Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы защиты информации

ОТЧЁТ

к лабораторной работе №7

на тему

на тему «Реализация схемы шифрования (дешифрования) для аналога алгоритма Эль-Гамаля на основе эллиптических кривых»

Выполнил: Е.А. Киселева

Проверил: А. В. Герчик

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_Toc181565713)

[2 Краткие теоретические сведения 4](#_Toc181565714)

[3 Результаты выполнения лабораторной работы 7](#_Toc181565715)

[Выводы 8](#_Toc181565716)

Список использованных источников.....................................................................9

[Приложение А (обязательное) Листинг программного кода 10](#_Toc181565717)

# **ПОСТАНОВКА ЗАДАЧИ**

Целью выполнения данной лабораторной работы является реализация схемы шифрования (дешифрования) для аналога алгоритма Эль-Гамаля на основе эллиптических кривых.

# **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ** **СВЕДЕНИЯ**

Преимущество подхода на основе эллиптических кривых заключается в том, что в данном случае обеспечивается эквивалентная защита при меньшей длине ключа.

В общем случае уравнение эллиптической кривой   Е имеет вид:

y2 + axy + by = x3 + cx2 + dx + e

В качестве примера рассмотрим эллиптическую кривую   Е, уравнение которой имеет вид:

y2 + y = x3 - x2

На этой кривой лежат только четыре точки, координаты которых являются целыми числами. Это точки А(0, 0), В(1, -1), С(1, 0) и D(0, -1).

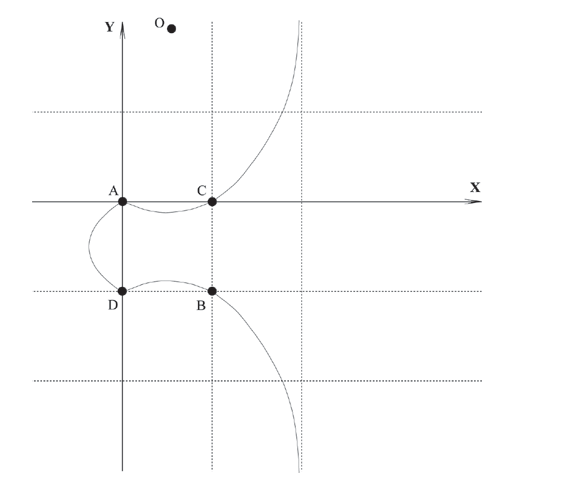


Рисунок 2.1 – Пример эллиптической кривой с четырьмя точками

Для определения операции сложения для точек на эллиптической кривой сделаем следующие предположения:

1 На плоскости существует бесконечно удаленная точка 0Е, в которой сходятся все вертикальные прямые.

2 Будем считать, что касательная к кривой пересекает точку касания два раза.

3 Если три точки эллиптической кривой лежат на прямой линии, то их сумма есть 0.

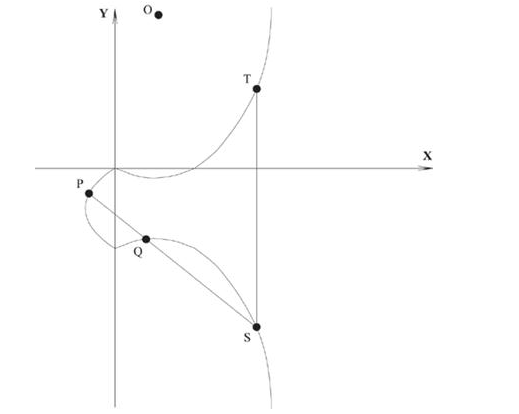


Рисунок 2.2 – Сложение точек на эллиптической кривой

Введем следующие правила сложения точек на эллиптической кривой:

1 Точка 0 выступает в роли нулевого элемента. Так, 0 = -0 и для любой точки Р на эллиптической кривой Р + 0 = Р.

2 Вертикальная линия пересекает кривую в двух точках с одной и той же координатой х - скажем, S = (x, y) и T = (x, -y). Эта прямая пересекает кривую и в бесконечно удаленной точке. Поэтому Р1 + Р2 + 0 = 0 и Р1 = -Р2.

3 Чтобы сложить две точки P и Q (см. рисунок 11.2) с разными координатами х, следует провести через эти точки прямую и найти точку пересечения ее с эллиптической кривой. Если прямая не является касательной к кривой в точках P или Q, то существует только одна такая точка, обозначим ее S. Согласно нашему предположению P + Q + S = О. Следовательно, P + Q = -S или P + Q = T.

Если прямая является касательной к кривой в какой-либо из точек P или Q, то в этом случае следует положить S = P или S = Q соответственно.

Чтобы удвоить точку Q, следует провести касательную в точке Q и найти другую точку пересечения S с эллиптической кривой. Тогда Q + Q = 2 × Q = -S.

Введенная таким образом операция сложения подчиняется всем обычным правилам сложения, в частности коммутативному и ассоциативному законам. Умножение точки Р эллиптической кривой на положительное число k определяется как сумма k точек Р.

В криптографии с использованием эллиптических кривых все значения вычисляются по модулю р, где р является простым числом. Элементами данной эллиптической кривой являются пары неотрицательных целых чисел, которые меньше р и удовлетворяют частному виду эллиптической кривой:

y2 ≡x3 + ax + b (mod p)

Такую кривую будем обозначать Ep (a,b). При этом числа а и b должны быть меньше р и должны удовлетворять условию 4a3 + 27b2 (mod p) ≠ 0. Множество точек на эллиптической кривой вычисляется следующим образом.

Для каждого такого значения х, что 0≤х≤р, вычисляется x3 + ax + b (mod p).

Для каждого из полученных на предыдущем шаге значений выясняется, имеет ли это значение квадратный корень по модулю р. Если нет, то в Ep (a,b) нет точек с этим значением х. Если корень существует, имеется два значения y, соответствующих операции извлечения квадратного корня (исключением является случай, когда единственным значением оказывается y = 0). Эти значения (x,y) и будут точками Ep (a,b).

Множество точек Ep (a,b) обладает следующими свойствами:

1.           Р + 0 = Р.

2.           Если Р = (x,y), то Р + (x,-y) = 0. Точка (x,-y) является отрицательным значением точки Р и обозначается -Р. Заметим, что (x,-y) лежит на эллиптической кривой и принадлежит Ep (a,b).

3.           Если Р = (x1,y1) и Q = (x2,y2), где P ≠ Q, то P + Q = (x3,y3) определяется по следующим формулам:

x3≡  λ2 - x1 - x2 (mod p)

y3 ≡ λ (x1 - x3) - y1 (mod p)

где (y2 - y1)/(x2 - x1) , если P ≠ Q,  λ = (3x12 + a)/2y1 , если P = Q

Число λ есть угловой коэффициент секущей, проведенной через точки P = (x1, y1) и Q = (x2, y2). При P = Q секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ.

Задача, которую должен решить в этом случае атакующий, есть своего рода задача **"дискретного логарифмирования на эллиптической кривой"**, и формулируется она следующим образом. Даны точки P и Q на эллиптической кривой   Ep (a,b). Необходимо найти коэффициент k < p такой, что

P = k × Q.

Относительно легко вычислить P по данным k и Q, но довольно трудно вычислить k, зная P и Q.

**3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ**

В ходе выполнения лабораторной была реализована схема шифрования (дешифрования) для аналога алгоритма Эль-Гамаля на основе эллиптических кривых.

Начальный текст находится в файле input.txt. Программа считывает необходимую информацию, а именно значение начальной точки P, такие как x, y, a, b, p. При шифровании проводится расчет двух точек кривой C1 и C2. После некоторого преобразования над этими точками и начальной точкой P выводится расшифрованное сообщение.

Результат выполнения лабораторной работы представлен на рисунке 3.1.

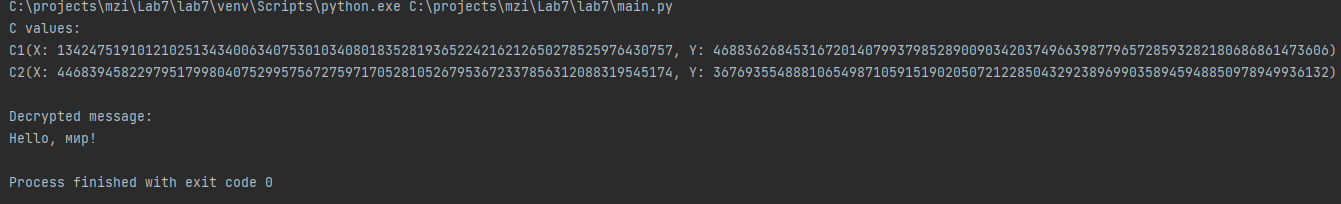


Рисунок 3.1 – Результат выполнения лабораторной работы

Таким образом результатом лабораторной работы является реализованная схема шифрования (дешифрования) для аналога алгоритма Эль-Гамаля на основе эллиптических кривых.

# **ВЫВОДЫ**

В ходе данной лабораторной работы была разработана схема шифрования (дешифрования) для аналога алгоритма Эль-Гамаля на основе эллиптических кривых.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Схема шифрования Эль Гамаля [Электронный ресурс]. – Режим доступа: https://crypto-r.narod.ru/glava4/glava4\_5.html/. – Дата доступа: 26.10.2024.

[2] Эллиптические кривые [Электронный ресурс]. – Режим доступа: https://homepage.mi-ras.ru/. – Дата доступа: 27.10.2024.

## **ПРИЛОЖЕНИЕ А**

## **(обязательное)**

## **Листинг программного кода**

Листинг 1 – Программный код файла main.py

from EllipticCurvePoint import EllipticCurvePoint

from ElGamal import ElGamal

with open("./input.txt", "rb") as f:

message = f.read()

# Инициализация эллиптической кривой и точек

P = EllipticCurvePoint(

x=2,

y=4018974056539037503335449422937059775635739389905545080690979365213431566280,

a=90,

b=43308876546767276905765904595650931995942111794451039583252968842033849580414,

p=57896044618658097711785492504343953926634992332820282019728792003956564821041

)

d = 47296044618658097711785492524343953912234992332820282019728792003956564821041

Q = P.multiply(d)

#Шифровка, вывод С1, С2 и соотв У

CValues = ElGamal.encrypt(message, P, Q)

print(f"C values:\nC1(X: {CValues[0].x}, Y: {CValues[0].y})\nC2(X: {CValues[1].x}, Y: {CValues[1].y})\n")

# Дешифровка и вывод сообщения

decrypted\_message = ElGamal.decrypt(CValues, d)

print(f"Decrypted message:\n{decrypted\_message.decode('utf-8')}")

Листинг 2 – Программный код файла ElGamal.py

from EllipticCurvePoint import EllipticCurvePoint

from Crypto.Random import random

from Crypto.Util.number import long\_to\_bytes, bytes\_to\_long

class ElGamal:

@staticmethod

def generate\_random\_big\_integer(N):

# Генерация случайного числа, меньшего N

bytes\_len = (N.bit\_length() + 7) // 8

while True:

r = random.getrandbits(bytes\_len \* 8) % N

if r < N:

return r

@staticmethod

def get\_point\_from\_bytes(message\_bytes, P):

# Преобразование байтового сообщения в точку эллиптической кривой

p\_length = (P.p.bit\_length() + 7) // 8

if len(message\_bytes) >= p\_length - 2:

raise Exception(f"M({len(message\_bytes)}) should be less than p (Max M Length = {p\_length - 2} symbols)")

# Дополнение сообщения байтами для преобразования в координату x

message = message\_bytes + bytes([0xff]) + b'\x00' \* (p\_length - len(message\_bytes) - 1)

return EllipticCurvePoint(

x=bytes\_to\_long(message),

y=0,

a=P.a,

b=P.b,

p=P.p

)

@staticmethod

def get\_bytes\_from\_point(P):

# Извлечение сообщения из координаты x точки эллиптической кривой

message\_bytes = long\_to\_bytes(P.x)

if 0xff in message\_bytes:

return message\_bytes[:message\_bytes.index(0xff)]

return message\_bytes

@staticmethod

def encrypt(message\_bytes, P, Q):

M = ElGamal.get\_point\_from\_bytes(message\_bytes, P)

k = ElGamal.generate\_random\_big\_integer(P.p)

C1 = P.multiply(k)

C2 = M + Q.multiply(k)

return C1, C2

@staticmethod

def decrypt(CValues, d):

temp = CValues[0].multiply(d)

temp.y = -temp.y % temp.p

P = temp + CValues[1]

return ElGamal.get\_bytes\_from\_point(P)

Листинг 3 – Программный код файла EllipticCurvePoint.py

class EllipticCurvePoint:

def \_\_init\_\_(self, x, y, a, b, p):

self.x = x

self.y = y

self.a = a

self.b = b

self.p = p

def \_\_add\_\_(self, other):

# Операция сложения точек

if self.x == other.x and self.y == other.y:

return self.double()

dy = (other.y - self.y) % self.p

dx = (other.x - self.x) % self.p

m = (dy \* pow(dx, -1, self.p)) % self.p

x3 = (m \* m - self.x - other.x) % self.p

y3 = (m \* (self.x - x3) - self.y) % self.p

return EllipticCurvePoint(x3, y3, self.a, self.b, self.p)

def double(self):

# Удвоение точки

dy = (3 \* self.x \* self.x + self.a) % self.p

dx = (2 \* self.y) % self.p

m = (dy \* pow(dx, -1, self.p)) % self.p

x2 = (m \* m - 2 \* self.x) % self.p

y2 = (m \* (self.x - x2) - self.y) % self.p

return EllipticCurvePoint(x2, y2, self.a, self.b, self.p)

def multiply(self, k):

# Умножение точки на скаляр

result = None

addend = self

while k:

if k & 1:

result = addend if result is None else result + addend

addend = addend.double()

k >>= 1

return result

def \_\_str\_\_(self):

return f"({self.x}, {self.y})"

def \_\_eq\_\_(self, other):

return self.x == other.x and self.y == other.y and self.a == other.a and self.b == other.b and self.p == other.p