Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

ОТЧЁТ

по лабораторной работе №7

По дисциплине «Методы защиты информации»

Выполнил:

Студент гр. 853505

Лазарева Е.В.

Проверил:

Олисейчик В.В.

Минск 2021

**1. Постановка задачи**

Необходимо реализовать программное средство формирования электронной цифровой подписи на основе эллиптических кривых ECDSA и программное средство, реализующее простой подход к шифрованию/дешифрованию с использованием эллиптических кривых*.*

**ЗАДАНИЕ:**

1. Изучить теоретические сведения.
2. Создать программы, читающие данные из файла и формирующие в качестве выхода электронную цифровую подпись на основе эллиптических кривых ECDSA и программное средство, реализующее простой подход к шифрованию/дешифрованию с использованием эллиптических кривых.
3. Реализовать проверку ЭЦП на основе эллиптических кривых ECDSA.

**2. Теоретическая** **сведения**

**Математические понятия**

Преимущество подхода на основе *эллиптических кривых* в сравнении с задачей факторизации числа, используемой в RSA, или задачей целочисленного логарифмирования, применяемой в алгоритме Диффи-Хеллмана и в DSS, 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)

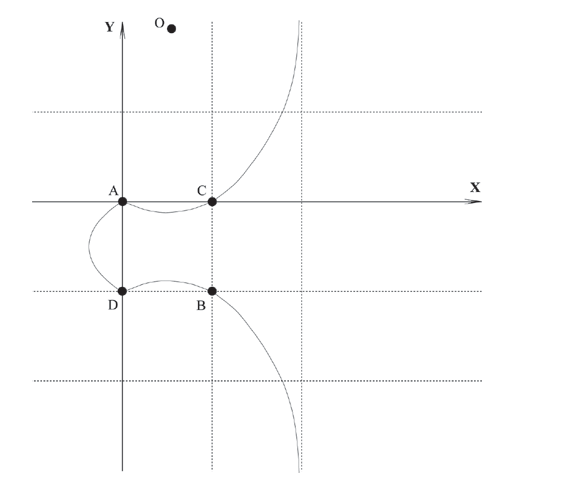


Рисунок 1 – Пример эллиптической кривой с четырьмя точками

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

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

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

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

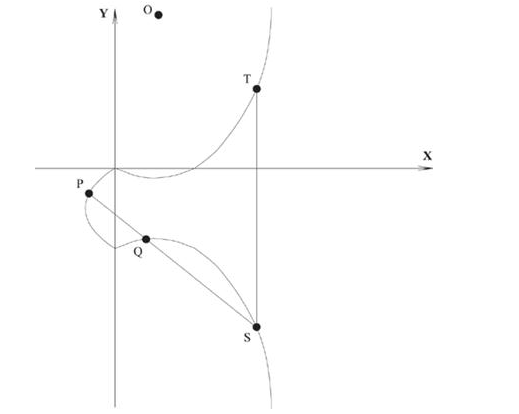


Рисунок 2 – Сложение точек на эллиптической кривой

Введем следующие правила сложения точек на *эллиптической кривой*:

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

                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.

                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:

4.           x3≡  λ2 - x1 - x2 (mod p)

5.           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.

**Аналог алгоритма Диффи-Хеллмана обмена ключами**

Обмен ключами с использованием *эллиптических кривых* может быть выполнен следующим образом. Сначала выбирается простое число р ≈ 2180 и параметры a и b для уравнения *эллиптической кривой*. Это задает множество точек Ep (a,b). Затем в Ep (a,b) выбирается генерирующая точка G = (x1,y1). При выборе G важно, чтобы наименьшее значение n, при котором n × G = 0, оказалось очень большим простым числом. Параметры Ep (a,b) и G криптосистемы являются параметрами, известными всем участникам.

Обмен ключами между пользователями А и В производится по следующей схеме.

1.           Участник А выбирает целое число nA, меньшее n. Это число является закрытым ключом участника А. Затем участник А вычисляет открытый ключ PA = nA × G, который представляет собой некоторую точку на Ep (a,b).

2.           Точно так же участник В выбирает закрытый ключ nB и вычисляет открытый ключ PB.

3.           Участники обмениваются открытыми ключами, после чего вычисляют общий секретный ключ K

Участник А: K = nA × PB

       Участник В: K = nВ × PА

Следует заметить, что общий секретный ключ представляет собой пару чисел. Если данный ключ предполагается использовать в качестве сеансового ключа для алгоритма симметричного шифрования, то из этой пары необходимо создать одно значение.

**Алгоритм цифровой подписи на основе эллиптических кривых ECDSA**

Алгоритм ECDSA (Elliptic Curve Digest Signature Algorithm) принят в качестве стандартов ANSI X9F1 и IEEE P1363.

Создание ключей:

1.           Выбирается *эллиптическая кривая*   Ep (a,b). Число точек на ней должно делиться на большое целое n.

2.           Выбирается точка РEp (a,b).

3.           Выбирается случайное число d  [1, n-1].

4.           Вычисляется Q = d × P.

5.           Закрытым ключом является d, открытым ключом – (E, P, n, Q).

Создание подписи:

1.           Выбирается случайное число k [1, n-1].

2.           Вычисляется k × P = (x1, y1)   и  r = x1 (mod n).

Проверяется, чтобы r не было равно нулю, так как в этом случае подпись не будет зависеть от закрытого ключа. Если r = 0, то выбирается другое случайное число k.

3.           Вычисляется k-1 mod n

4.           Вычисляется s = k-1 (Н(M) + dr) (mod n)

Проверяется, чтобы s не было равно нулю, так как в этом случае необходимого для проверки подписи числа s-1 mod n не существует. Если s = 0, то выбирается другое случайное число k.

Подписью для сообщения М является пара чисел (r,s).

Проверка подписи:

1.           Проверить, что целые числа r и s принадлежат диапазону чисел [0, n-1]. В противном случае результат проверки отрицательный, и подпись отвергается.

2.           Вычислить w = s-1 (mod n) и H(M)

3.           Вычислить u1 = H(M) w (mod n), u2 = rw (mod n)

4.           Вычислить  u1P + u2Q = (x0, y0), v = x0 (mod n)

5.           Подпись верна в том и только том случае, когда v = r.

**Шифрование/дешифрование с использованием эллиптических кривых**

Рассмотрим самый простой подход к шифрованию/дешифрованию с использованием *эллиптических кривых*. Задача состоит в том, чтобы зашифровать сообщение М, которое может быть представлено в виде точки на эллиптической кривой Pm (x,y).

Как и в случае обмена ключом, в системе шифрования/дешифрования в качестве параметров рассматривается *эллиптическая кривая*   Ep (a,b) и точка G на ней. Участник B выбирает закрытый ключ nB и вычисляет открытый ключ PB = nB × G. Чтобы зашифровать сообщение Pm используется открытый ключ получателя B   PB. Участник А выбирает случайное целое положительное число k и вычисляет зашифрованное сообщение Cm, являющееся точкой на *эллиптической кривой*.

Cm = {k × G, Pm + k × PB}

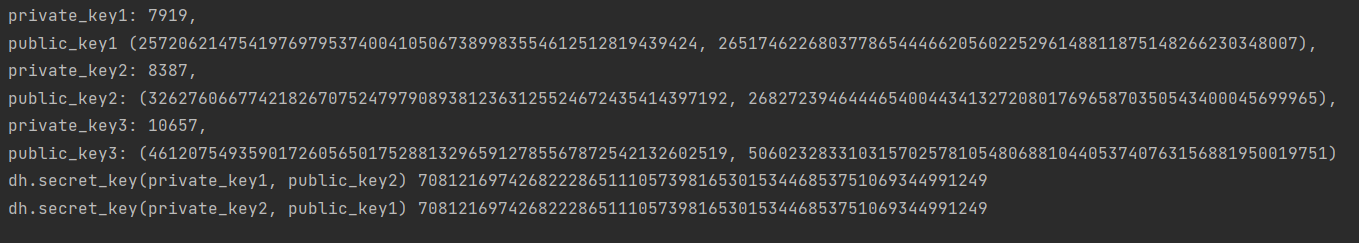
Чтобы дешифровать сообщение, участник В умножает первую координату точки на свой закрытый ключ и вычитает результат из второй координаты:

Pm + k × PB - nB × (k × G) = Pm + k × (nB × G) - nB × (k × G) = Pm

Участник А зашифровал сообщение Pm добавлением к нему kxPB. Никто не знает значения k, поэтому, хотя PB и является открытым ключом, никто не знает k × PB. Противнику для восстановления сообщения придется вычислить k, зная G и k × G. Сделать это будет нелегко.

Получатель также не знает k, но ему в качестве подсказки посылается k × G. Умножив k × G на свой закрытый ключ, получатель получит значение, которое было добавлено отправителем к незашифрованному сообщению. Тем самым получатель, не зная k, но имея свой закрытый ключ, может восстановить незашифрованное сообщение.

1. **Пример работы программы**



**4. Код программы**

class ECC(object):  
  
 def \_\_init\_\_(self, a, b, p, base\_point):  
 self.curve = (a,b,p)  
 self.base\_point = base\_point  
 self.double\_base\_point = self.double\_point(base\_point)  
  
 def xgcd(self, b, n):  
 x0, x1, y0, y1 = 1, 0, 0, 1  
 while n != 0:  
 q, b, n = b // n, n, b % n  
 x0, x1 = x1, x0 - q \* x1  
 y0, y1 = y1, y0 - q \* y1  
 return b, x0, y0  
  
 def inverse(self, b):  
 g, x, \_ = self.xgcd(b, self.curve[2])  
 if g == 1:  
 return x % self.curve[2]  
  
 def add\_points(self, p, q):  
 delta = 0  
  
 if p == None or q == None:  
 return p if q == None else q  
 elif p[0] == q[0] and p[1] == q[1]:  
 delta = (3 \* p[0]\*\*2 + self.curve[0]) \* \  
 self.inverse(2 \* p[1]) % self.curve[2]  
 else:  
 delta = (p[1] - q[1]) \* self.inverse((p[0] - q[0])) % self.curve[2]  
  
 x = (delta \* delta - p[0] - q[0]) % self.curve[2]  
 y = (delta \* (p[0] - x) - p[1]) % self.curve[2]  
  
 return (x,y)  
  
 def double\_point(self, p, k = 1):  
 Q = p  
 for i in range(0,k):  
 Q = self.add\_points(Q,Q)  
 return Q  
  
 def base\_point\_mult(self, k):  
 Q = None  
 for i in [1 if digit == '1' else 0 for digit in bin(k)[2:]]:  
 Q = self.double\_point(Q)  
 if i == 1:  
 Q = self.add\_points(Q, self.base\_point)  
  
 return Q  
  
class DiffieHellman(object):  
  
 def \_\_init\_\_(self, elliptic\_curve, point\_g):  
 self.elliptic\_curve = elliptic\_curve  
 self.point\_g = point\_g  
  
 def generate\_public\_key(self, private\_key):  
 if private\_key < 0:  
 raise ValueError  
 return self.elliptic\_curve.double\_point(self.point\_g, private\_key)  
  
 def secret\_key(self, private\_key, public\_key):  
 return self.elliptic\_curve.double\_point(public\_key, private\_key)[0]  
  
point\_g = (  
 0x188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012, # Gx  
 0x07192b95ffc8da78631011ed6b24cdd573f977a11e794811) # Gy  
  
curve = ECC(  
 -3, # a  
 0x64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1, # b  
 6277101735386680763835789423207666416083908700390324961279, # p  
 point\_g)  
  
dh = DiffieHellman(curve, point\_g)  
  
private\_key1 = 7919  
public\_key1 = dh.generate\_public\_key(private\_key1)  
  
private\_key2 = 8387  
public\_key2 = dh.generate\_public\_key(private\_key2)  
  
private\_key3 = 10657  
public\_key3 = dh.generate\_public\_key(private\_key3)  
  
print("private\_key1: {}, \npublic\_key1 {}, \nprivate\_key2: {}, \npublic\_key2: {}, \nprivate\_key3: {}, \npublic\_key3: {}"  
 .format(private\_key1, public\_key1, private\_key2, public\_key2, private\_key3, public\_key3))  
print("dh.secret\_key(private\_key1, public\_key2) {}".format(dh.secret\_key(private\_key1, public\_key2)))  
print("dh.secret\_key(private\_key2, public\_key1) {}".format(dh.secret\_key(private\_key2, public\_key1)))

**5. Вывод**

В 1985 году независимо Нилом Коблицем и Виктором Миллером было предложено использовать в криптографии алгебраические свойства эллиптических кривых. С этого момента началось бурное развитие нового направления криптографии, для которого используется термин «криптография на эллиптических кривых». Роль основной криптографической операции выполняет операция скалярного умножения точки на эллиптической кривой на данное целое число, определяемая через операции сложения и удвоения точек эллиптической кривой. Последние, в свою очередь, выполняются на основе операций сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая. Особый интерес к криптографии эллиптических кривых обусловлен теми преимуществами, которые дает её применение в беспроводных коммуникациях — высокое быстродействие и небольшая длина ключа.