МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра компьютерной алгебры и теории чисел

Криптография на эллиптических кривых

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 227 группы

направление 02.04.01 — Математика и компьютерные науки

механико-математического факультета

Зорикова Егора Владиславовича

Научный руководитель
доцент, к.ф.-м.н., доцент _____ В.В. Кривобок

Зав. кафедрой
зав. каф., к.ф.-м.н., доцент _____ А.М. Водолазов

Саратов 2022

Введение

Тема. Тема данной магистерской работы – "Криптография на эллиптических кривых".

Актуальность темы исследования. Криптография сегодня важна как никогда ранее. В прошлом веке беспрецедентными темпами развивались вычислительные мощности и компьютерные науки. Поэтому методы шифрования, которые считались надежными еще сто лет назад, в данный момент не являются криптостойкими, то есть потенциально могут быть взломаны и поэтому не могут рассматриваться как надежные. Эллиптические же кривые позволяют строить надежные и криптостойкие алгоритмы даже с учетом современных вычислительных ресурсов, дотупных человечеству.

Криптография – достаточно широкая область, которая находит свое применение в таких задачах, как: электронные подписи, шифрование текста, гарантия целостности данных, конфидециальность. Существует множество различных алгоритмов, каждый из которых может более оптимально подходить для конкретной задачи.

Цели и задачи. Основными целями и задачами работы являются исследование и обоснование корректности алгоритмов криптографии на эллиптических кривых, а также сравнение этих алгоритмов с известными аналогами. В работе также имеется практическая часть – реализация базовых операций на эллиптических кривых. Написание программы было выполнено с помощью языка программирования Python, который благодаря своей компактности и простому синтаксису активно используется для прототипирования математических задач.

В работе выведены правила и закономерности работы с эллиптическими кривыми. На языке Python самостоятельно реализованы различные способы задания эллиптических кривых и следующие базовые операции на них.

1. Сложение точек на эллиптической кривой.
2. Удвоение точек на эллиптической кривой.
3. Обработка краевого случая с точкой в бесконечности.

Содержание работы. Работа состоит из 5 разделов. Первый раздел посвящен необходимым понятиям и сущностям из абстрактной алгебры и тео-

рии чисел. Во втором разделе разбирается алгебраическая геометрия и уравнения, частным случаем которых является эллиптическая кривая. В третьем разделе дается описание непосредстенно самой теории эллиптических кривых: способ их задания, формальные определения, свойства. Также в этом разделе приводятся математические выкладки сложения точек на эллиптической кривой. В четвертом разделе описано краткое знакомство с языком программирования Python, представлена программная реализация сложения точек на эллиптической кривой, а также рассмотрен краевой случай для особой точки. В пятом разделе разбираются способы и методы современной криптографии. Рассматриваются популярные алгоритмы шифрования и приводятся их аналоги, реализованные с помощью эллиптических кривых.

Методы исследования. Поставленные цели были достигнуты с помощью применения математического аппарата, необходимого для теории эллиптических кривых и криптографии на них: поля с выведенными на них свойствами операций, дискретное логарифмирование и сравнительный анализ криптосистем.

Апробация. Доклад по теме исследования был представлен на кафедре компьютерной алгебры и теории чисел.

Основное содержание работы

Для начала необходимо ввести понятие поле. Пусть F – множество, на котором заданы две операции, называемые сложением и умножением и которое содержит два выделенных элемента 0 и e, причем $0 \neq e$. Далее, поле F – абелева группа по сложению, единичным элементом которой является 0, а элементы из F, отличные от 0, образуют абелеву группу по умножению, единичным элементом которой является e. Две операции, сложение и умножение, связаны законом дистрибутивности $a(b+c) = ab + ac$. Второй закон дистрибутивности $(b+c)a = ba+ca$ выполняется автоматически в силу коммутативности умножения [1]. Элемент 0 называется нулевым элементом или просто нулем, а e – единичным элементом или просто единицей поля F. Иногда единицу обозначают символом 1.

Можно привести следующие примеры полей, которые являются основными в теории чисел и множестве других областей математики:

1. Поле $\mathbb{Q}$, состоящее из всех рациональных чисел;
2. Поле $\mathbb{R}$, состоящее из всех вещественных чисел;
3. Поле $\mathbb{C}$ комплексных чисел;
4. Поле $\mathbb{Z}/p\mathbb{Z}$ классов вычетов целых чисел по модулю простого числа p.

Под конечным полем или полем Галуа понимается поле с конечным числом элементов. Это число называется порядком поля [2]. Примером конечного поля является простое поле, состоящее из n элементов, то есть поле классов вычетов по модулю n.

Также вводится понятие отношения сравнимости. Пусть $m \geq 2$ – целое число. Два целых числа a и b называются сравнимыми по модулю m, если $m|(a-b)$, то есть если их разность делится на m. Число m называется модулем сравнения [3].

Отношение сравнимости обозначается символом $a \equiv b(mod\ m)$. Каждое целое число a представимо в виде

$$a = mq + r,\ 0 \leq r < m,$$

с целыми q, r. Отсюда следует, что a и r лежат в одном классе вычетов и, значит, целый класс вычетов содрежит целое число r из промежутка $0 \leq r < m$.

Таким образом, количество классов вычетов по модулю m не превосходит m. С другой стороны, каждое из чисел $0, 1, 2, ..., m-1$ лежит в некотором классе вычетов и классы эти различны, ведь разность двух чисел из указанного списка не делится на m. Это доказывает, что существует ровно m классов вычетов по модулю m. Каждый из них содержит единственное число r из промежутка $0 \leq r < m$, называемое наименьшим неотрицательным вычетом класса, и потому

$$\bar{0}, \bar{1}, ..., \overline{m-1}$$

– все классы вычетов по модулю m.

На множестве классов вычетов по модулю m можно ввести операции сложения, вычитания и умножения по правилам

$$\bar{a} + \bar{b} = \overline{a+b};\ \bar{a} - \bar{b} = \overline{a-b};\ \bar{a}\bar{b} = \overline{ab}.$$

Перед тем, как ввести понятие эллиптической кривой, необходимо рассмотреть и ввести основные понятия и законы из алгебраической геометрии. Пусть K – поле. Аффинным n-мерным пространством $A^n(K)$ над K называется множество точек $P = (x_1, ..., x_n) \in K^n$; значения $x_1, ..., x_n$ называются аффинными координатами. В случае $n = 2$ аффинное пространство называется аффинной плоскостью, а алгебраическое многообразие, идеал которого образован одиночным полиномом, – плоской алгебраической кривой. Степень полинома, задающего кривую, называют степенью кривой [4].

Проективной плоскостью $P^2(K)$ над полем K называется множество троек $\{(X, Y, Z) \in K^3/(0, 0, 0)\}$ с эквивалентностью $(X, Y, Z) \sim (aX, aY, aZ)$, если $a \in K^*$. Здесь X, Y, Z – проективные координаты. Проективной прямой $P^1(K)$ над полем K называется множество пар $\{(X, Y) \in K^2/(0, 0)\}$ с эквивалентностью $(X, Y) \sim (aX, aY)$, если $a \in K^*$. Если $y \neq 0$, то каждая точка $(X, Y) \in P^l(K)$ эквивалентна точке $(x, 1)$ аффинной прямой. Но на проективной прямой есть точка вида $(X, 0)$, не соответствубщая никакой аффинной точке. Эта точка P_∞ называется бесконечно удаленной [5]. Поэтому

$$P^1(K) = A^1(K) \bigcup \{P_\infty\}.$$

Плоской аффинной или кубической кривой (кубикой) над полем K называется алгебраическая кривая $C(K)$, заданная полиномом $\sum_{i,j} a_i x^i y^j$, где $a \in K$ и наибольшее значение $i + j$ равно 3.

Эллиптической кривой называют неособую кубическую кривую.

Существует несколько видов записей уравнений эллиптической кривой [6]. Одна из самых распространенных форм записей – нормальная форма – получается путем линейной замены проективных переменных. Большинство форм записей представляют из себя частный случай обобщенной формы Вейерштрасса, содержащей единственную бесконечно удаленную точку, которая является точкой перегиба, а бесконечно удаленная прямая есть касательная в этой точке.

Спреведливы следующие утверждения:

1. Уравнение кубики над произвольным полем может быть сведено к обобщенной форме Вейерштрасса

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

2. Если характеристика поля отлична от 2, то это уравнение с помощью преобразований может быть представленно в виде

$$y^2 = x^3 + b_2x^2 + b_4x + b_6,$$

а над алгебраически замкнутым полем – к виду $E[a, b]$

$$y^2 = x^3 + ax^2 + bx$$

или к форме Лежандра

$$y^2 = x(x - 1)(x - \lambda).$$

3. Если характеристика поля отлична от 3, то уравнение от обобщенной формы Вейерштрасса с помощью преобразований может быть сведено к виду

$$y^2 + a_1xy + a_3y = x^3 + a_4x + a_6,$$

а над алгебраически замнутым полем – к нормальной форме Дойринга

$$y^2 + \alpha xy + y = x^3.$$

4. Если зарактеристика поля отлична от 2 и 3, то уравнение от обобщенной формы Вейерштрасса с помощью преобразований может быть сведено к виду

$$y^2 = x^3 + Ax + B.$$

Точка $(x, y) \in l$ лежит на эллиптической кривой тогда и только тогда, когда $(mx + b)^2 = f(x) = 4x^3 - g_2x - g_3$, то есть тогда и только тогда, когда x есть корень кубического полинома $f(x) - (mx + b)^2$. Этот полином имеет три корня, каждый из которых дает точку пересечения [7]. Если x – двойной или тройной корень, то l пересекает кривую с кратностью 2 или 3 в точке (x, y). В любом случае полное число точек пересечения с учетом кратностей равно 3.

Пусть $\tilde{F}(x, y, z)$ и $\tilde{G}(x, y, z)$ – однородные полиномы степеней m и n соответственно над алгебраически замкнутым полем K. Предположим, что $\tilde{F}$ и $\tilde{G}$ не имеют общего полиномиального множителя. Тогда кривые в $\mathbb{P}^2_{\mathbb{C}}$, определенные полиномами $\tilde{F}$ и $\tilde{G}$, имеют mn точек пересечения с учетом кратностей [8].

Если $P_1 + P_2 = P_3$, то $-P_3$ есть третья точка пересечения прямой $l = \overline{P_1P_2}$ с эллиптической кривой. Если $P_1 = P_2$, то под $\overline{P_1P_2}$ подразумевается касательная прямая в точке P_1. Данное утверждение проиллюстрировано в соответствии с рисунком 1, на котором изображена группа точек эллиптической кривой $y^2 = x^3 - x$. Для того чтобы сложить две точки P_1 и P_2, проводится соединяющая их прямая, находится третья точка пересечения этой прямой с кривой и берется симметричная точка, лежащая по другую сторону от оси x.

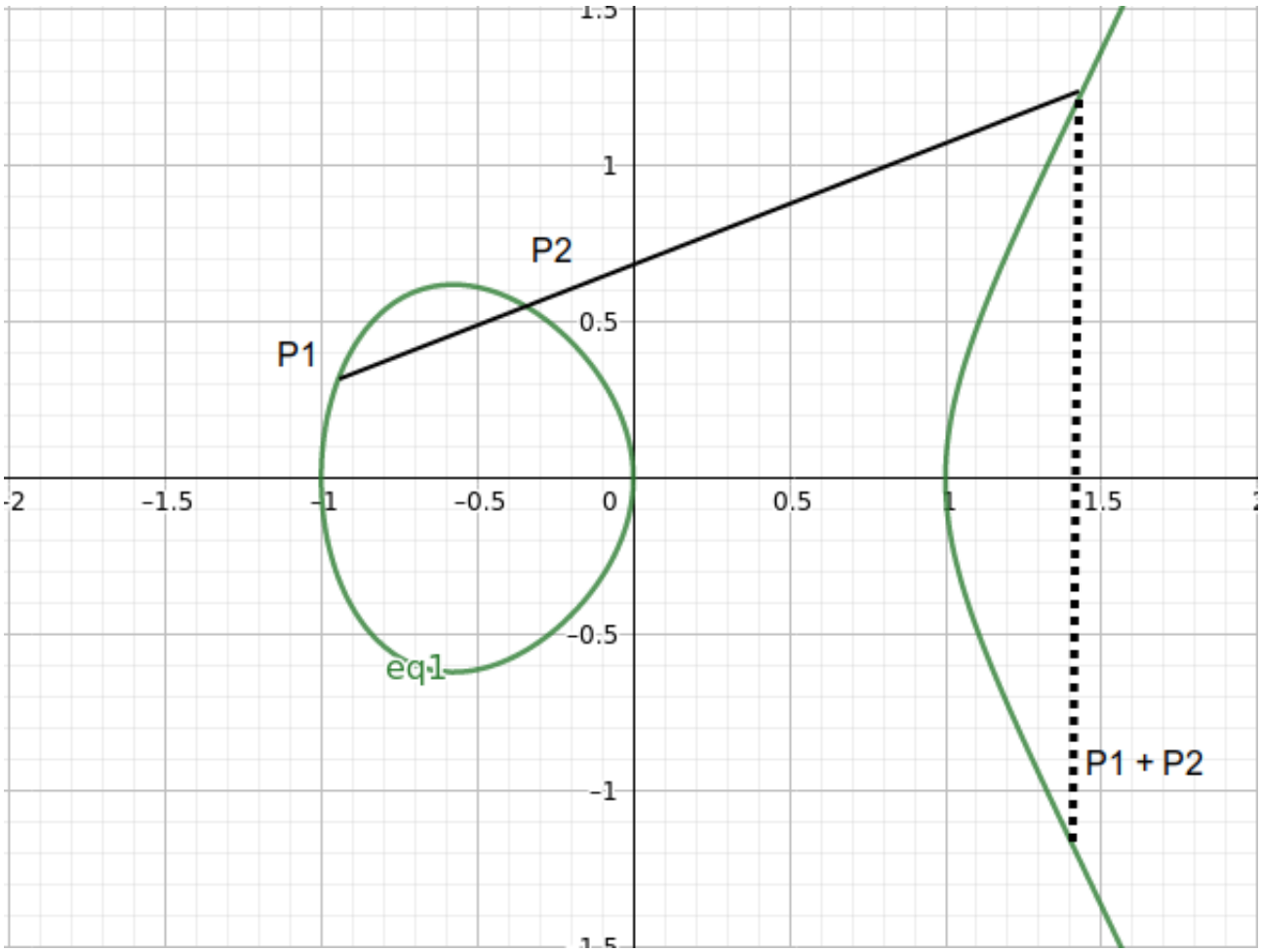

Рисунок 1 — Сложение двух точек на эллиптической кривой

В практической части реализована программа сложения точек на ээллиптической кривой. Для начала определяется класс, который описывает эту кривую:

```
class Curve:
  def __init__(self, a, b, c):
    self.a = a
    self.b = b
    self.c = c
```

Для того, чтобы в дальнейшем можно было сравнивать разные экземпляры класса с помощью удобного оператора ==, переопределим в классе *Curve* этот оператор [9]:

```
def __eq__(self, other):
  return self.a == other.a and self.b == other.b and self.c == other.c
```

Равенство кривых определяется следующим способом: если все три коэффициента у обоих кривых равны, то они считаются одинаковыми, в противном случае – разными [10].

Также для вывода на экран коэффициентов кривой в удобночитаемом формате стоит переопределить метод $\_\_str\_\_$, который возвращает строковое представление класса [11].

```
def __str__(self):
  return "Curve: a={:.3f}, b={:.3f}, c={:.3f}" \
        .format(self.a, self.b, self.c)
```

Точка тоже задается как класс. При создании экземпляра класса необходимо передать следующие параметры: координата x, координата y и заранее созданный экземпляр класса кривой $Curve$.

```
class Point:
  def __init__(self, x, y, curve):
    self.x = x
    self.y = y
    self.curve = curve
```

Следующим шагом определяется метод is_on_curve, который проверяет условие принадлежания точки кривой, переданной в конструктор. Для этого можно было бы использовать обычное сравнение левой и правой части, но существует вероятность некорректного ответа из-за накопившейся погрешности при расчетах в числах с плавающей точкой [12]. Для того, чтобы обойти это, результаты сравниваются по модулю разности. Этот модуль должен лежать в окрестности малого значения, в данном случае – 0.0001.

```
def is_on_curve(self):
  return abs(self.y ** 2 - (self.x ** 3 + self.curve.a * 3 * self.x**2 +
                                    self.curve.b * self.x + self.curve.c))
                                     < 0.0001
```

Так как точки создаются как экземпляры класса, то на них можно переопределить оператор сложения.

```
def __add__(self, other):
  if self.curve != other.curve:
    raise ValueError("Both points must be on the same curve.")

  if self == other:
```

```
    s = (3 * self.x**2 + 2 * self.curve.a * self.x + self.curve.b) / (2 *
                                        self.y)
  elif self.x == other.x:
    return Point(x=None, y=None, curve=None)
  else:
    s = (other.y - self.y) / (other.x - self.x)

  nu = self.y - s * self.x
  x3 = s ** 2 - self.curve.a - self.x - other.x
  y3 = -(s * x3 + nu)

  return Point(x3, y3, self.curve)
```

Стоит отметить одну важную особенность: при сложении двух симметричных отнсительно оси x точек, то есть таких точек $a = (x_1, y_1)$, $b = (x_2, y_2)$, что $x_1 = x_2$ и $y_1 = -y_2$, должна вернуться особенная точка – нулевой элемент, который является бесконечно удаленной точкой. В этом случае метод вернет точку $Point(x = None, y = None, curve = None)$.

По аналогии с переопределением методов $\_\_str\_\_$ и $\_\_eq\_\_$ в классе $Curve$, то же самое выполняется в классе $Point$:

```
def __eq__(self, other):
  return self.x == other.x and self.y == other.y and self.curve==other.curve

def __str__(self):
  return "({:.3f}, {:.3f}), {}".format(self.x, self.y, self.curve)
```

Теперь классы готовы к использованию. В качестве наглядного примера создается кривая $y^2 = x^3 + 17$ и две точки, которые лежат на ней.

```
curve = Curve(0, 0, 17)
a = Point(-1, 4, curve)
b = Point(2, 5, curve)
```

Их принадлжность кривой можно проверить с помощью метода is_on_curve:

```
print("Is point 'a' on the curve? {}".format(a.is_on_curve()))
print("Is point 'b' on the curve? {}".format(b.is_on_curve()))
```

При запуске в терминал выведутся следующие строки:

```
Is point 'a' on the curve? True
Is point 'b' on the curve? True
```

Метод $\_\_add\_\_$ позволяет производить сложение через обычный знак $+$.

```
c = a + b
```

Переопределенный метод $\_\_str\_\_$ позволяет вывести точку на экран в отформатированном виде с помощью всего лишь одной команды. Также заодно проверяется принадлжность полученной точки кривой:

```
print(c)
print("Is point 'c' on the curve? {}".format(c.is_on_curve()))
```

Выведется следующий результат:

```
(-0.889, -4.037), Curve: a=0.000, b=0.000, c=17.000
Is point 'c' on the curve? True
```

Два основных вида шифрования – это симметричное и асимметричное шифрование. Асимметричное шифрование также известно как шифрование с открытым ключом [13]. При симметричном шифровании существует только один ключ и все общающиеся стороны используют один и тот же секретный ключ как для шифрования, так и для дешифрования. При асимметричном шифровании, или шифровании с открытым ключом, существует два ключа: один ключ используется для шифрования, а другой – для расшифровки. Ключ расшифровки хранится в тайне, откуда и берется название закрытый ключ, а ключ шифрования находится в открытом доступе и может использоваться любым пользователем – это открытый ключ [14].

Самым популярным алгоритмом асиммитричного шифрования на сегодняшнйи день является алгоритм RSA, названный так по фамилиям его авторов: Rivest, Shamir, Adleman. Криптосистема открытого ключа RSA основана на огромном различии между легкостью нахождения больших простых чисел и сложностью факторизации произведения двух больших простых чисел [15].

Алгоритм RSA выглядит следующим ообразом:

1. Случайным образом выбрать два больших простых числа p и q такие, что $p \neq q$. Числа p и q должны быть, например, по 1024 бита каждый;
2. Вычислить $n = pq$;
3. Выбрать маленькое нечетное число $1 < e < \phi(n)$, которое будет взаимно простым с $\phi(n)$, где $\phi(n) = (p-1)(q-1)$;

4. Вычислить $d \equiv e^{-1}(mod\ \phi(n))$, то есть мультипликативно обратное к числу e по модулю $\phi(n)$;
5. Опубликовать пару $P = (e, n)$ как публичный ключ участника;
6. Сохранить пару $S = (d, n)$ как секретный ключ.

На эллиптических кривых над кольцом вычетов $\mathbb{Z}/n\mathbb{Z}$ для составного числа $n = pq$, где p, q – различные простые числа, можно переложить большинство криптографических протоколов, первоначально разработанных в рамках системы RSA [16]. Эллиптические кривые вида $y^2 \equiv x^3 + B\ (mod\ p)$ при $p \equiv 5\ (mod\ 6)$ и $y^2 \equiv x^3 + Ax\ (mod\ p)$ при $p \equiv 3\ (mod\ 4)$ имеют порядок группы $p+1$. Поэтому эллиптические кривые $E(\mathbb{Z}/n\mathbb{Z})$ вида $y^2 \equiv x^3 + B\ (mod\ n)$ при $p \equiv q \equiv 5\ (mod\ 6)$ и $y^2 \equiv x^3 + Ax\ (mod\ n)$ при $p \equiv q \equiv 3\ (mod\ 4)$ имеют порядок группы $(p+1)(q+1)$. Оригинальная система RSA имеет порядок группы $\phi(n) = (p-1)(q-1)$. Поэтому такие эллиптические кривые позволяют строить криптосистемы, аналогичные RSA [17]. Если безопасность системы RSA связана с вычислением функции $\phi(n)$, то в случае эллиптических кривых она связана с нахождением числа $(p+1)(q+1)$.

Предлагается рассмотреть аналог шифрования с открытым ключом на эллиптической кривой $y^2 \equiv x^3 + B\ (mod\ n)$. Открытым ключом здесь является пара (n, e), секретным – показатель $d \equiv e^{-1}\ (mod(p+1)(q+1))$. Для зашифрования сообщения m отправитель генерирует случайное положительное число $y < n$, вычисляет шифрограмму, умножая точку $(m, y) \in E(\mathbb{Z}/n\mathbb{Z})$ на показатель e и посылает шифрограмму получателю. Получатель умпножает полученную точку на число d, находит точку (m, y) и восстанавливает сообщение m. Стоит отметить, что в этом случае длина шифрограммы вдвое больше длины открытого текста, при этом уравнение эллиптической кривой не фиксировано.

Аналогичный протокол может быть использован в случае эллиптической кривой $y^2 \equiv x^3 + Ax\ (mod\ n)$. Для удвоения точки нужно предварительно вычислить коэффициент A. Это вычисление не требует разложения числа n и может быть выполнено расширенным алгоритмом Евклида [18].

Протокол подписи Эль-Гамаля строится следующим образом. Секретным ключом формирования подписи служит целое число $x,\ 0 < x < r$, открытым ключом проверки подписи служит простое целое число p, образующая a под-

группы простого порядка r мультипликативной группы F_P^* простого поля и экспонента $b \equiv a^x (mod\ p)$.

Для формирования подписи сообщения $m,\ 0 < m < r$, отправитель выполняет следующие действия:

1. Генерирует случайное число $k,\ 0 < k < r$.
2. Полагает $w = d^k (mod\ p)$.
3. Находит число s, решая сравнение $m \equiv xw + ks (mod\ r)$, где $s = (m - sw)k^{-1}\ (mod\ r)$. Подписью для сообщения m является пара (w, s).

Для проверки подписи получатель выполняет следующие действия:

1. Проверяет неравенство $w < p$. Если оно не выполнено, то результат: подпись недействительна.
2. Проверяет сравнение для экспонент: $d^m \equiv b^w w^x\ (mod\ p)$. Если оно выполняется, то результат: подпись подлинная, иначе результат: подпись недействительна.

Поскольку число k, используемое на этапе формирования подписи, взаимно просто с числом r, то k обратимо по модулю r, то есть число s всегда существует и оно единственное [19].

В протоколе подписи Эль-Гамаля открытый ключ проверки подписи содержит уравнение эллиптической кривой $E(K)$, образующую Q простого порядка r и точку $P = lQ$. Секретным ключом формирования подписи является показатель l. Кроме того, в протоколе подписи используется хэш-фнкция h и функция f.

Для подписи сообщения m отправитель вычисляет хэш-функцию $h(m)$, вырабатывает случайное целое число $k,\ 0 < k < r$, вычисляет точку $R = kQ,\ R = (x_R, y_R)$, и находит число s решением сравнения

$$h(m) \equiv lf(R) + ks\ (mod\ r).$$

При этом необходимо, чтобы выполнялось неравенство $f(R) \neq 0\ (mod\ r)$, в противном случае нужно выбрать другое число k, так как если $f(R) \equiv 0\ (mod\ r)$, то подпись не зависит от персонального ключа.

Подписанное таким образом сообщение представляет из себя стройку (m, R, s). При этом можно сократить объем подписи, задав вместо R только координату x_R и знак координаты y_R. Тогда при проверке подписи необходимо будет восстановить координату y_R решением квадратного уравнения в поле K.

При проверке подписи левая и правая часть сравнения используются в качестве показателей [20]. Получатель сначала восстанавливает точку R и убеждается, что она лежит на эллиптической кривой $E(K)$, а затем проверяет равенство

$$h(m)Q = f(R)P + sR.$$

Как и в оригинальном протоколе Эль-Гамаля, использование показателя k дважды или однократное использование известного показателя k приводит ко вскрытию ключа l. В качестве функции $f(R)$ могут использоваться функции от координат: $x_R, y_R, x_R + y_R$ и тому подобные.

Заключение

В данной работе были представлены теоретические основы эллиптических кривых. Были предоставлены доводы в пользу того, почему эта теория является актуальной. Также были перечислены алгоритмы и системы на эллиптических кривых, которые обширно применяются на практике.

Практическая часть реализована на языке Python. Для представления эллиптической кривой и точки была выбрана объектно-ориентированная парадигма. Она позволила инкапсулировать логику операций в классах и обращаться с экземплярами этого классса как с обычными переменными. Для удобного вывода точек и кривой в программе был применен механизм перегрузки операторов и переопределения методов. Также в написанной программе были учтены и протестированы краевые случаи.

Были рассмотрены и описаны основные виды шифрования. Особое внимание уделилось асимметричному шифрованию и алгоритмам, которые его используют: RSA и система Эль-Гамаля. Приведены аналоги этих алгоритмов в теории эллиптических кривых, а также при сравнении приведены преимущества и недостатки этих алгоритмов перед друг другом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лидл, Р. Конечные поля / Р. Лидл, Г. Нидеррайтер – М. : Мир, 1988. – 430 с.

2. Коблиц, Н. Курс теории чисел и криптографии / Н. Коблиц – М. : Мир, 2001. – 254 с.

3. Василенко, О. Теоретико-числовые алгоритмы в криптографии / О. Коблиц – М. : МЦНМО, 2003. – 328 с.

4. Ростовцев, А.Г. Теоретическая криптография / А.Г. Ростовцев, Е.Б. Маховенко – Санкт-Петербург. : Профессионал, 2004. – 465 с.

5. Жданов, О. Эллиптические кривые. Основы теории и криптографические приложения / О. Жданов – М. : Либроком, 2012. – 421 с.

6. Болотов, А. Элементарное введение в эллиптическую криптографию. Книга 2. / А. Болотов, С. Гашков, А. Фролов – Ленанд, 2020. – 376 с.

7. Кнэпп, Э. Эллиптические кривые / Э. Кнэпп – Факториал Пресс, 2004. – 488 с.

8. Коблиц, Н. Введение в эллиптические кривые и модулярные формы / Н. Коблиц – М. : Мир, 1988. – 320 с.

9. Python Official Website [Электронный ресурс] : [сайт]. – URL: `https://www.python.org/` (дата обращения: 22.01.2021). – Загл. с экрана. – Яз. англ.

10. Linux Official Website [Электронный ресурс] : [сайт]. – URL: `https://www.linux.org/` (дата обращения: 27.01.2021). – Загл. с экрана. – Яз. англ.

11. Numpy Python Package [Электронный ресурс] : [сайт]. – URL: `https://numpy.org/` (дата обращения: 4.05.2021). – Загл. с экрана. – Яз. англ.

12. Oracle Documentation [Электронный ресурс] : [сайт]. – URL: `https://docs.oracle.com/` (дата обращения: 5.05.2021). – Загл. с экрана. – Яз. англ.

13. Понасенко, С. Алгоритмы шифрования. Специальный справочник / С. Понасенко – БХВ-Петербург, 2009. – 576 с.

14. Фороузан, Б. Криптография и безопасность сетей / Б. Фороузан – БИНОМ, 2016. – 315 с.

15. Cormen, T. Introduction to algorithms / T. Cormen, C. Leiserson, R. Rivest, C. Stein – The MIT Press, 2009. – 1292 p.

16. Гатченко, Н. Криптографическая защита информации / Н. Гатченко, А. Исаев, А. Яковлев – СПб : НИУ ИТМО, 2012. – 142 с.

17. Washington, L. Elliptic curves number theory and cryptography / L. Washington – Taylor & Francis Group, 2008. – 513 p.

18. Silverman, J. The arithmetic of elliptic curves / J. Silverman – Springer, 2000. – 513 p.

19. Elgamal, T. Public key cryptosystem and a signature scheme based on discrete logarithms / T. Elgamal – IEEE, 1985. – 4 p.

20. Rahman, R. Elliptic curves, the group law, and the J invariant / R. Rahman – Springer, 2007. – 14 p.