализации задач, связанных с эллиптическими кривыми.