МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра дифференциальных уравнений и системного анализа

АНАЛИЗ ЭФФЕКТИВНОСТИ ВЫЧИСЛЕНИЯ ОПЕРАЦИИ СЛОЖЕНИЯ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ ПРИ ПОМОЩИ ПРОЕКТИВНЫХ КООРДИНАТ

Курсовая работа

Шклярик Веры Сергеевны
студентки 2-го курса
специальности 1-31 03 09
«Компьютерная математика
и системный анализ»

Научный руководитель:
кандидат физ.-мат. наук,
доцент Д. Н. Чергинец

Минск, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Эллиптические кривые являются одним из наиболее важных математических инструментов в современной криптографии. Многие современные криптографические системы сильно зависят от использования эллиптических кривых. В центре эллиптической криптографии находится операция сложения точек на кривой, которая лежит в основе многих криптографических протоколов и алгоритмов, таких как, например, алгоритм формирования и проверки электронной цифровой подписи.

Эффективность операции сложения точек на эллиптических кривых является ключевым аспектом при реализации алгоритмов на практике. Эллиптические кривые, используемые в криптографии, могут быть представлены в различных системах координат. Выбор системы координат может значительно влиять на скорость выполнения операции сложения. В данной работе мы рассмотрим алгоритмы сложения точек эллиптической кривой в аффинных координатах, проективных координатах, координатах Якоби и обобщенных координатах Якоби в конечном поле порядка p, где p – простое, $p > 3$ и выведем соответствующие формулы сложения для каждой системы координат. Кроме того, в ходе работы будут реализованы алгоритмы сложения точек эллиптической кривой в аффинных координатах, проективных координатах, координатах Якоби и обобщенных координатах Якоби на языке Python. Также мы сосредоточимся на анализе эффективности вычисления этой операции: проведем теоретический анализ эффективности изучаемых алгоритмов, оценив количество арифметических операций, проведем практический анализ эффективности изучаемых алгоритмов, оценив время работы реализованных в Python алгоритмов.

Результаты работы по анализу эффективности вычисления операции сложения точек эллиптической кривой в различных системах координат могут иметь широкое применение в криптографии. Это позволит выбирать оптимальные системы координат для конкретных задач и повышать эффективность вычислений при работе с эллиптическими кривыми.

ГЛАВА 1

Теоретические сведения и вывод формул

1.1 Эллиптическая кривая.

Впервые кубическая кривая, которую мы теперь называем эллиптической, встречается в рукописи "Арифметика" древнегреческого математика Диофанта. Задача 24 из книги IV звучит так: "Разделить данное число на два числа, произведение которых равно кубу без его стороны". Мы бы записали это уравнение следующим образом:

$$y(a - y) = x^3 - x.$$

Сам Диофант решал данное уравнение для a = 6 с помощью подстановки $x = 3y - 1$. Ньютон, работая в 17 веке над изучением кубических кривых, заметил, что Диофант в своем решении, по существу, пересек кривую $y(6 - y) = x^3 - x$ касательной $x = 3y - 1$. Это открытие помогло лучше понять операцию сложения точек, а также получить формулы сложения точек кубических кривых вида $y^2 = ax^3 + bx^2 + cx + d$, частным случаем которых и является эллиптическая кривая, рассматриваемая в данной работе. [1, с. 640], [2, с. 168]

Эллиптической кривой E над полем K называется множество точек $(x, y) \in K^2$, удовлетворяющих уравнению:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{1.1}$$

где $a_i \in K$. Такая кривая должна быть неособой в том смысле, что частные производные функции $F = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6$ по переменным x, y не должны обращаться в нуль одновременно ни в одной точке кривой. Кроме того, к эллиптической кривой добавляют бесконечно удаленную точку O. Уравнение (1.1) называется длинной формой уравнения Вейерштрасса в аффинных координатах.

Для эллиптической кривой E вводятся следующие константы, используемые в дальнейших формулах:

$$b_2 = a_1^2 + 4a_2,$$
$$b_4 = a_1a_3 + 2a_4,$$
$$b_6 = a_3^2 + 4a_6,$$
$$b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_4^2,$$
$$\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6,$$

где число Δ называют дискриминантом кривой E. Кривая неособа тогда и только тогда, когда $\Delta \neq 0$, что мы и будем полагать далее. В конечном поле порядка p, p > 3 допускается замена переменных

$$x = x' - \frac{b_2}{12},$$
$$y = y' - \frac{a_1}{2}(x' - \frac{b_2}{12}) - \frac{a_3}{2},$$

переводящая кривую E', заданную уравнением (1.1) в изоморфную ей кривую E, определяемую следующим уравнением (короткая форма Вейерштрасса):

$$y^2 = x^3 + ax + b \tag{1.2}$$

при некоторых $a, b \in K$. Дискриминант такой кривой равен $-16(4a^3 + 25b^2)$. На таких представителях эллиптических кривых можно наглядно ввести правила сложения точек с помощью метода хорд и касательных.

1.2 Правила сложения точек эллиптической кривой

Сложение точек эллиптической кривой лучше всего объяснить геометрически, с помощью хорд. Пусть P и Q — две различные точки кривой. Соединим их прямой линией, которая при этом обязательно пересечет кривую в некоторой третьей точке R, поскольку мы пересекаем кубическую кривую прямой. Полученная точка R определена над тем же полем K, что сама кривая и исходные точки P и Q. Отразим затем точку R относительно горизонтальной оси координат и получим точку, также определенную над полем K. Полученная точка и будет суммой $P + Q$.

Метод касательных служит для удвоения точки, поскольку точку нельзя сложить точку с собой, используя хорды. Пусть P — произвольная точка эллип-

тической кривой. Проведем касательную к кривой в точке P. Она пересечет кривую еще в какой-то одной точке R, поскольку кубическая кривая пересекается с прямой по трем точкам с учетом кратности пересечения. Отразив R относительно горизонтальной оси, мы получим точку $P + P$. Если в точке P проходит вертикальная касательная, то она «пересекает» кривую в бесконечно удаленной точке, то есть $P + P = O$. [3, с. 57]

1.3 Вывод формул сложения в аффинных координатах

Пусть $P = (x_1, y_1)$ и $Q = (x_2, y_2)$ — две точки на кривой E, заданной уравнением (1.2), а их сумма R имеет координаты (x_3, y_3). Предположим сначала, что среди точек P, Q нет бесконечно удаленной и $P \neq Q$. Проведем секущую через P, Q. Получаем, что угловой коэффициент данной секущей прямой

$$k = \frac{y_2 - y_1}{x_2 - x_1}.$$

Рассмотрим случай, когда $x_1 \neq x_2$. Тогда уравнение секущей имеет вид

$$y = k(x - x_1) + y_1. \tag{1.3}$$

Чтобы найти пересечение с эллиптической кривой E, подставим (1.3) в (1.2):

$$(y = k(x - x_1) + y_1)^2 = x^3 + ax + b.$$

Преобразовав, получим кубическое уравнение:

$$x^3 - k^2x^2 + x(a + 2ky_1 + 2k^2x_1) + (2kx_1y_1 + b - k^2x_1^2 - k^2y_1^2) = 0. \tag{1.4}$$

Воспользуемся теоремой Виета для кубического уравнения и получим, что

$$x_1 + x_2 + x_3 = k^2,$$

откуда

$$x_3 = k^2 - x_1 - x_2.$$

Наконец, отразим точку R относительно горизонтальной оси координат и получим

$$y_3 = k(x_1 - x_3) - y_1.$$

В случае, когда $x_1 = x_2$, но $y_1 \neq y_2$, через точки P, Q проходит вертикальная прямая, таким образом точкой $R = P + Q$ будет являться бесконечно удаленная точка O.

Теперь рассмотрим случай равенства точек $P = Q = (x_1, y_1)$. Проведем касательную к кривой E в точке P. Мы можем найти угловой коэффициент касательной, продифференцировав (1.2) как неявную функцию:

$$2y\frac{dy}{dx} = 3x^2 + a,$$

откуда

$$k = \frac{dy}{dx} = \frac{3x_1^2 + a}{2y_1}.$$

Если $y_1 = 0$, то касательная является вертикальной прямой и $R = P + Q = O$. Пусть далее $y_1 \neq 0$. Следуя аналогичным рассуждениям, получим уравнение (1.4). Теперь мы знаем единственный корень x_1 уравнения (1.2), но, учитывая его двойную кратность, получим

$$x_3 = k^2 - 2x_1, \quad y_3 = k(x_1 - x_3) - y_1.$$

Наконец, пусть $Q = O$. Прямая, проходящая через P и O — вертикальная, поэтому она пересекает кривую E в некоторой точке P', симметричной точке P относительно горизонтальной оси. По правилу сложения точек эллиптической кривой, чтобы получить точку $R = P + O$, нужно отразить точку P', что вновь даст нам точку P. Таким образом,

$$P + O = P$$

для любой точки P на кривой E. Заметим также, что если $P = O$, то $O + O = O$.

Подведем итог вышеизложенному, определив алгоритм сложения точек эллиптической кривой.
Пусть E — эллиптическая кривая, определяемая уравнением (1.2).

$P = (x_1, y_1)$ и $Q = (x_2, y_2)$ — две точки на кривой E, $P + Q = R = (x_3, y_3)$.

1. Если $x_1 \neq x_2$, то

$$x_3 = k^2 - x_1 - x_2, \quad y_3 = k(x_1 - x_3) - y_1, \tag{1.5}$$

где $k = \dfrac{y_2 - y_1}{x_2 - x_1}$.

2. Если $x_1 = x_2$, но $y_1 \neq y_2$, то $P + Q = O$.

3. Если $P = Q$ и $y_1 \neq 0$, то

$$x_3 = k^2 - 2x_1, \quad y_3 = k(x_1 - x_3) - y_1, \tag{1.6}$$

где $k = \dfrac{3x_1^2 + a}{2y_1}$.

4. Если $P = Q$ и $y_1 = 0$, то

$$P + Q = O. \tag{1.7}$$

Кроме того,

$$P + O = P$$

для любой точки P на кривой E.

Стоит отметить, что эллиптическая кривая E с операцией сложения является аддитивной абелевой группой. Сформулируем теорему:

Теорема 1. *Эллиптическая кривая E с операцией сложения обладает следующими свойствами:*

1. *Коммутативность: $P + Q = Q + P$ для всех точек P, Q на кривой E.*

2. *Существование нейтрального элемента: $P + O = P$ для всех точек P на кривой E.*

3. *Существование противоположного элемента: Для данной точки P на кривой E существует точка P' также на кривой E, такая, что $P + P' = O$. Такая точка P' обозначается $-P$.*

4. *Ассоциативность:* $(P + Q) + R = P + (Q + R)$ *для всех точек* P, Q, R *на кривой* E. *Таким образом, точки на кривой* E *образуют аддитивную абелеву группу с точкой* O *в качестве нейтрального элемента.*

[4, с. 15]

Доказательство. Коммутативность очевидным образом следует из того, что прямая, проходящая через точки P, Q, и прямая, проходящая через точки Q, P — это одна и та же прямая.
Свойство нейтрального элемента O

$$P + O = P$$

мы вывели, определяя формулы сложения точек.
Противоположным элементом для точки P естественным образом является точка $-P$, полученная отражением точки P относительно горизонтальной оси координат. Заметим, что для точек, лежащих на оси абсцисс, $P = -P$.
Для доказательства ассоциативности воспользуемся следующей теоремой:
[5, с. 22-24]

Теорема 2. *Если 8 из 9 точек пересечения двух троек прямых лежат на кривой третьего порядка, то девятая точка тоже лежит на ней.*

Заметим, что для доказательства ассоциативности можно предполагать, что среди точек P, Q, R нет бесконечно удаленной, поскольку в противном случае справедливость тождества $(P+Q)+R = P+(Q+R)$ очевидна, так как $P+O = P$ для любой точки P.
Теперь отметим, что точки $R, -R, O$ лежат на некоторой вертикальной прямой l_1. Точки $-P, -Q, P + Q$ лежат на некоторой прямой l_2, а точки $P, Q + R, T_1 = -(P + (Q + R))$ — на прямой l_3.
Далее, точки $P, -P, O$ лежат на некоторой прямой m_1, точки $-Q, -R, Q + R$ - на прямой m_2, точки $R, P + Q, T_2 = -((P + Q) + R)$ — на m_3.
Таким образом, 8 точек $(O, P, -P, R, -R, P + Q, Q + R, -Q)$, в которых пересекаются прямые $l_1, l_2, l_3, m_1, m_2, m_3$ лежат на кривой E. Тогда, согласно теореме 2, на кривой E лежит еще одна точка, в которой пересекаются прямые m_3 и l_3. Поскольку прямые m_3 и l_3 пересекаются в единственной точке, которая лежит

на E, а точки T_1 и T_2 принадлежат как l_3 и m_3 соответветственно, так и кривой E, то $T_1 = T_2$, откуда получаем

$$T_1 = -(P + (Q + R)) = -((P + Q) + R) = T2,$$

и, следовательно, требуемое равенство

$$(P + Q) + R = P + (Q + R).$$

□

1.4 Проективные координаты

Заметим, что в формулах сложения точек (1.5), (1.6) присутствует деление, в конечном поле представляющее из себя поиск обратного элемента при помощи расширенного алгоритма Евклида. Эта операция значительно медленнее, чем, например, умножение или сложение в поле конечного порядка. Для того, чтобы избежать вычисления обратных элементов, эллиптические кривые рассматривают в проективных координатах. Кроме того, проективные координаты помогают раскрыть смысл существования бесконечно удаленной точки на эллиптической кривой.

Проективная плоскость $\mathbb{P}(K)$ над полем K определяется как множество троек (X, Y, Z) не равных одновременно нулю элементов $X, Y, Z \in K$, на котором введено отношение эквивалентности:

$$(X, Y, Z) \sim (\lambda X, \lambda Y, \lambda Z)$$

для любых $\lambda \in K$. Класс эквивалентности таких троек называется проективной точкой [3, с. 53]. При этом плоскости, проходящие через точку $(0, 0, 0)$, называются проективными прямыми.

Для того, чтобы определить эллиптическую кривую на проективной плоскости, в уравнении (1.2) сделаем замену

$$x = X/Z, \quad y = Y/Z$$

Получим следующее уравнение:

$$Y^2Z = X^3 + aXZ^2 + bZ^3. \tag{1.8}$$

Таким образом, эллиптической кривой E в проективных координатах называется множество проективных точек, удовлетворяющих уравнению (1.8), называемому короткой формой уравнения Вейерштрасса в проективных координатах.

Чтобы определить бесконечно удаленную точку в проективных координатах, подставим в данное уравнение $Z = 0$. Получим, что $X^3 = 0$, то есть $X = 0$, а Y может принимать любое значение из поля K. Таким образом $(0, Y, 0) = (0, 1, 0)$ — это бесконечно удаленная аффинная точка O.
Рассмотрим две вертикальные прямые в аффинных координатах

$$x = c_1, \quad x = c_2, \quad c_1 \neq c_2.$$

Найдем их пересечение в проективной плоскости:

$$X = c_1Z, \quad X = c_2Z.$$

$$(c_1 - c_2)Z = 0, \quad c_1, c_2, Z \in K,$$

откуда

$$Z = 0, \quad X = 0.$$

Таким образом, мы получили, что две вертикальные прямые пересекаются в проективной точке $(0, Y, 0) = (0, 1, 0)$. Кроме того, это означает, что проективная точка $(0, 1, 0)$ лежит на всякой вертикальной прямой, а значит каждая вертикальная прямая пересекает кривую E в бесконечно удаленной точке.

1.5 Вывод формул сложения точек эллиптической кривой в проективных координатах

Определим правила перевода аффинных точек в проективные. Если координата Z проективной точки (X, Y, Z) отлична от нуля, то для удобства в качестве представителя класса эквивалентности можно рассматривать точку $(\frac{X}{Z}, \frac{Y}{Z}, 1)$. При этом точка $(\frac{X}{Z}, \frac{Y}{Z})$ удовлетворяет уравнению (1.2). Таким образом про-

ективная точка (X, Y, Z) кривой, отличная от бесконечно удаленной ($Z \neq 0$), переходит в аффинную точку с координатами $(\frac{X}{Z}, \frac{Y}{Z})$
Обратно, если конечная точка (x, y), удовлетворяет уравнению (1.2), то точка $(x, y, 1)$ удовлетворяет уравнению (1.8), то есть аффинная точка (x, y), отличная от бесконечно удаленной, переходит в проективную точку $(x, y, 1)$. Бесконечно удаленная аффинная точка O сопоставляется с проективной точкой $(0, 1, 0)$.

Пусть $P = (X_1, Y_1, Z_1), \quad Q = (X_2, Y_2, Z_2)$ - точки эллиптической кривой в проективных координатах. Найдем координаты их суммы $P + Q = R = (X_3, Y_3, Z_3)$. В аффинных координатах точки P, Q имеют вид

$$P = (\frac{X_1}{Z_1}, \frac{Y_1}{Z_1}), \quad Q = (\frac{X_2}{Z_2}, \frac{Y_2}{Z_2}).$$

Для вывода формул суммы будем пользоваться формулами сложения точек эллиптической кривой в аффинных координатах (1.5), (1.6), (1.7). Будем считать вначале, что $Z_1 \neq 0, \quad Z_2 \neq 0$. Выведем формулы сложения различных точек $P \neq Q$. Поскольку для различных точек $x_1 \neq x_2$, то мы можем выразить через проективные координаты коэффициент k:

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{Y_2/Z_2 - Y_1/Z_1}{X_2/Z_2 - X_1/Z_1} = \frac{Y_2Z_1 - Y_1Z_2}{X_2Z_1 - X_1Z_2}.$$

Введем следующие обозначения

$$u_1 = Y_2Z_1, \quad u_2 = Y_1Z_2, \quad u = u_1 - u_2, \quad v_1 = X_2Z_1, \quad v_2 = X_1Z_2, \quad v = v_1 - v_2.$$

В новых обозначениях имеем $k = \frac{u}{v}$ и, пользуясь (1.5), выразим в проективных координатах $x_3, \quad y_3$

$$x_3 = k^2 - x_1 - x_2 = \frac{u^2}{v^2} - \frac{X_1}{Z_1} - \frac{X_2}{Z_2} = \frac{u^2Z_1Z_2 - v^2(X_1Z_2 + X_2Z_1)}{v^2Z_1Z_2} = \frac{w}{v^2Z_1Z_2},$$

где

$$w = u^2Z_1Z_2 - v^3 - 2v^2v_2.$$

$$y_3 = k(x_1 - x_3) - y_1 = \frac{u}{v}(\frac{X_1}{Z_1} - \frac{w}{v^2Z_1Z_2}) - \frac{Y_1}{Z_1} = \frac{u(v^2X_1Z_2 - w) - v^3Y_1Z_2}{v^3Z_1Z_2}.$$

Таким образом, мы выразили сумму $P+Q$ через проективную точку

$$(\frac{w}{v^2 Z_1 Z_2}, \frac{u(v^2 X_1 Z_2 - w) - v^3 Y_1 Z_2}{v^3 Z_1 Z_2}, 1).$$

Чтобы избежать деления, выберем следующего представителя данной проективной точки:

$$(vw, u(v^2 v_2 - w) - v^3 u_2, v^3 Z_1 Z_2).$$

Перейдем к случаю $P = Q = (X, Y, Z)$. Тогда $P+Q = P+P = R = (X_3, Y_3, Z_3)$. Если $Y = 0$, то, согласно формуле (1.7), получаем $R = (0, 1, 0)$. Иначе, пользуясь формулой (1.6), выразим через проективные координаты k:

$$k = \frac{3x_1^2 + a}{2y_1} = \frac{3X^2/Z^2 + a}{2Y/Z} = \frac{3X^2 + aZ^2}{2YZ}.$$

Для удобства введем следующие обозначения

$$w = 3X^2 + aZ^2, \quad s = YZ.$$

В новых обозначениях получим

$$k = \frac{w}{2s}.$$

Теперь выразим x_3, y_3 через проективные координаты:

$$x_3 = k^2 - 2x_1 = \frac{w^2}{4s^2} - \frac{2X}{Z} = \frac{w^2 - 8sXY}{4s^2}.$$

Обозначим $t = sXY, \quad h = w^2 - 8t$, тогда получим

$$x_3 = \frac{h}{4s^2}.$$

Теперь найдем y_3

$$y_3 = k(x_1 - x_3) - y_1 = \frac{w}{2s}(\frac{X}{Z} - \frac{h}{4s^2}) - \frac{Y}{Z} = \frac{w(4t - h) - 8Y^2 s^2}{8s^3}.$$

Теперь мы можем выбрать подходящего представителя полученной проективной

точки $R = P + P$:

$$(2hs, w(4t - h) - 8Y^2s^2, 8s^3).$$

Рассмотрим наконец случай, когда $Z_1 = 0$ или $Z_2 = 0$. Пусть $P = (0, 1, 0)$, тогда $P+Q = Q$ для любой проективной точки Q, что следует из взаимно однозначного соответствия бесконечно удаленной аффинной точки O и проективной точки $(0, 1, 0)$.

1.6 Координаты Якоби

Благодаря переходу к проективным координатам, мы избежали такой затратной операции, как вычисление обратного элемента. Продолжая исследование эффективности вычисления операции сложения точек эллиптической кривой в различных системах координат, постараемся уменьшить количество умножений в алгоритме вычисления суммы. С этой целью рассмотрим эллиптическую кривую в проективных координатах Якоби.
Выполним следующую замену в уравнении (1.2):

$$x = X/Z^2, \quad y = Y/Z^3.$$

Получим уравнение эллиптической кривой в проективных координатах Якоби:

$$Y^2 = X^3 + aXZ^4 + bZ^6. \tag{1.9}$$

Аналогично ранее рассмотренной проективной плоскости, определим проективную плоскости в координатах Якоби как множество троек (X, Y, Z) не равных одновременно нулю элементов $X, Y, Z \in K$ с отношением эквивалентности:

$$(X, Y, Z) \sim (\lambda^2 X, \lambda^3 Y, \lambda Z)$$

для любых $\lambda \in K$.

1.7 Вывод формул сложения точек эллиптической кривой в координатах Якоби

Определим правила перевода точек между аффинной системой координат и системой координат Якоби. Проективная точка эллиптической кривой в координатах Якоби (X, Y, Z), отличная от бесконечно удаленной точки ($Z \neq 0$), переходит в аффинную точку с координатами $(\frac{X}{Z^2}, \frac{Y}{Z^3})$. Обратно, аффинная точка (x, y) переходит в точку с координатами $(xZ^2, yZ^3, Z) = (x, y, 1)$.
Чтобы определить бесконечно удаленную точку в проективных координатах Якоби, подставим в уравнение (1.9) $Z = 0$. Получим $Y^2 = X^3$, или, иначе говоря, $(X^2, X^3, 0) = (1, 1, 0)$ — бесконечно удаленная точка O.

Перейдем непосредственно к выводу формул сложения точек эллиптической кривой.
Пусть $P = (X_1, Y_1, Z_1), \quad Q = (X_2, Y_2, Z_2)$ — точки эллиптической кривой в координатах Якоби. Найдем координаты их суммы $P + Q = R = (X_3, Y_3, Z_3)$. Точки P, Q имеют в аффинных координатах следующий вид:

$$P = (\frac{X_1}{Z_1^2}, \frac{Y_1}{Z_1^3}), \quad Q = (\frac{X_2}{Z_2^2}, \frac{Y_2}{Z_2^3}).$$

Аналогично выводу формул сложения в проективных координатах, воспользуемся формулами сложения точек эллиптической кривой в аффинных координатах (1.5), (1.6), (1.7).
Рассмотрим вначале случай, когда $Z_1 = 0$ или $Z_2 = 0$, то есть одна из точек P, Q является бесконечно удаленной. Допустим, $P = (1, 1, 0)$, тогда $P + Q = Q$ для любой точки Q, удовлетворяющей уравнению (1.9).
Теперь будем считать, что $Z_1 \neq 0, \quad Z_2 \neq 0$. Рассмотрим различные точки P, Q. Выразим через координаты Якоби коэффициент k:

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{Y_2/Z_2^3 - Y_1/Z_1^3}{X_2/Z_2^2 - X_1/Z_1^2} = \frac{Y_2 Z_1^3 - Y_1 Z_2^3}{X_2 Z_1^3 Z_2 - X_1 Z_2^3 Z_1}.$$

Введем обозначения

$$U_1 = X_1 Z_2^2, \quad U_2 = X_2 Z_1^2, \quad H = U_2 - U_1, \quad S_1 = Y_1 Z_2^3, \quad S_2 = Y_2 Z_1^3, \quad r = S_2 - S_1.$$

В новых обозначениях получаем $k = \dfrac{r}{Z_1 Z_2 H}$ и, используя формулу (1.5), выразим в координатах Якоби x_3, y_3:

$$x_3 = k^2 - x_1 - x_2 = \frac{r^2}{Z_1^2 Z_2^2 H^2} - \frac{X_1}{Z_1^2} - \frac{X_2}{Z_2^2} = \frac{r^2 - X_1 Z_2^2 H^2 - X_2 Z_1^2 H^2}{Z_1^2 Z_2^2 H^2} =$$
$$= \frac{r^2 - U_1 H^2 - U_2 H^2}{Z_1^2 Z_2^2 H^2} = \frac{r^2 - H^2(U_2 - U_1 + 2U_1)}{Z_1^2 Z_2^2 H^2} = \frac{r^2 - H^3 - 2U_1 H^2}{Z_1^2 Z_2^2 H^2}.$$

$$y_3 = k(x_1 - x_3) - y_1 = \frac{r}{Z_1 Z_2 H}\left(\frac{X_1}{Z_1^2} - \frac{r^2 - H^3 - 2U_1 H^2}{Z_1^2 Z_2^2 H^2}\right) - \frac{Y_1}{Z_1^3} =$$
$$= \frac{r}{Z_1 Z_2 H} \frac{X_1 Z_2^2 H^2 - r^2 + H^3 + 2X_1 Z_2^2 H^2}{Z_1^2 Z_2^2 H^2} - \frac{Y_1}{Z_1^3} =$$
$$= \frac{r(2U_1 H^2 - r^2 + H^3) - S_1 H^3 + r U_1 H^2}{Z_1^3 Z_2^3 H^3}.$$

Таким образом,

$$P + Q = R = \left(\frac{r^2 - H^3 - 2U_1 H^2}{Z_1^2 Z_2^2 H^2}, \frac{r(2U_1 H^2 - r^2 + H^3) - S_1 H^3 + r U_1 H^2}{Z_1^3 Z_2^3 H^3}, 1\right).$$

В качестве представителя точки выберем следующую тройку чисел:

$$(r^2 - H^3 - 2U_1 H^2, -r(r^2 - H^3 - 2U_1 H^2) - S_1 H^3 + r U_1 H^2, Z_1 Z_2 H),$$

или, при последовательном вычислении координат

$$X_3 = r^2 - H^3 - 2U_1 H^2, \quad Y_3 = r(U_1 H^2 - X_3) - S_1 H^3, \quad Z_3 = Z_1 Z_2 H. \quad (1.10)$$

Перейдем к случаю $P = Q = (X, Y, Z)$. Тогда $P + Q = P + P = R = (X_3, Y_3, Z_3)$. Если $Y = 0$, то, согласно формуле (1.7), получаем $R = (1, 1, 0)$. Если же $Y \neq 0$, то выразим через координаты Якоби коэффициент k:

$$k = \frac{3x_1^2 + a}{2y_1} = \frac{3X^2/Z^4 + a}{2Y/Z^3} = \frac{3X^2 + aZ^4}{2YZ}.$$

Теперь выразим x_3, y_3 через координаты Якоби:

$$x_3 = k^2 - 2x_1 = \frac{(3X + aZ^4))^2}{(2YZ)^2} - \frac{2X}{Z^2} = \frac{(3X + aZ^4))^2 - 8XY^2}{(2YZ)^2}.$$

Обозначим $S = 4XY^2, \quad M = 3X + aZ^4, \quad T = M^2 - 2S$, тогда получим

$$x_3 = \frac{T}{(2YZ)^2}.$$

Теперь выразим y_3:

$$y_3 = k(x_1 - x_3) - y_1 = \frac{M}{2YZ}(\frac{X}{Z^2} - \frac{T}{(2YZ)^2}) - \frac{Y}{Z^3} =$$
$$= \frac{M}{2YZ}\frac{4XY^2 - T}{(2YZ)^2} - \frac{Y}{Z^3} = \frac{4MXY^2 - MT - 8Y^2}{(2YZ)^3} =$$
$$= \frac{M(S-T) - 8Y^2}{(2YZ)^3}.$$

Мы можем выбрать следующего представителя в качестве точки $R = P + P$ в проективных координатах Якоби:

$$(T, M(S-T) - 8Y^2, 2YZ). \tag{1.11}$$

1.8 Обобщенные координаты Якоби

Вновь обращаясь к формулам (1.10), (1.11) заметим, что алгоритмы вычисления суммы можно сделать еще более эффективными, если сохранять результаты некоторых промежуточных вычислений. Существует несколько подходящих систем координат, позволяющих более эффективно выполнять сложение точек. Такие системы координат называют модифицированными координатами Якоби. В данной работе мы рассмотрим такую систему координат, как обобщенные координаты Якоби (Chudnovsky Jacobian coordinates).

В обобщенных координатах Якоби точка представляется как пятерка чисел (X, Y, Z, Z^2, Z^3), при этом эллиптическая кривая сохраняет вид (1.9). Кроме того, точка (X, Y, Z, Z^2, Z^3) естественным образом соответствует аффинной точке $(\frac{X}{Z^2}, \frac{Y}{Z^3})$, и, обратно, аффинная точка (x, y) переходит в точку с координатами $(xZ^2, yZ^3, Z, Z^2, Z^3) = (x, y, 1, 1, 1)$.

1.9 Вывод формул сложения точек эллиптической кривой в обобщенных координатах Якоби

Пусть $P = (X_1, Y_1, Z_1, Z_1^2, Z_1^3), \quad Q = (X_2, Y_2, Z_2, Z_2^2, Z_2^3)$ — точки эллиптической кривой в обобщенных координатах Якоби. Найдем координаты их суммы $P + Q = R = (X_3, Y_3, Z_3, Z_3^2, Z_3^3)$. Переход к обобщенным координатам Якоби не меняет алгоритм сложения точек, поэтому мы можем воспользоваться полученными ранее формулами.
Рассмотрим случай, когда точки P, Q различны. Изменим вспомогательные обозначения с учетом новых координат:

$$U_1 = X_1(Z_2^2), \quad U_2 = X_2(Z_1^2), \quad H = U_2 - U_1,$$
$$S_1 = Y_1(Z_2^3), \quad S_2 = Y_2(Z_1^3), \quad r = S_2 - S_1.$$

Получим координаты искомой точки напрямую из формулы (1.10)

$$X_3 = r^2 - H^3 - 2U_1H^2, \quad Y_3 = r(U_1H^2 - X_3) - S_1H^3,$$
$$Z_3 = Z_1Z_2H, \quad Z_3^2 = Z_3^2, \quad Z_3^3 = Z_3^3$$

Рассмотрим теперь случай равенства точек $P = Q = (X, Y, Z, Z^2, Z^3)$. Введем вспомогательные обозначения:

$$S = 4XY^2, \quad M = 3X + a(Z^2)^2, \quad T = M^2 - 2S$$

Пользуясь формулой (1.11), получаем координаты искомой точки $P + P = R = (X_3, Y_3, Z_3, Z_3^2, Z_3^3)$:

$$X_3 = T, \quad Y_3 = M(S - T) - 8Y^2, \quad Z_3 = 2YZ, \quad Z_3^2 = Z_3^2, \quad Z_3^3 = Z_3^3$$

Легко заметить, что дополнительные координаты упрощают процесс вычисления суммы различных точек, но в случае удвоения точки эффективность перехода к обобщенным координатам Якоби уже не так очевидна. Мы подробнее рассмотрим вопросы эффективности алгоритмов сложения и удвоения точек во всех ранее рассмотренных системах координат в следующей главе данной курсовой работы, а именно: проведем теоретический анализ эффективности изучаемых алгоритмов, реализуем описанные алгоритмы сложения точек эллипти-

ческой кривой в различных системах координат в Python, а также оценим на практике время работы реализованных алгоритмов.

ГЛАВА 2

Реализация и анализ эффективности алгоритмов

2.1 Описание реализации алгоритмов сложения и удвоения точек эллиптической кривой

Перейдем к реализации на языке Python рассмотренных в Главе 1 алгоритмов. Для каждой из рассматриваемых систем координат определим классы, представляющий эллиптическую кривую, а также точку на эллиптической кривой в данной системе координат. Реализованные нами классы будут содержать не только методы, связанные непосредственно со сложением точек эллиптических кривых, но и дополнительные методы, позволяющие более полно описать такие математические объекты, как эллиптические кривые и точки на эллиптических кривых. Для каждого класса определим основные методы, такие как метод инициализации экземпляра класса и метод строкового представления, перегрузим операцию сравнения `==`, а для классов, представляющих точки на эллиптических кривых, также перегрузим арифметические операции `+`, `-`, `*`. Кроме перечисленных, для каждого класса определим методы перевода точек и кривых между различными системами координат, а для классов, представляющих эллиптические кривые, напишем методы нахождения всех точек кривой, случайной точки на кривой, а также метод, определяющий, принадлежит ли точка с заданными координатами кривой.

Остановимся подробнее на аффинной системе координат. Реализуем классы Curve, Point, infPoint соответсвующие эллиптической кривой в короткой форме Вейерштрасса (1.2), конечной аффинной точке на кривой и бесконечно удаленной точке.

Опишем класс Curve: поскольку кривая вида (1.2) однозначно определяется числами $a, b \in K$, где K - поле порядка p, то экземлпляры класса Curve должны содержать соответсвующие атрибуты a, b, p. Пользуясь однозначностью представления кривой в поле K, перегрузим также операцию сравнения `==`. Кроме того, определим метод `findPointsShanks` для нахождения всех точек кривой. Данный метод используюет алгоритм Тонелли-Шенкса нахождения квадратного

корня по простому модулю. Для дальнейшего тестирования алгоритмов сложения на конкретных кривых нам будет полезно определить метод `findRandomPoint`, возвращающий случайную точку, принадлежащую данной кривой. Также добавим метод `isPoint`, определяющий, принадлежит ли точка с заданными координатами (x, y) данной эллиптической кривой. Этот метод будет полезен при проверке корретности работы алгоритма сложения точек. Наконец, определим методы `toProjective`, `toJacobian`, `toChudnovskyJacobian` перевода кривой из аффинных координат в проективные координаты, координаты Якоби и обобщенные координаты Якоби соответсвенно.

Перейдем к классу Point. Определим следующие атрибуты класса: $curve$ - экземпляр класса Curve, эллиптическая кривая, которой принадлежит данная точка, x, y - координаты точки в аффинной системе координат, причем $x, y \in \mathbb{Z}p$, где p - простое, $p > 3$ - порядок поля K, заданный для кривой $curve$. Далее перегрузим методы строкового представления и операцию сравнения. Пользуясь тем фактом, что для точки P противоположным элементом являестя точка является точка $-P$, полученная отражением точки P относительно горизонтальной оси координат, перегрузим унарный минус. Теперь перейдем к перегрузке арифметических операций `+`, `-`, `*`. Определим операцию сложения согласно описанному нами в разделе 1.3 алгоритму, в частности, используем формулы (1.5), (1.6), (1.7). Фрагмент алгоритма, соответсвующий удвоению точки, выделим в отдельную функцию `doubling`, чтобы иметь в дальнейшем возможность оценить время выполнения алгоритмов сложения и удвоения точек по отдельности. Теперь перегрузим операцию вычитания как сложение с противоположным элементом. Операцию `*` перегрузим для вычисления результата умножения точки эллиптической кривой на скаляр. Умножение точки P на целое число m представляет собой сумму m точек P, и сам алгоритм можно определить по аналогии с алгоритмом быстрого возведения в степень, изучаемым в курсе «Математические основы защиты информации». Наконец, пользуясь ранее описанными правилами перевода точек, определим методы `toProjective`, `toJacobian`, `toChudnovskyJacobian` перевода точки из аффинных координат в проективные координаты, координаты Якоби и обобщенные координаты Якоби соответсвенно.

Кроме того, опишем класс infPoint, представляющий бесконечно удаленную точ-

ку на кривой. Данный класс будет иметь единственный атрибут `curve` - кривая, которой принадлежит данная бесконечная точка. Для данного класса определим те же методы, что и для класса Point. Необходимость создания отдельного класса для бесконечно удаленной точки вызвана тем, что для бесконечно удаленной аффинной точки нельзя задать конечные координаты. Кроме того, на мой взгляд, удобнее не прописывать в каждом методе класса Point особый случай для бесконечной точки, а оформить все необходимые методы для отдельного класса.

Отметим сразу, что из-за особенностей языка Python возведение числа x в степень с помощью встроенной функции `pow` медленнее, чем последовательное умножение числа x на себя, поэтому возведение по модулю p, например, в квадрат, будем записывать как `x*x % p` вместо варианта `pow(x,2,p)`. Кроме того, если в алгоритах сложения или удвоения требуется повторно возводить число в некоторую степень, то мы предварительно вычислим данную величину и будем далее использовать полученный результат.

Так, мы описали атрибуты и методы для классов, представляющих эллиптическую кривую и точку в аффинной системе координат. Для остальных интересующих нас систем координат будем определять по 2 класса: эллиптическая кривая в заданной системе координат и точка на данной кривой. Такие классы будут иметь те же методы и атрибуты, что и описанные ранее классы Curve и Point, поэтому рассмотрим подробнее только существенные отличия и особенности реализации классов кривой и точки для проективных координат, координат Якоби и обобщенных координат Якоби.

Для проективных координат определим классы ProjectiveCurve, ProjectivePoint. Особенность реализации класса ProjectiveCurve состоит в том, что методы `findPoints` и `findRandomPoint` не используют алгоритм Тонелли-Шенкса, поскольку в противном случае необходимо было бы искать обратный элемент к Z для того, чтобы из формулы (1.8) выразить Y^2, но сам переход к проективным координатам мы осуществляли ради того, чтобы не выполнять такую медленную операцию, как вычисление обратного элемента. Также отметим, что для всех классов, представляющих кривую и точку в системах координат, помимо аффинной, не имеет смысла делать методы перевода между проективными координатами, координатами Якоби, обобщенными координатами Якоби.

Для таких классов определим лишь методы перевода кривых и точек в аффинную систему координат. Класс ProjectivePoint, кроме атрибутов, определенных для класса Point будет содержать атрибут z - третью координату точки. Особенность реализации данного класса будет содержаться в перегрузке операции ==, поскольку для проективных точек вводится отношение эквивалентности, описанное в разделе 1.4. Для эквивалентных в проективной плоскости точек (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) верно

$$\frac{X_1}{X_2} = \frac{Y_1}{Y_2} = \frac{Z_1}{Z_2} = \lambda,$$

откуда

$$\frac{X_1}{X_2} = \frac{Z_1}{Z_2}, \quad \frac{Y_1}{Y_2} = \frac{Z_1}{Z_2},$$

таким образом, достаточно проверить выполнение условий

$$X_1 Z_2 = X_2 Z_1, \quad Y_1 Z_2 = Y_2 Z_1,$$

— такие точки (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) будут считаться равными в проективной плоскости. Алгоритм сложения, реализованный при перегрузке операции +, основывается на формулах сложения и удвоения точек, полученных в разделе 1.5. Отметим также, что в системах координат, отличных от аффинных, бесконечно удаленная точка будет иметь конечные координаты, поэтому отдельный класс для ее представления создавать не нужно.

Перейдем к проективным координатам Якоби. Для представления эллиптической кривой и точки в данной системе координат определим классы JacobianCurve и JacobianPoint. Для класса JacobianPoint операция сравнения перегружается согласно введенному в разделе 1.6 отношению эквивалентности, то есть

$$\frac{X_1}{X_2} = \lambda^2, \quad \frac{Y_1}{Y_2} = \lambda^3, \quad \frac{Z_1}{Z_2} = \lambda$$

для эквивалентных точек (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), откуда

$$\frac{X_1}{X_2} = \frac{Z_1^2}{Z_2^2}, \quad \frac{Y_1}{Y_2} = \frac{Z_1^3}{Z_2^3},$$

таким образом, для определения равенства точек в координатах Якоби необхо-

димо проверить выполнение условий

$$X_1 Z_2^2 = X_2 Z_1^2, \quad Y_1 Z_2^3 = Y_2 Z_1^3. \tag{2.1}$$

Для реализации алгоритмов сложения и удвоения точек в координатах Якоби воспользуемся формулами, полученными в разделе 1.7, в частности, (1.10), (1.11).

Сделаем важное замечание: при реализации алгоритма сложения двух точек необходимо разделять случаи, когда точки различны или равны между собой. В алгоритме данный шаг подразумевает использование перегруженной операции сравнения ==. Как видно из полученных нами условий проверки равенства точек (2.1), данная операция потребует возведения чисел Z_1, Z_2 в степени, что также требуется в самом алгоритме сложения точек. Чтобы не замедлять алгоритм необходимостью дважды вычислять одни и те же величины, предварительно вычислим $Z_1^2, Z_1^3, Z_2^2, Z_2^3$ и проведем проверку равенства точек внутри алгоритма сложения, без вызова операции == для объектов класса JacobianPoint.

Далее, определим классы ChudnovskyJacobianCurve и ChudnovskyJacobianPoint для представления эллиптической кривой и точки в обобщенных координатах Якоби. Поскольку точка в данной системе координат выражается пятеркой чисел, зададим дополнительные атрибуты для класса ChudnovskyJacobianPoint, такие как zz, zzz, представляющие величины Z^2, Z^3 соответственно. Условия проверки равенства точек в обобщенных координатах Якоби точно такие же, как в проективных координатах Якоби (2.1), отличие в реализации будет заключаться лишь в том, что координату z не нужно будет дополнительно возводить в квадрат и куб, поскольку нужные величины уже хранятся в атрибутах zz, zzz соответственно. Перегрузим операцию +, пользуясь формулами, полученными в разделе 1.9, и, по аналогии с классами ProjectivePoint, JacobianPoint, фрагмент кода, соответствующий удвоению точки, продублируем в отдельной функции `doubling`. Различия в классах ChudnovskyJacobianCurve и JacobianCurve обусловлены лишь тем, что обобщенные координаты Якоби позволяют избавиться от некоторых возведений в степень, используя предватительно вычисленные величины Z^2, Z^3, в остальном методы данных классов не различаются.

Мы описали классы, с помощью которых представили на языке Python такие математические объекты, как эллиптические кривые и точки эллиптических

кривых в различных системах координат. Перечислим функции и пакеты, которые нам понадобились при реализации данных классов, либо при тестировании их работы:

- Для реализации алгоритмов: функция `Jacobi` для вычисления символа Якоби, функция `Shanks`, реализующая алгоритм Тонелли-Шенкса нахождения квадратного корня по простому модулю, функция st для представления числа в виде $2^s t$, а также вспомогательная функция `D`, вычисляющая дискриминант кривой.

- Для тестирования алгоритмов: функция генерации случайного целого числа `randint` из модуля `random`, функция генерации случайного простого числа `randprime` из библиотеки `sympy`, модуль `time`.

Прежде, чем приступить к анализу эффективности алгоритмов сложения и удвоения, проверим корректность работы определенных нами методов классов, в частности, проверим правильность работы методов перевода точек и кривых между системами координат, умножения точек эллиптических кривых на скаляр, а также проведем тесты на ассоциативность и коммутативность умножения: согласно Теореме 1 данные свойста обязаны выполняться. Реализация классов на языке Python, а также выполненные тесты приведены в приложении.

2.2 Теоретический анализ эффективности алгоритмов сложения и удвоения точек эллиптической кривой

Чтобы провести теоретический анализ эффективности реализованных алгоритмов сложения и удвоения точек эллиптической кривой в различных системах координат, оценим количество арифметических операций, которые выполняются в каждом из алгоритмов. Мы будем подсчитывать следующие типы операций: I - операция вычисления обратного элемента, M - умножение чисел, которые мы полагаем различными, S - возведение числа в квадрат, которое, согласно замечанию, сделанному в предыдущем разделе, в нашей реализации представляет собой умножение числа на себя, C - умножение числа на константу, меньшую 10, A - сложение/вычитание. Необходимо разделять операции умножения и возведения в степень, поскольку Python выполняет умножение числа на себя быстрее,

чем умножение различных чисел того же порядка. Также отметим, что, хотя мы и подсчитываем количество операций типов C, A, но выполняются они значительно быстрее операций I, M, S и большого влияния на скорость выполнения алгоритма, по сути, не имеют.

В аффинных координатах алгоритм удвоения точки эллиптической кривой выполняет $I + 2M + 2S + 3C + 3A$ операций, а алгоритм сложения различных точек выполняет $I + 2M + S + 6A$ операций.

Перейдем к проективным координатам. Отметим сразу, что поскольку необходимо разделять случаи, когда точки различны или равны между собой, то при подсчете числа операций нужно учитывать также арифметические операции, выполняемые при сравнении двух точек. Так, в проективных координатах при сравнении точек выполняется $4M + 2A$ операций, а непосредственно алгоритмы удвоения и сложения выполняют $8M+5S+5C+4A$ и $17M+2S+C+6A$ операций соответственно.

Для координат Якоби операция сравнения выполняет $6M + 2S + 2A$ операций, алгоритм удвоения вычисляет $4M + 5S + 5C + 4A$ операций, а алгоритм сложения — $12M + 2S + C + 6A$ операций.

Наконец, в обобщенных координатах Якоби сумма равных точек вычисляется за $5M + 6S + 5C + 4A$, а различных точек — за $12M + 3S + C + 5A$ операций. Однако операция сравнения выполняет всего $4M + 2A$ операций благодаря дополнительным координатам, в которых хранятся необходимые ранее вычисленные величины.

Для того, чтобы сравнить число арифметических операций, выполняемых алгоритмами сложения и удвоения точки в различных системах координат, выведем полученные результаты в виде таблицы. Как уже было сказано, операции сложения и умножения на константу практически не влияют на скорость алгоритма, поэтому опустим данные типы операций. Также, поскольку перед тем, как алгоритм перейдет к удвоению либо сложению точек, выполняется их сравнение, добавим арифметические операции, выполняемые при сравнении точек к общему числу операций в соответствующих алгоритмах сложения и удвоения.

Количество арифметических операций		
Система координат	Удвоение точки	Сложение точек
Аффинные координаты	$I + 2M + 2S$	$I + 2M + S$
Проективные координаты	$12M + 5S$	$21M + 2S$
Координаты Якоби	$10M + 7S$	$18M + 4S$
Обобщенные координаты Якоби	$9M + 6S$	$16M + 3S$

Известно, что операция поиска обратного элемента значительно медленнее операции умножения, поэтому алгоритмы в аффинных координатах заведомо уступают в скорости алгоритмам в других системах координат. В остальном, можно ожидать, что на практике быстрее всего себя покажет алгоритм, выполняющий сложение и удвоение точек в обобщенных координатах Якоби, однако остаются открытыми вопросы, будет ли наше предположение верным для всех точек некоторой заданной кривой, как в действительности арифметическая сложность влияет на время выполнения операций, а также, насколько эффективно выполнение операции сложения в системах координат, отличных от аффинной, с последующим переводом точки обратно.

2.3 Практический анализ эффективности алгоритмов сложения и удвоения точек эллиптической кривой

Для того, чтобы оценить время работы реализованных алгоритмов, вначале протестируем алгоритмы удвоения и сложения точек по отдельности и сравним полученные результаты в различных системах координат. Для этого определим вспомогательные функции `doubl` и `addition`, последовательно вызывающие метод `doubling` и операцию `+` для заданных точек соответственно. Конечно, мы не можем гарантировать, что складываться будут только лишь различные точки, но мы можем уменьшить вероятность такого события, выбрав кривую в поле достаточно большого порядка. А именно, сложение точек мы будем выполнять на эллиптической кривой SECP256k1, которая используется в протоколах системы Bitcoin. Такая кривая рассматривается над полем порядка $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, а константы a, b равны 0 и 7 соответсвенно. Все тесты будут проходить следующим образом: после задания кривой SECP256k1 в аффинных координатах мы будем выбирать на ней

случайную аффинную точку, а затем получим еще 3 три точки с помощью перевода исходной в проективные координаты, координаты Якоби и обобщенные координаты Якоби. Далее операции будут выполняться в соответствующих системах координат. Отметим также, что полученные после выполнения операций в различных координатах точки, мы будем переводить обратно в аффинные координаты. Поскольку мы оцениваем эффективность использования различных систем координат с учетом того, что конечный результат мы должны получить в аффинных координатах, то перевод точки обратно также должен учитываться при оценивании скорости работы алгоритмов. Кроме того, проще проверить корректность полученных результатов, сравнивая точки в одной системе координат.

Проведя множество тестов для определения скорости выполнения алгоритма удвоения над различными случайными точками кривой, можно сделать следующие выводы:

- Скорость работы алгоритма удвоения значительно зависит от выбора точки, полученные для различных точек результаты измерений могут различаться на 20%.

- Система координат, показывающая самый быстрый результат также может меняться в зависимости от выбора точки. Чаще всего самым быстрым оказывалось удвоение в проективных координатах Якоби. Вероятно, это связано с тем, что функция `doubling` вызывается для заведомо равных точек, то есть проверка на равенство не выполняется, а как мы выяснили в предыдущем разделе, именно у координат Якоби самая медленная операция сравнения.

- Соотношение между скоростью удвоения точки в аффинных координатах и вторым по скорости результатом сохраняется: в аффинных координатах сумма равных точек считается примерно в 4 раза медленнее.

Приведем один из полученных результатов оценки скорости выполнения алгоритма удвоения точки эллиптической кривой, отметим, что для более точного результата, в данном тесте проводилось 2^{19} удвоения.

```
Аффинные координаты:--- 22.50896120071411 seconds ---
```

```
Проективные координаты:--- 5.310141324996948 seconds ---
Координаты Якоби:--- 5.354511976242065 seconds ---
Обобщенные координаты Якоби:--- 5.073979139328003 seconds ---
```

Далее, протестируем алгоритм сложения различных точек и сделаем некоторые выводы о его эффективности:

- Скорость выполнения алгоритма сложения в меньшей степени зависит от выбора исходных точек, полученное время лишь незначительно менялось от теста к тесту.

- Для алгоритма сложения прослеживается явная градация: быстрее всего выполняется сложение в обобщенных координатах Якоби, затем в проективных координатах Якоби, на третьем месте по скорости проективные координаты, и, наконец, медленнее всего считается сумма точек в аффинной системе координат. Такое расположение результатов не менялось в различных тестах.

- Соотношение между временем вычисления операции сложения точек в аффинных координатах и проективных координатах составляло примерно $2:1$ и практически не изменялось при тестировании на различных точках.

- При различных измерениях также сохранялось соотношение между временем вычисления суммы в проективных координатах и координатах Якоби, а также между координатами Якоби и обобщенными координатами Якоби: в обоих случаях разница составила около 15%

Приведем результаты одного измерения времени выполнения алгоритма сложения точек эллиптической кривой. В данном тесте проводилось 2^{19} сложения.

```
Аффинные координаты:--- 20.042898654937744 seconds ---
Проективные координаты:--- 10.82426643371582 seconds ---
Координаты Якоби:--- 9.010782241821289 seconds ---
Обобщенные координаты Якоби:--- 8.495838403701782 seconds ---
```

Во многих криптографических протоколах и алгоритмах ключевой является операция умножения точки эллиптической кривой на скаляр. Мы также реализовали данную операцию для классов точек во всех рассматриваемых системах

координат с помощью алгоритма, выполняющего последовательное сложение точек. Из-за широкого применения в криптографии операции скалярного умножения, будет важно оценить эффективность выполнения данной операции, более того, при внимательном рассмотрении алгоритма скалярного умножения, можно заметить, что на каждой итерации выполняется как сложение различных точек, так и удвоение точки. Таким образом, данная операция объединяет две составляющие алгоритма сложения точек эллиптической кривой: сложение различных точек и удвоение точки.
В ходе оценивания скорости выполнения операции скалярного умножения мы как многократно умножали точки в различных системах координат на фиксированное число, так и генерировали случайные значения скаляров, различные для всех точек. Проведя множество тестов, можно сделать некоторые выводы:

- От теста к тесту мы получали стабильные результаты, практически не менялось как соотношение между получаемыми измерениями, так и очередность: медленнее всего выполнялось умножение в аффинных координатах, следом идут проективные координаты, затем координаты Якоби, и наконец, быстрее всего вычисления происходили в обобщенных координатах Якоби.

- В большинсве случаев сохранялось соотношение между временем вычисления операции в проективных координатах и координатах Якоби, а также между координатами Якоби и обобщенными координатами Якоби: в обоих случаях разница вновь составляла порядка 15%.

В целом, результаты оценивания скорости выполнения операции скалярного умножения схожи с результатами выполнения операции сложения различных точек. Приведем один из результатов измерений: случай, когда для каждой точки мы генерировали случайные числа, на которые производилось умножение.

```
Аффинные координаты:--- 3.001615524291992 seconds ---
Проективные координаты:--- 1.9091746807098389 seconds ---
Координаты Якоби:--- 1.297048807144165 seconds ---
Обобщенные координаты Якоби:--- 1.173142433166504 seconds ---
```

Как видно, полученные результаты практического анализа эффективности алгоритмов сложения соответствуют результатам теоретического анализа алгоритмов. Можно утверждать, что переход от аффинной системы координат

к проективным координатам, координатам Якоби и обобщенным координатам Якоби целесообразен, ведь в любой из данных координат сложение выполняется значительно быстрее: как минимум в 2 раза быстрее оказались проективные координаты, еще быстрее координаты Якоби, а время, затрачиваемое на вычисление операции сложения в обобщенных координатах Якоби и вовсе может составить треть от того времени, которое мы тратим при вычислении суммы в аффинных координатах.

ЗАКЛЮЧЕНИЕ

В ходе работы был рассмотрен и описан предмет исследования, эллиптическая кривая, а также процесс сложения точек на эллиптической кривой с геометрической точки зрения. Также был проведен подробный вывод формул сложения и удвоения точек эллиптической кривой в проективных координатах, координатах Якоби, обобщенных координатах Якоби с описанием особенностей рассматриваемых систем координат. Результатом работы стали реализованные в Python классы, соответсвующие эллиптическим кривым и точкам в рассматриваемых системах координат, содержащие ряд методов для более полного описания таких математических объектов, как эллиптические кривые и точки на эллиптических кривых. В частности, были реализованы алгоритмы сложения точек эллиптической кривой в конечном поле в обычных координатах, проективных координатах, координатах Якоби и обобщенных координатах Якоби, а также был проведен теоретический анализ эффективности реализованных алгоритмов. Кроме того, был проведен практический анализ эффективности изучаемых алгоритмов путем оценивания времени работы реализованных алгоритмов. В итоге, мы сформулировали выводы об эффективности использования различных систем координат при вычислении операции сложения точек эллиптической кривой.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

[1] Elliptic Curves from Mordell to Diophantus and Back [Электронный ресурс] – Режим доступа: https://dokumen.tips/documents/elliptic-curves-from-mordell-to-diophantus-and-curves-from-mordell-to-diophantus.html?page=2

[2] Why Ellipses Are Not Elliptic Curves [Электронный ресурс]– Режим доступа: https://www.maa.org/sites/default/files/pdf/upload_library/2/Rice-2013.pdf

[3] Смарт, Н. Криптография / Н. Смарт ; пер. с англ. С. А. Кулешова под ред. С. К. Ландо. - Москва : Техносфера, 2006.

[4] Elliptic Curves: Number Theory and Cryptography[Электронный ресурс]– Режим доступа: https://people.cs.nctu.edu.tw/~rjchen/ECC2012S/Elliptic%20Curves%20Number%20Theory%20And%20Cryptography%202n.pdf

[5] В. В. Острик, М. А. Цфасман. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые. — М.: МЦНМО, 2001. — С. 20—24. — 48 с. — (Математическое просвещение) [Электронный ресурс]– Режим доступа: https://www.mccme.ru/mmmf-lectures/books/books/book.8.pdf

[6] Кононов, С.Г. Аналитическая геометрия: учеб. пособие для студ. учреждений высш. образования по математическим спец. / С. Г. Кононов; БГУ. - Минск: БГУ, 2014.

[7] Cohen, H. Efficient Elliptic Curve Exponentiation Using Mixed Coordinates / Henri Cohen, Atsuko Miyaji, Takatoshi Ono // International Conference on the Theory and Application of Cryptology and Information Security. – 1998. – P. 51-65.

ПРИЛОЖЕНИЕ

Реализация алгоритмов, проводимые тесты

Режим доступа: https://github.com/ksprski/coursework2023.git