МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №11 НА ТЕМУ:

Исследование криптографических алгоритмов на основе эллиптических кривых

Выполнила студентка 3 курса 1 группы

Пригодич Вера Валерьевна

Минск 2023

**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).

**Теоретические сведения**

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

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

*у2* = *х3* + *aх* + *b*

при этом константы (*а* и *b* – вещественные числа) должны удовлетворять условию

4*a*3 + 27*b*2 ≠ 0.

Условие исключает из рассмотрения кривые с особыми точками или особые кривые.

Частью ЭК является бесконечно удаленная точка (также известная как идеальная точка), которую мы обозначим символом *О.*

Группа для ЭК есть непустое множество, элементы которого являются точками ЭК, обладающими следующими свойствами:

• единичный элемент – это бесконечно удаленная точка *О*;

• обратная величина точки *R* – это точка, симметричная относительно оси Х;

• сложение задается следующим правилом: сумма трех ненулевых точек *P*, *Q* и –*R*, лежащих на одной прямой, будет равна *P* + *Q* + (–*R*) = *О*.

В соответствии с этим можем сформулировать законы сложения точек эллиптической кривой:

• прямая, проходящая через точки *R* и –*R*, является вертикальной прямой, которая не пересекает ЭК ни в какой третьей точке; если *R* = (*х*, –*у*), то *R* + (*х*, *у*) = *О*. Точка (*х*, *у*) является отрицательным значением точки *R* и обозначается –*R*. Таким образом, по определению *R* + (–*R*) = *О*;

• *P* + *Q* = *R*: пусть *P* и *Q* – две различные точки ЭК (рис. 1), и *Р* не равно *Q*; если проведем через *P* и *Q* прямую, то она пересечет ЭК еще только в одной точке, называемой –*R*; точка –*R* отображается относительно оси *Х* в точку *R*, равную сумме точек *P* и *Q*: *P* + *Q* = *R*.

Геометрическая интерпретация операции сложения двух точек показана на рис. 1.

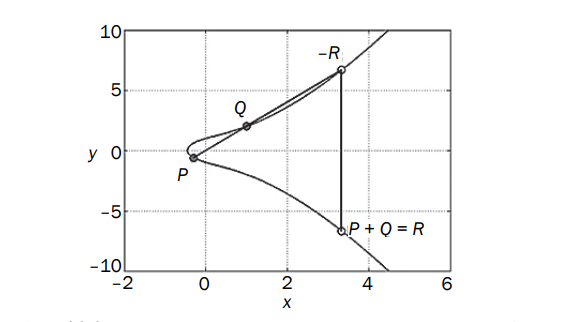


Рис 1 – Пояснение к операции сложения двух точек P и Q эллиптической кривой *у*2 = *х3* + 2*х* +1 (*а* = 2, *b* = 1)

Принцип умножения точки *Р* на целое положительное число *n* – это сумма *n* точек *Р*: *nP* = *P* + *P* + *P* + … + *P*.

Понятно, что каждая точка на плоскости задается парой координат: *х* и *у*. Числа *х* и *у* являются рациональными, а точки *P*, *Q*, *R* и –*R* (как и любые точки ЭК) – рациональными точками.

Если *Р* = (*х*1, *у*1) и *Q* = (*х*2, *у*2), то *Р* + *Q* = (*х*3, *у3*) определяется в соответствии с правилами:

*x*3= λ2 – *х*1 – *х*2;

*у*3= λ(*х*1 – *х*3) – *у*1,

где

λ = (*у*2 – *у*1)/(*х*2 – *х*1) при *Р* ≠ *Q* и λ = (3(*х*1)2 +*а*)/2*у*1 при *Р* = *Q*

Из этого следует, что число λ – угловой коэффициент секущей, проведенной через точки Р = (х1, у1) и Q = (х2, у2). При Р = Q секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ.

Эллиптическая кривая над полем *Fp* задается теми же уравнениями, что и ЭК над действительными числами, только все вычисления производятся по модулю *р* (mod *p*):

*у2* = *х3* + *aх* + *b* (mod *p*),

4*a*3 + 27*b*2 ≠ 0 (mod *p*),

и т.д.

Важно отметить, что, как и ранее, существует точка (бесконечно удаленная) *О*; *а* и *b* – вещественные числа

Для каждого x существует максимум две точки. Отметим также симметрию в расположении точек относительно *y* = *p*/2.

Если мы складываем два значения, кратных *Р*, то получаем значение, кратное *Р* (т. е. значения, кратные *Р*, замкнуты относительно операции сложения). Это означает, что множество кратных *Р* значений – это циклическая подгруппа группы, образованной эллиптической кривой.

Порядок группы точек эллиптической кривой равен числу различных точек ЭК, включая точку *О*.

Точка *Р* называется генератором или базовой точкой циклической подгруппы (такую точку во многих документах обозначают символом *G*). Порядок точки *Р* связан с порядком m ЭК теоремой Лагранжа, согласно которой порядок подгруппы – это делитель порядка исходной группы. Иными словами, если ЭК содержит *m* точек, а одна из подгрупп содержит *q*, то *q* является делителем *m*.

Как и в случае с непрерывными ЭК, теперь важным является вычисление некоторого числа *d*, если мы знаем *P* и *Q* для *Q* = *dP*. Это и есть задача дискретного логарифмирования для эллиптических кривых. Эта задача аналогична задаче дискретного логарифмирования, используемой в других криптосистемах, таких как алгоритм DSA, протокол Диффи – Хеллмана и схема Эль-Гамаля.

В криптографии на основе ЭК тайный ключ – это случайное целое *d*, выбранное из множества {1, 2, ..., *q* – 1}, где *q* – порядок подгруппы; открытый ключ – это точка *Q*, такая, что *Q* = *dG*, где *G* – базовая точка подгруппы.

Основные этапы генерации ключевой информациина основе ЭК

Первый этап: выбор (генерация) ЭК. Обычно он основан на выполнении следующих условий и операций.

1.1. Входными параметрами являются: число *l*, число *р*, удовлетворяющее условию 22*l* – 1 < *р* < 22*l* , *р* = 3 mod 4, 0 < *a* < *p*. Можно использовать некоторое простое число р = 22*l*– *с*, где *с* – небольшое натуральное число.

1.2. Выбирается число *b* такое, что 0 < *b* < *p*.

Таким образом, задана ЭК: *Ер*(*а*, *b*).

1.3. Выбираются порядок *q* (простое число) и генерирующая точка *G*, которая задается двумя координатами, например, *G* = (0, *уG*). Дополнительно к рассмотренным действиям стандарт [51] предусматривает использование вспомогательного параметра (*s*, *seed*) – произвольное 64-битное число.

2.1. Входными параметрами являются: *р*, *а*, *b*, *q* и *G*.

2.2. Генерируется тайный ключ – число *d*, выбранное из множества {1, 2, …, *q* – 1}.

2.3. Вычисляется открытый ключ – точка *Q*:

*Q* = *dG*,

к открытому ключу также относятся *р*, *а*, *b*, *q*.

3 направления использования ЭК в криптографии:

• в алгоритмах согласования (передачи) ключевой информации (на основе идеи Диффи – Хеллмана);

• в алгоритмах асимметричного шифрования/дешифрования сообщений;

• в алгоритмах генерации/верификации ЭЦП.

**Практическое задание**

**Вариант 8**

В основе задания – ЭК вида *у*2 = *х*3 – *х* + 1 (mod 751): *а* = –1, *b* = 1, *р* = 751, т. е. *Е*751(–1, 1).

Задание 1.

* 1. Найти точки ЭК для значений *х*: *x­min*  = 481и *xmax* = 515.

Координаты расположения точек должны быть ограничены квадратом некоторых чисел по модулю 751.

Найдем *y* для *x* = 481.

*у*2 = *х*3 – *х* + 1 (mod 751)

*у*2 = 4813 – 481 + 1 = 111284161 (mod 751)

*у*2= 230

Проверим, является ли 230 квадратичным вычетом по модулю 751. Воспользуемся критерием Эйлера.

Число *a*, взаимно простое с *p*, является квадратичным вычетом по модулю *p* тогда и только тогда, когда *a*(p−1)/2≡1(mod *p*).

230(750 – 1)/2(mod 751) = 1

Число 230 является квадратичным вычетом по модулю 751.

Наибольший общий делитель НОД (230; 751) = 1.

Необходимо найти такое *у*2 mod 751 = 230. Воспользуемся малой теоремой Ферма, которая утверждает, что если *p* – простое число и *a* не делится на *p*, то *a* (*p*–1) ≡ 1 (mod *p*).Квадратный корень из a может быть найден по модулю *p*, если выполняется следующее равенство: *a*(*p*+1)/4) ≡ ±1 (mod *p*).

Тогда *у*= *a* (*p* + 1) / 4) (mod *p*) = 230 (751 + 1) / 4 mod 751 = 382.

Таким образом получаем точку (481, 382).

Так как точки расположены симметрично относительно *y* = *p*/2, можем найти точку, симметричную полученной.

Это будет точка точку (481, 369).

Программно эти вычисления реализованы следующим образом:

def is\_quadratic\_residue(num, mod):

    # Проверка, является ли число квадратным вычетом

    return pow(num, (mod-1)//2, mod) == 1

def calculate\_y(x, a, b, p):

    # Функция для нахождения значений y на ЭК для заданного x

    x\_cubed = (x\*\*3 + a\*x + b) % p

    if not is\_quadratic\_residue(x\_cubed, p):

        return None  # x\_cubed не является квадратным вычетом, точка не находится на кривой

    y = pow(x\_cubed, (p+1)//4, p)  # Используем алгоритм квадратного корня Ферма

    return y

def get\_points\_in\_range(x\_min, x\_max, a, b, p):

    points = []

    for x in range(x\_min, x\_max+1):

        y = calculate\_y(x, a, b, p)

        if y is not None:

            points.append((x, y))

            points.append((x, p - y))  # Добавляем и отрицательное значение y

    return points

Листинг 1 – Нахождение точек в диапазоне *x­min*  и *xmax*

Результат выполнения программы будет иметь следующий вид:

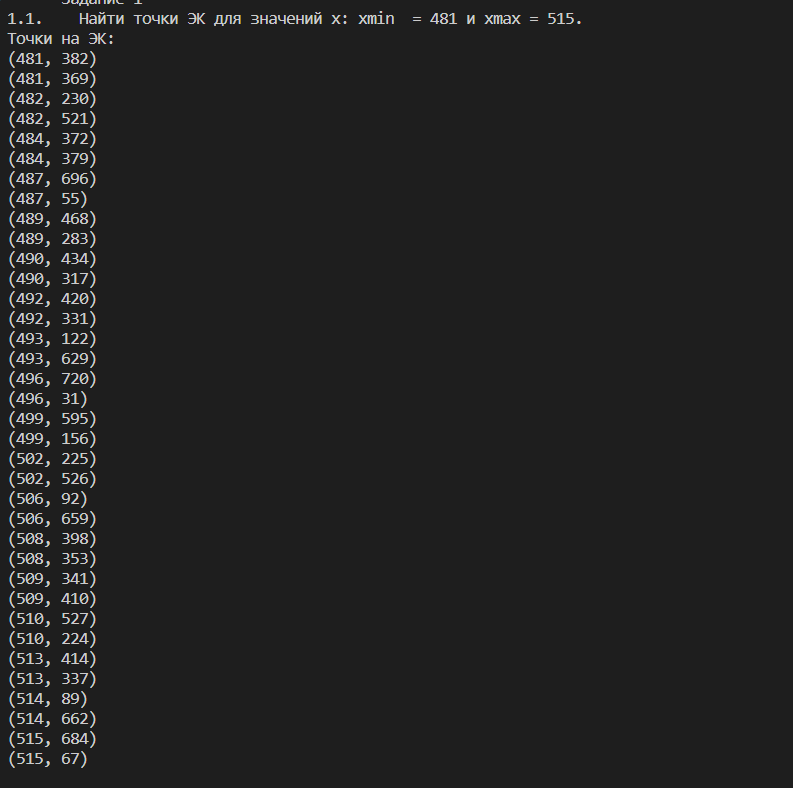


Рисунок 1 – Точки ЭК в диапазоне *x­min*  и *xmax*

* 1. Разработать приложение для выполнения операций над точками кривой:

а) *kР*; б) *Р* + *Q*; в) *kР* + *lQ* – *R*; г) *Р* – *Q* + *R*.

*k* = 12, *l* = 5.

Возьмем следующие координаты точек:

Координаты точек ЭК

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Координаты точек | | |
| *P* | *Q* | *R* |
| 8 | (48, 702) | (69, 241) | (98, 338) |

а) Умножим точку *P* (48, 702) на *k* = 12.

Принцип умножения точки *Р* на целое положительное число *n* – это сумма *n* точек *Р*: *nP* = *P* + *P* + *P* + … + *P*.

12*P* = 8*P* + 4*P*, где

8*P* = 4*P* + 4*P,*

4*P* = 2*P* + 2*P,*

2*P* = *P* + *P.*

Выполним приведенные операции в порядке снизу вверх.

Воспользуемся формулами

Если *Р* = (*х*1, *у*1) и *Q* = (*х*2, *у*2), то *Р* + *Q* = (*х*3, *у3*) определяется в соответствии с правилами:

*x*3= λ2 – *х*1 – *х*2 (mod p);

*у*3= λ(*х*1 – *х*3) – *у*1 (mod p),

где

λ = (3(*х*1)2 +*а*)/2*у*1 (mod p) при *Р* = *Q*

* 2*P* = *P* + *P*

λ = (3 · 482) – 1) / 2 · 702 (mod 751) = 259,

*x*3= 2592 – 48 – 48(mod 751) = 146,

*у*3= 259(48 – 146) – 702(mod 751) = 201.

Получаем точку 2*P* (146, 201).

* 4*P* = 2*P* + 2*P*

λ = (3 · 1462) – 1) / 2 · 201 (mod 751) = 316,

*x*3= 3162 – 146 – 146(mod 751) = 90,

*у*3= 316(146 – 90) – 201(mod 751) = 730.

Получаем точку 4*P* (90, 730).

* 8*P* = 4*P* + 4*P*

λ = (3 · 902) – 1) / 2 · 730 (mod 751) = 387,

*x*3= 3872 – 90 – 90(mod 751) = 416,

*у*3= 387(90 – 416) – 730(mod 751) = 55.

Получаем точку 8*P* (416, 55).

* 12*P* = 8*P* + 4*P*

В данном случае необходимо воспользоваться формулой λ = (*у*2 – *у*1)/(*х*2 – *х*1), так как *Р* ≠ *Q.*

λ = (730 – 55) / (90 – 416) mod 751 = 355,

*x*3= 3552 – 90 – 416(mod 751) = 102,

*у*3= 355(416 – 102) – 55(mod 751) = 267.

Можно проверить данные вычисления, воспользовавшись онлайн-сервисом https://andrea.corbellini.name/ecc/interactive/modk-mul.html:

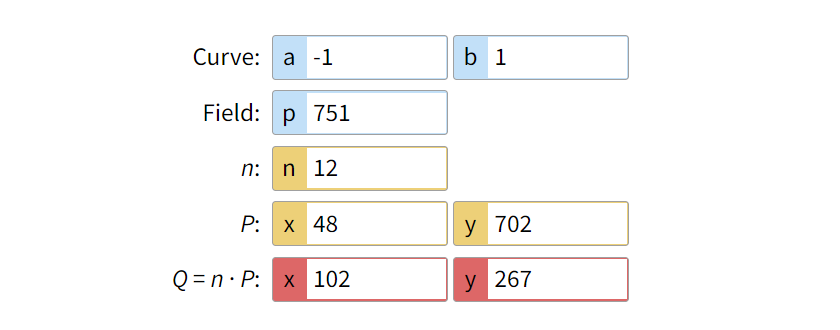


Рисунок 2 – Операция *kР*

б) Выполним операцию сложения точек *Р* + *Q*

Воспользуемся приведенными выше формулами того, чтобы сложить точки *P* (48, 702) и *Q* (69, 241).

λ = (241 – 702) / (69 – 48) mod 751 = 586,

*x*3= 5862 – 48 – 69 (mod 751) = 72,

*у*3= 586(48 – 72) – 702(mod 751) = 254.

Получаем точку (72, 254).

Проверяем данные вычисления:

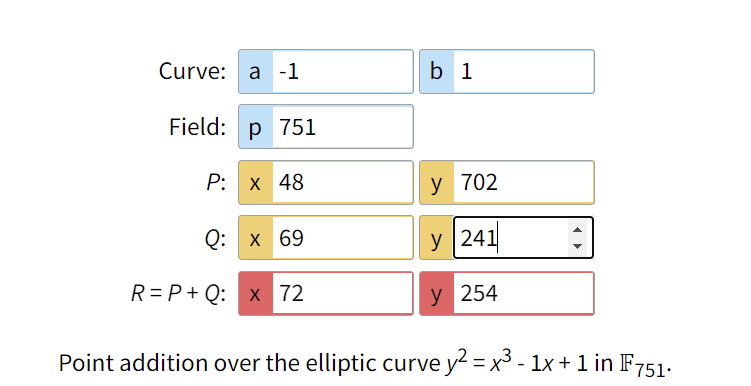


Рисунок 3 – Операция *Р* + *Q*

в) Найдем результат *kР* + *lQ* – *R*

Воспользуемся результатами вычисления из задания «а», *kP* = (102, 267).

Аналогичным образом найдем *lQ* = (376, 686).

Сложение двух разных точек проведем аналогично вычислению из задания «б», получим *kР* + *lQ* =(687, 91).

Для нахождения *kР* + *lQ* – *R* воспользуемся правилом, что *kР* + *lQ* – *R* = *kР* + *lQ* + (– *R*). Найдем (– *R*). Обратной точкой для точки *R* (*x*; *y*) на эллиптической кривой называют точку – *R* (*x*; – *y*).

*– y* mod *p* = – 338 mod 751 = 413.

Получаем точку – *R* (98, 413).

Результатом операции будет *kР* + *lQ* – *R* = (203, 324).

г) Найдем результат *Р* – *Q* + *R*.

Аналогично прошлому заданию найдем точку – *Q* как обратную точке *Q.* Получаем точку – *Q* (69, 510).

Результатом операций сложений будет *Р* – *Q* + *R* = (16, 335).

Программная реализация этих операций приведена в листинге 2:

def point\_addition(point1, point2, a, p):

    # Функция для сложения двух точек на ЭК

    if point1 is None:

        return point2

    if point2 is None:

        return point1

    x1, y1 = point1

    x2, y2 = point2

    if x1 == x2 and y1 == -y2 % p:

        return None  # Сложение точки с ее обратной дает бесконечно удаленную точку

    if x1 == x2 and y1 == y2:

        lam = (3 \* x1\*\*2 + a) \* mod\_inverse(2 \* y1, p) % p

    else:

        lam = (y2 - y1) \* mod\_inverse(x2 - x1, p) % p

    x3 = (lam\*\*2 - x1 - x2) % p

    y3 = (lam \* (x1 - x3) - y1) % p

    return x3, y3

def point\_multiplication(point, number,  a, p):

    # Функция умножения точки на число

    result = None

    current = point

    while number > 0:

        if number & 1 == 1:

            result = point\_addition(result, current, a, p)

        current = point\_addition(current, current, a, p)

        number >>= 1

    return result

def point\_negation(point, p):

    # Возращает обратную точку

    if point is None:

        return None

    x, y = point

    return x, -y % p

Листинг 2 – Операции над точками

Результат выполнения программы будет иметь следующий вид:

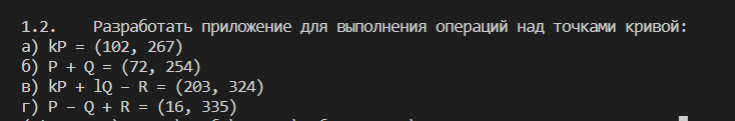


Рисунок 4 – Резльтат выполнения операций над точками

Задание 2.

2.1. Создать приложение для зашифрования/расшифрования собственной фамилии (или имени – по выбору) на основе ЭК, указанной в задании 1, для генерирующей точки *G* = (0, 1). Тайный ключ – в соответствии с вариантом из табл. 11.8.

2.2. Вычислить самостоятельно значение открытого ключа *Q*.

При этом была использована основная формула

*С*1 = *kG*, *С*2 = *P* + *kQ*,

а также соотношения

*x*3= λ2 – *х*1 – *х*2 (mod p);

*у*3= λ(*х*1 – *х*3) – *у*1 (mod p),

где

λ = (3(*х*1)2 +*а*)/2*у*1 (mod p) при *Р* = *Q*

Было принято, что шифруемым блоком является один символ сообщения, координаты которого на ЭК соответствуют табл. 11.9.

Значение параметра *k* было использовано из предыдущего задания.

Координаты точек ЭК

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Координаты точек | | |
| *P* | *Q* | *R* |
| 8 | (48, 702) | (69, 241) | (98, 338) |

Численное значение тайного ключа *d* = 34.

По заданным условия зашифруем сообщение «ВЕРА».

Найдем значение открытого ключа *Q* как *Q* = *dG*, где *G* – базовая точка подгруппы. Получаем значение *Q* (665, 153).

Результат вычисления открытого ключа приведен на рисунке 5.

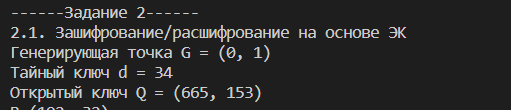


Рисунок 5 – Резльтат вычисления открытого ключа

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: *С*1 и *C*2 или *Сi*1 и *Ci*2.

Вычислим *С*1.

*С*1 = *kG*.

Возьмем значение *k* из предыдущего задания, *k* = 12. Генерирующая точка *G* (0, 1).

Получаем точку *C*1 (286, 136).

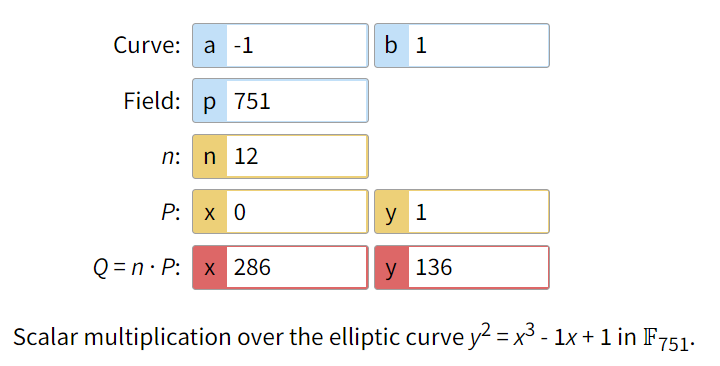


Рисунок 6 – Вычисление *С*1

Из таблицы 11.10 лабораторного практикума определим координаты точек, соответствующих символам шифруемого сообщения.

В – (192, 32)

Е – (194, 546)

Р – (206, 106)

А – (189, 297)

Зашифруем букву «В». Значение *С*1 вычислено выше, найдем *С*2 как *С*2 = *P* + *kQ*.

Сначала найдем значение *kQ*, *k* = 12, *Q* (665, 153). Получаем координаты точки (175, 192).

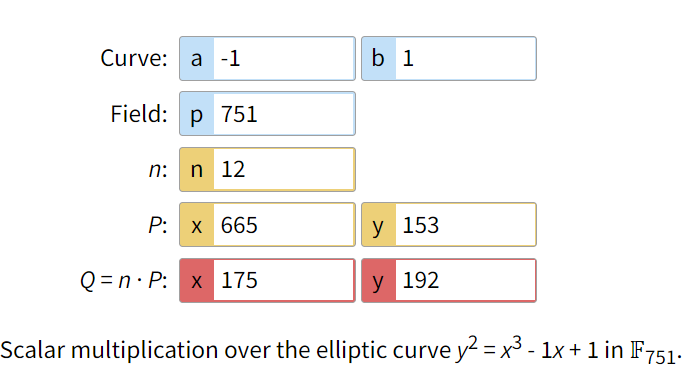


Рисунок 7 – Вычисление *kQ*

Далее необходимо получить сумму точки *kQ* (175, 192) и *P* (192, 32).

λ = (32 – 192) / (192 – 175) mod 751 = 344,

*x*3= 3442 – 175 – 192 (mod 751) = 62,

*у*3= 344(175 – 62) – 192(mod 751) = 379.

Получаем точку (62, 379).

Можно проверить полученное значение, используя онлайн-сервис.

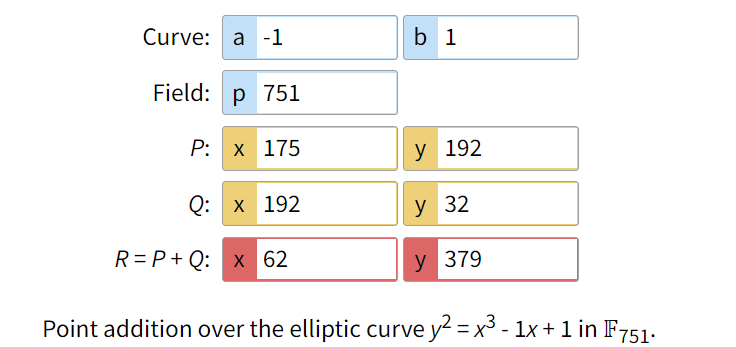


Рисунок 8 – Шифр буквы «В»

Можно заметить, что *C*1, *kQ* не зависят от входящего сообщения, поэтому для шифрования оставшихся символов необходимо только повторить предыдущую операцию сложения точек.

Зашифруем букву «Е». Суммируем точки *kQ* (175, 192) и *P* (194, 546).

λ = (546 – 192) / (194 – 175) mod 751 = 572,

*x*3= 5722 – 175 – 194 (mod 751) = 130,

*у*3= 572(175 – 130) – 192(mod 751) = 14.

Получаем точку (130, 14).

Зашифруем букву «Р». Суммируем точки *kQ* (175, 192) и *P* (206, 106).

λ = (106 – 192) / (206 – 175) mod 751 = 724,

*x*3= 7242 – 175 – 206 (mod 751) = 348,

*у*3= 724(175 – 348) – 192(mod 751) = 724.

Получаем точку (348, 724).

Зашифруем букву «А». Суммируем точки *kQ* (175, 192) и *P* (189, 297).

λ = (297 – 192) / (189 – 175) mod 751 = 383,

*x*3= 3832 – 175 – 189 (mod 751) = 631,

*у*3= 383(175 – 631) – 192(mod 751) = 143.

Получаем точку (631, 143).

Для расшифрования сообщения необходимо вычислить: P = *С*2 – *dC*1.

Найдем значение *dC*1, *d* = 34, *C*1(286, 136). Получим координаты точки (175, 192). Проверим данное вычисление, используя онлайн-сервис.

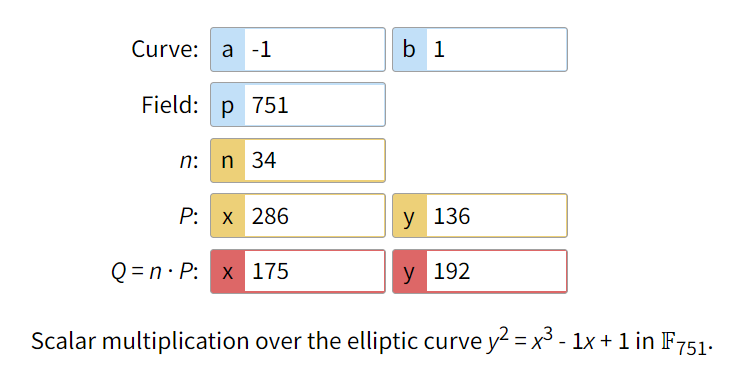


Рисунок 9 – Вычисление *dC*1

Найдем – *dC*1, как точку, обратную *dC*1.

– 192 mod 751 = 751 – 192 mod 751 = 559.

Получаем точку – *dC*1 (175, 559).

Расшифруем сообщение.

1. *C*2 (62, 379)

λ = (379 – 559) / (62 – 175) mod 751 = 407,

*x*3= 4072 – 175 – 62 (mod 751) = 192,

*у*3= 407(175 – 192) – 559(mod 751) = 32.

Получаем точку (192, 32). Находим в таблице 11.10 точку, соответсвующую данным координатам. Получаем букву «В».

1. *C*2 (130, 14)

λ = (14 – 559) / (130 – 175) mod 751 = 179,

*x*3= 1792 – 175 – 130 (mod 751) = 194,

*у*3= 179(175 – 192) – 559(mod 751) = 546.

Получаем точку (194, 546). Находим в таблице 11.10 точку, соответсвующую данным координатам. Получаем букву «Е».

1. *C*2 (348, 724)

λ = (724 – 559) / (348 – 175) mod 751 = 27,

*x*3= 272 – 175 – 348 (mod 751) = 206,

*у*3= 27(175 – 206) – 559(mod 751) = 106.

Получаем точку (206, 106). Находим в таблице 11.10 точку, соответсвующую данным координатам. Получаем букву «Р».

1. *C*2 (631, 143)

λ = (143 – 559) / (631 – 175) mod 751 = 368,

*x*3= 3682 – 175 – 631 (mod 751) = 189,

*у*3= 368(175 – 189) – 559(mod 751) = 297.

Получаем точку (189, 297). Находим в таблице 11.10 точку, соответсвующую данным координатам. Получаем букву «А».

Для реализации в приложении этого задания, воспользуемся функциями, разработанными в предыдущем.

Результат работы приложения будет иметь следующий вид:

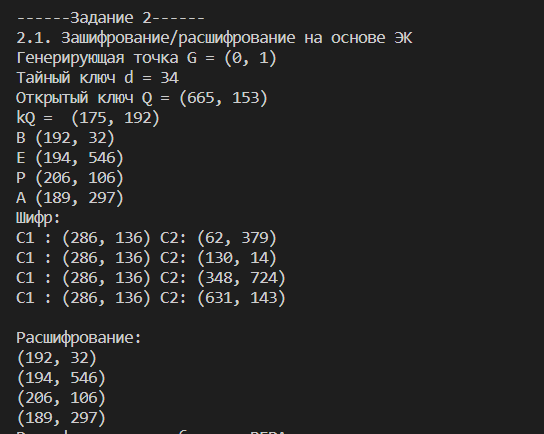


Рисунок 10 – Результат работы приложения

Задание 3.

3.1. 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.

3.2. Вычислить самостоятельно значение открытого ключа *Q*. Параметры *k* – по собственному усмотрению.

Найдем значение открытого ключа *Q* как *Q* = *dG*, где *G* – базовая точка подгруппы. Получаем значение *Q* (455, 368).

Численное значение тайного ключа *d* = 9.

Для генерации ЭЦП необходимо выполнить следующие шаги:

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

Пусть *k* = 2.

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

*kG* = 2*G* = *G* + *G*

λ = (3 · 4162) – 1) / 2 · 55 (mod 751) = 439,

*x*3= 4392 – 416 – 416(mod 751) = 384,

*у*3= 439(416 – 384) – 55(mod 751) = 475.

Получаем точку 2*G* (384, 475).

Найдем *r*.

*r* = 384 mod 13 = 7.

3. Вычислить *t* ≡ *k*–1 mod *q* (например, на основе расширенного алгоритма Евклида).

Найдем *t* ≡ *k*-1 mod *q* = 2-1 mod 13.

13 = 2\*6 + 1

*t* = (13 – 6) mod 13 = 6.

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

Пусть *H*(*M*) = 11, тогда

*s* = (*t* (*H*(*M*) + *dr*)) mod *q* = 6 \* (11 + 9 \* 7) mod 13 = 2.

Подпись сообщения – 7, 2.

Верификация ЭЦП. Получатель знает алгоритм хеширования, который использовался отправителем, открытый ключ отправителя, с помощью чего выполняет следующие операции над *М* и полученной ЭЦП (числа *r* = 7 и s = *2*).

1. Подтверждается выполнение условия 1 < *r*, *s* < *q*.

2. Вычисляется *Н*(*М*) – положим, что в результате хеширования полученного сообщения *М* его хеш не изменился: *Н*(*М*) = 12; далее вычисляется *w* = *s*–1mod *q* = 2–1 mod 13 = 6.

3. Вычисляются *u*1 = *w* *Н*(*М*) (mod *q*) = 6 · 11 (mod 13) = 1 и *u*2 = *wr* (mod *q*) = 6 · 7 (mod 13) = 3.

4. Вычисляются *Gu*1 + *Qu*2 = 1(416, 55) + 3(455, 368) = (416, 55) + (416, 55) = (384, 475) = (*x*', *y*'); *v* = *x*' mod *q* = 384 mod 13 = 7.

5. Сравниваются *v* = 7 и r = 7: равенство выполняется – подтверждается легитимность подписи и целостность полученного сообщения М.

Программно генерация и верификация ЭЦП реализовано следующим образом:

def sign(hash\_int, G, q, d, a, p):

    r = 0

    s = 0

    while not r or not s:

        k = random.randint(1, q)

        x, y = point\_multiplication(G, k, a, p)

        r = x % q

        s = ((hash\_int + r \* d) \* mod\_inverse(k, q)) % q

    print('k', k)

    print('x', x)

    return r, s

def verify(hash\_int, signature, G, Q, q, a, p):

    r, s = signature

    w = mod\_inverse(s, q)

    u1 = (hash\_int \* w) % q

    u2 = (w \* r) % q

    Gu1 = point\_multiplication(G, u1, a, p)

    Qu2 = point\_multiplication(Q, u2, a, p)

    x, y = point\_addition(Gu1, Qu2, a, p)

    return (r % q) == (x % q)

Листинг 3 – Генерация и верификация ЭЦП

Результат выполнения программы будет иметь следующий вид:

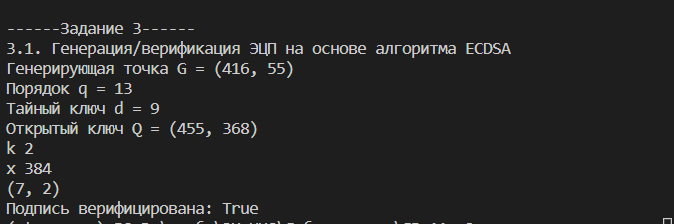


Рисунок 11 – Результат работы приложения

Вывод: были изучены и приобретены практические навыки разработки и использования приложений для реализации криптографических алгоритмов на основе эллиптических кривых.

**Контрольные вопросы**

1. Дать определение эллиптической кривой.

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

1. Записать уравнение ЭК над вещественными числами (ЭК в криптографии, ЕСС).

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

*у2* = *х3* + *aх* + *b*

при этом константы (*а* и *b* – вещественные числа) должны удовлетворять условию

4*a*3 + 27*b*2 ≠ 0.

1. Объяснить и показать на примере правила выполнения основных операций над точками ЭК.

Законы сложения точек эллиптической кривой:

• прямая, проходящая через точки *R* и –*R*, является вертикальной прямой, которая не пересекает ЭК ни в какой третьей точке; если *R* = (*х*, –*у*), то *R* + (*х*, *у*) = *О*. Точка (*х*, *у*) является отрицательным значением точки *R* и обозначается –*R*. Таким образом, по определению *R* + (–*R*) = *О*;

• *P* + *Q* = *R*: пусть *P* и *Q* – две различные точки ЭК, и *Р* не равно *Q*; если проведем через *P* и *Q* прямую, то она пересечет ЭК еще только в одной точке, называемой –*R*; точка –*R* отображается относительно оси *Х* в точку *R*, равную сумме точек *P* и *Q*: *P* + *Q* = *R*.

Если *Р* = (*х*1, *у*1) и *Q* = (*х*2, *у*2), то *Р* + *Q* = (*х*3, *у3*) определяется в соответствии с правилами:

*x*3= λ2 – *х*1 – *х*2;

*у*3= λ(*х*1 – *х*3) – *у*1,

где

λ = (*у*2 – *у*1)/(*х*2 – *х*1) при *Р* ≠ *Q* и λ = (3(*х*1)2 +*а*)/2*у*1 при *Р* = *Q*

Пусть ЭК задается уравнением с параметрами *а* = –7, *b* = 10. Точки *Р* (1, 2) и *Q* (3, 4). Нужно вычислить сумму точек: *P* + *Q* = *R.*

λ = (2 – 4) / (1 – 3) = 1,

*xR* = *x*3 = 12 – 1 – 3 = –3,

*yR* = *y*3 = 1 · (1 + 3) – 2 = 2.

Таким образом, получили точку *R* (–3, 2).

Результат совпадает с результатом в приложении (рис. 12)

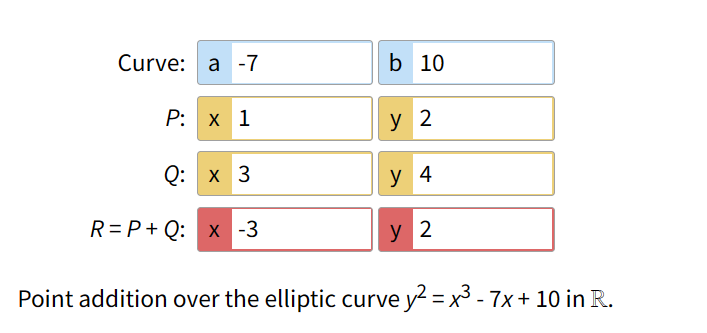


Рисунок 12 – Сложение точек *P* и *Q* для кривой y2 = x3 – 7x + 10

Для той же ЭК при *P* (1, 2) = *Q* (1, 2) получим для *P* + *Q* = *R* = 2*Р*:

λ = (3 · 12 – 7) / (2 · 2) = –1,

*xR* = *x*3 = (–1)2 – 1 – 1 = –1,

*yR* = *y*3 = –1(1+1) – 2 = –4.

Таким образом, получили точку 2Р (–1, –4).

Результат совпадает с результатом в приложении (рис. 13)

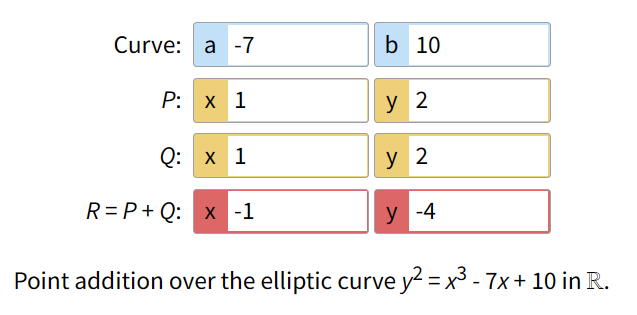


Рисунок 13– Сложение точек *P* и *Q* для кривой y2 = x3 – 7x + 10

1. Что такое «рациональная точка»?

В теории чисел и алгебраической геометрии рациональной точкой алгебраического многообразия является точка, координаты которой принадлежат заданному полю.

1. Как производится умножение точки ЭК?

Принцип умножения точки *Р* на целое положительное число *n* – это сумма *n* точек *Р*: *nP* = *P* + *P* + *P* + … + *P*.

1. Как производится умножение точки *Р* на число *k*, если *k* принимает значение: 2, 5, 11, 20, 32, 100, 256, 751, 1024?

Для *k* = 2:

2P = *P* + *P*

Для *k* = 5:

4*P* = 2*P* + 2*P*

5*P* = 4*P* + *P*.

Для *k* = 11:

8*P* = 4*P* + 4*P*

11*P* = 8*P* + 2*P* + *P*

Для *k* = 20:

16*P =* 8*P +* 8*P*

20*P* = 16*P* + 4*P*

Для *k* = 32:

32*P* = 16*P* + 16*P*

Для *k* = 100:

64*P* = 32*P +* 32*P*

100*P =* 64*P* + 32*P* + 4*P*

Для *k* = 256:

128*P* = 64*P* +64*P*

256*P* = 128*P* + 128*P*

Для *k* = 751:

512*P* = 256*P* + 256*P*

751*P* = 512*P* + 128*P* + 64*P* + 32*P* + 8*P* + 4*P* + 2*P* + *P*

Для *k* = 1024:

1024*P* = 512*P* + 512*P*

1. Составить алгоритм многократного сложения точки ЭК (умножения точки на число) на основе примера 7.

Вход: точка *P* на эллиптической кривой *E* и натуральное число *k*.

Выход: точка *Q = kP*.

* Инициализируем переменную *Q = O*, где *O* – это точка в бесконечности на кривой ЭК.
* Представляем число k в двоичном виде.
* Для каждого бита bi числа *k* в двоичном представлении, начиная с младшего бита:
  + Удваиваем точку *Q: Q = 2Q*.
  + Если bi равен 1, то прибавляем к точке *Q* исходную точку *P*:

*Q = Q + P.*

* После прохождения всех битов числа *k*, точка *Q* будет равна *kP*.
* Возвращаем точку *Q*.

1. Привести расчеты для точки *Q* при известных *d* и *G* из примера 7.

ЭК вида *Е*67(2, 3), *G* = (2, 22) и *d* = 4. Тогда *Q* = *dG* = 4*G*

4*G* = 2*G* + 2*G*

2*G* = *G* + *G*

Воспользуемся формулами:

*x*3= λ2 – *х*1 – *х*2 (mod p);

*у*3= λ(*х*1 – *х*3) – *у*1 (mod p),

где

λ = (3(*х*1)2 +*а*)/2*у*1 (mod p) при *Р* = *Q*

λ = (3 · 2^2 + 2) / (2 · 22) mod 67 = 46,

*x*3 = 462 – 2 – 2(mod 67) = 35,

*у*3 = 46(2 – 35) – 22(mod 67) = 1.

Получаем точку 2*G* (35, 1).

Вычислим 4*G*.

λ = (3 · 35^2 + 2) / (2 · 1) mod 67 = 63,

*x*3 = 632 – 35 – 35(mod 67) = 13,

*у*3 = 63(35 – 13) – 1(mod 67) = 45.

Получаем точку *Q* (13, 45)

9. Есть ли отличия в применении операций над точками ЭК над конечными полями и над действительными числами?

Основное отличие заключается в том, что операции над точками на эллиптических кривых над конечными полями производятся с помощью арифметики в конечных полях (например, полях Галуа), тогда как операции над точками на эллиптических кривых над действительными числами производятся с помощью арифметики действительных чисел.

В арифметике конечных полей умножение и сложение выполняются по модулю простого числа *p*, которое определяет размер поля. Это позволяет производить вычисления над точками на эллиптических кривых с использованием целочисленной арифметики, что важно для реализации криптографических алгоритмов на компьютерах.

С другой стороны, в арифметике действительных чисел умножение и сложение производятся без ограничений на размер чисел. В этом случае вычисления над точками на эллиптических кривых производятся с помощью вещественной арифметики, что может быть менее эффективным в плане вычислительной сложности.

10. Записать уравнение ЭК при формальном ее представлении в следующем виде: Е*р*(*а*, *b*).

Эллиптическая кривая над полем задается теми же уравнениями, что и ЭК над действительными числами, только все вычисления производятся по модулю *р* (mod *p*):

*у2* = *х3* + *aх* + *b* (mod *p*),

4*a*3 + 27*b*2 ≠ 0 (mod *p*),

и т.д.

11. Из какого числа точек состоит ЭК *Е*11(6, –9)? Дать их координаты.

Для определения числа точек на кривой E11 (6,-9) необходимо вычислить количество точек на этой кривой в пределах конечного поля  Z11. Для этого можно воспользоваться теоремой Хассе-Вейля, которая гласит, что количество точек на кривой Ep (*a*, *b*) в конечном поле Fp  примерно равно *p* + 1 - *t*, где *t* - параметр кривой, называемый порядком кривой.

Для кривой E11 (6,-9) порядок t можно вычислить, перебрав все точки на кривой и считая их количество. Воспользуемся для этого алгоритмом полного перебора точек, который заключается в последовательном переборе всех возможных значений координат (*x*, *y*) в конечном поле  Z 11 и проверке, является ли точка (*x*, *y*) находится на кривой  E11 (6,-9).

Результаты перебора показывают, что кривая  E11 (6,-9) состоит из 12 точек, перечисленных в таблице ниже, в которой координаты точек записаны в виде (*x*, *y*):

(0, 3), (0, 8), (1, 2), (1, 9), (6, 0), (6, 11), (7, 2), (7, 9), (8, 3), (8, 8), (9, 4), (9, 7).

12. Найти все точки ЭК Е11(1, 2).

13. На чем основа криптостойкость систем на основе ЭК? Области применения ЭК в криптографии.

Криптостойкость систем на основе эллиптических кривых (ЭК) основана на сложности задачи дискретного логарифмирования на кривых ЭК. В частности, для заданной точки P и числа k, которое является секретным ключом, найти точку Q = kP является вычислительно сложной задачей, к которой пока нет эффективных алгоритмов решения.

Одно из наиболее распространенных применений ЭК в криптографии – это создание криптографических систем на основе ЭК, таких как ECDSA (эллиптическая криптография с открытым ключом) и ECDH (эллиптический протокол Диффи-Хеллмана), которые используются для аутентификации и обмена ключами в интернет-протоколах SSL/TLS.

14. Что такое «порядок точки» ЭК? Показать на примере. Какую роль этот параметр играет в криптографии на основе ЭК?

Порядок точки относительно замкнутой кривой на плоскости – это целое число, представляющее число полных оборотов, которое делает кривая вокруг заданной точки против часовой стрелки.

Порядок точки Р связан с порядком m ЭК теоремой Лагранжа, согласно которой порядок подгруппы – это делитель порядка исходной группы. Иными словами, если ЭК содержит *m* точек, а одна из подгрупп содержит *q*, то *q* является делителем *m*.

15. Что такое «базовая точка» ЭК? Какую роль этот параметр играет в криптографии на основе ЭК?

Базовая точка (или точка генератор) на эллиптической кривой – это фиксированная точка P, которая используется в криптографии на основе эллиптических кривых для генерации случайных ключевых значений и подписей.

Роль базовой точки заключается в том, что она позволяет генерировать другие точки на кривой, которые могут быть использованы для создания ключей или подписей. В частности, любую точку Q на кривой можно представить в виде кратной суммы базовой точки P, т.е. Q = kP, где k - целое число. Это свойство позволяет использовать базовую точку для генерации случайных значений k, которые могут быть использованы в криптографии для создания ключей и подписей.

16. Объяснить порядок формирования ключевой информации на основе ЭК.

* Выбор параметров ЭК: выбираются параметры эллиптической кривой, такие как коэффициенты *a* и *b*, модуль *p* и базовая точка *P*.
* Генерация закрытого ключа: случайным образом выбирается целое число *d*, которое является закрытым ключом пользователя.
* Вычисление открытого ключа: открытым ключом пользователя является точка *Q = dP*, т.е. результат умножения базовой точки на закрытый ключ.
* Обмен ключами: для установления защищенного соединения пользователи обмениваются открытыми ключами и с помощью них вычисляют общий секретный ключ.
* Вычисление общего секретного ключа: общий секретный ключ вычисляется путем умножения открытого ключа другого пользователя на закрытый ключ текущего пользователя, т.е. *K = dQ = d*(*kP*), где *k* - закрытый ключ другого пользователя.

17. Сгенерировать ключевую информацию на основе кривой Е11(1, 2).