```
In[1]:= (*Лабораторная работа №2*)
(*По курсу «Защита информационных процессов в компьютерных системах»*)
(*
    Использование группы точек эллиптической кривой в протоколе
    Диффи-Хеллмана.
*)

(*
    Кутузов Илья
      А-12м-20
*)
```

```
In[2]:= (*1. Для кривой y^2==x^3+100x^2+10x+1 в поле GF(p),
проверить,является ли заданная кривая гладкой,
и определить число точек,принадлежащих этой кривой.*)a = 100;
b = 10;
c = 1;
nvar = 10; (*Номер варианта*)
p = 1061;
point = {120, 1015};
```

```
In[7]:= Mod[point[[1]]^3 + a * point[[1]]^2 + b * point[[1]] + c, p] ≠ 0
        остаток от деления
-16 * (4 * a^3 + 27 * b^2) ≠ 0

Out[7]= True

Out[8]= True
```

```
In[9]:= pointsList = Flatten[Table[FindInstance[
                     уплотнить табл··· найти частный случай
      y^2 == x^3 + a * x^2 + b * x + c && x == u, {x, y}, 2, Modulus → p], {u, 0, p - 1}], 1];
                                                             модуль
cnt = Length[pointsList]
      длина

Out[10]= 1059
```

```
In[11]:= (*Разложить полученное число точек на сомножители,
используя функцию FactorInteger[].Сравнить полученный результат с
                  факторизовать целое число
  проверкой на принадлежность к множеству простых чисел-PrimeQ[].*)
                                                         простое число?
FactorInteger[cnt]
факторизовать целое число
PrimeQ[cnt]
простое число?

Out[11]= {{3, 1}, {353, 1}}

Out[12]= False
```

```
In[13]:= (*3. Определить порядок точки для соответствующего варианта.*)
```

In[14]:=
```
EllipticAdd[p_, a_, b_, c_, P_List, Q_List] := Module[{lam, x3, y3, P3},
                                               [программный модуль
  Which[
  [условный оператор с множественными ветвями
   P == {O}, Q,
         [O большое
   Q == {O}, P,
         [O большое
   P[[1]] ≠ Q[[1]],
       lam = Mod[(Q[[2]] - P[[2]]) PowerMod[Q[[1]] - P[[1]], p - 2, p], p];
             [остаток от деления    [степень по модулю
       x3 = Mod[lam^2 - a - P[[1]] - Q[[1]], p];
            [остаток от деления
       y3 = Mod[- (lam (x3 - P[[1]]) + P[[2]]), p];
            [остаток от деления
       {x3, y3},
   (P == Q) ∧ (P[[2]] == 0), {O},
                              [O большое
   (P == Q) ∧ (P ≠ {O}),
                    [O большое
       lam = Mod[(3 * P[[1]]^2 + 2 a * P[[1]] + b) PowerMod[2 P[[2]], p - 2, p], p];
             [остаток от деления                 [степень по модулю
       x3 = Mod[lam^2 - a - P[[1]] - Q[[1]], p];
            [остаток от деления
       y3 = Mod[- (lam (x3 - P[[1]]) + P[[2]]), p];
            [остаток от деления
       {x3, y3},
   (P[[1]] == Q[[1]]) ∧ (P[[2]] ≠ Q[[2]]), {O}
                                           [O большое
  ]
 ]
```

In[15]:=
```
s = {};
i = 1;
q1 = point;
q2 = q1;
While[q2 ≠ {O}, q2 = EllipticAdd[p, a, b, c, q1, q2]; i++];
[цикл-пока  [O большое
pointOrder = i
```

Out[20]= 1060

In[21]:=
```
(*4. Представить порядок точки в двоичной форме*)
IntegerDigits[pointOrder, 2]
[цифры целого числа
```

Out[21]= {1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0}

In[22]:=
```
(*5. Проверить правильность определения порядка точки путем
 умножения с использованием метода аддитивных цепочек.Пример расчета для
 GF(863) и точки {121,517},порядок которой равен 432 приведен ниже:*)
```

In[23]:=
```
P =.;
P[0] = point;
P[i_] := P[i] = EllipticAdd[p, a, b, c, P[i - 1], P[i - 1]];
Q = EllipticAdd[p, a, b, c, EllipticAdd[p, a, b, c, P[10], P[5]], P[2]]
```

Out[26]= {O}

In[27]:=
```
(*7. Выбрать два "секретных" целых числа a
 и b (секретные ключи для стороны A,и для стороны B),
которые должны быть приблизительно равны 1/3 и 2/3 от величины порядка точки.
*)

delt = 10;
aKey = 356
PercentForm[N[Abs[Round[pointOrder / 3] - aKey] / pointOrder]] (*Разница*)
⌊··          ⌊·· ⌊аб··· ⌊округлить
bKey = 703
PercentForm[N[Abs[Round[pointOrder / 3 * 2] - bKey] / pointOrder]]
⌊форма проце··· ⌊·· ⌊аб··· ⌊округлить
(*Разница*) (*Разница*)
```

Out[28]= 356

Out[29]//PercentForm=
0.283%

Out[30]= 703

Out[31]//PercentForm=
0.3774%

In[32]:=
```
(*8. Найти открытые ключи,вычислив произведения QA=
 a*P и QB=b*P и определить порядки вновь полученных точек.*)
```

In[33]:=
```
IntegerDigits[aKey, 2]
⌊цифры целого числа
IntegerDigits[bKey, 2]
⌊цифры целого числа
```

Out[33]= {1, 0, 1, 1, 0, 0, 1, 0, 0}

Out[34]= {1, 0, 1, 0, 1, 1, 1, 1, 1, 1}

In[35]:=
```
QA = EllipticAdd[p, a, b, c,
  EllipticAdd[p, a, b, c, P[8], P[6]], EllipticAdd[p, a, b, c, P[5], P[2]]]
QB = EllipticAdd[p, a, b, c,
  EllipticAdd[p, a, b, c,
   EllipticAdd[p, a, b, c, P[9], P[7]],
   EllipticAdd[p, a, b, c, P[5], P[4]]],
  EllipticAdd[p, a, b, c,
   EllipticAdd[p, a, b, c, P[3], P[2]],
   EllipticAdd[p, a, b, c, P[1], P[0]]
  ]
 ]
```

Out[35]= {281, 613}

Out[36]= {440, 875}

In[37]:=
```
s = {};
i = 1;
q1 = QA;
q2 = q1;
While[q2 ≠ {O}, q2 = EllipticAdd[p, a, b, c, q1, q2];
ц⋯             О большое
 i++]; i
s = {};
i = 1;
q1 = QB;
q2 = q1;
While[q2 ≠ {O}, q2 = EllipticAdd[p, a, b, c, q1, q2];
цикл–пока      О большое
 i++]; i
```

Out[37]= 265

Out[38]= 1060

In[39]:=
```
(*9. Найти общие ключи KAB=a*QB и KBA=b*QA.*)
IntegerDigits[265, 2]
цифры целого числа
IntegerDigits[1060, 2]
цифры целого числа
```

Out[39]= {1, 0, 0, 0, 0, 1, 0, 0, 1}

Out[40]= {1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0}

```
In[41]:= QA =.; QB =.;
QA[0] = {281, 613};
QB[0] = {440, 875};
QB[i_] := QB[i] = EllipticAdd[p, a, b, c, QB[i - 1], QB[i - 1]];
QA[i_] := QA[i] = EllipticAdd[p, a, b, c, QA[i - 1], QA[i - 1]];

EllipticAdd[p, a, b, c, EllipticAdd[p, a, b, c, EllipticAdd[p, a, b, c, QA[9], QA[7]],
  EllipticAdd[p, a, b, c, QA[5], QA[4]]], EllipticAdd[p, a, b, c,
  EllipticAdd[p, a, b, c, QA[3], QA[2]], EllipticAdd[p, a, b, c, QA[1], QA[0]]]]

EllipticAdd[p, a, b, c, EllipticAdd[p, a, b, c, QB[8], QB[6]],
 EllipticAdd[p, a, b, c, QB[5], QB[2]]]

Out[46]= {408, 43}

Out[47]= {408, 43}

In[48]:= (*10. Провести проверку полученных результатов,
используя модуль прямого умножения точки эллиптической кривой на число.*)
Mult[p1_, a1_, b1_, c1_, t_, n1_] :=
  Module[{p = p1, a = a1, b = b1, e = t, c = c1, n = n1}, q = e;
  программный модуль
   Do[{q1 = EllipticAdd[p, a, b, c, e, q], q = q1}, {i, 2, n}]; q];
   оператор цикла
Mult[p, a, b, c, point, aKey * bKey]

Out[49]= {408, 43}

In[50]:= (*11. Найти точку,принадлежащую заданной эллиптической кривой в поле GF(p),
порядок которой является простым числом.В
  случае отсутствия такой точки для кривой y2==x3+100x2+10x+1,
перейти к кривой вида:y2==x3+100x2+10x+1+i,где i=1,2,3 ....,
т.е.параметр i увеличивается на 1,пока точка,у которой порядок-простое число,
не будет найдена.Обязательно выполнить проверку кривой на «гладкость»!

*)

In[51]:=
```

In[52]:=

```
a = 100; b = 10; c = 1; p = 1061; MinX = 0; MinRank = 0; Xmin = 0;
found = False; x1 = 0; y1 = 0; i = 0;
              ложь
While[found == False,
цикл–пока      ложь
  If[Mod[4 * b^3 + 27 * (c + i)^2, p] ≠ 0,
  у⋯ остаток от деления
   x1 = Xmin;
   While[(x1 < p) && (found == False),
   цикл–пока                  ложь
    If[Solve[y^2 == x1^3 + a * x1^2 + b * x1 + c + i, {y}, Modulus → p] ≠ {},
       решить уравнения                                 модуль
      y1 = y /. Flatten[Solve[y^2 == x1^3 + a * x1^2 + b * x1 + c + i, {y}, Modulus → p], 1];
                уплостить решить уравнения                                          модуль
      p1 = {x1, y1};
      p2 = {x1, y1};
      rank = 1;
      While[p2 ≠ {O},
      цикл–пока   O большое
       p2 = EllipticAdd[p, a, b, c + i, p1, p2];
       rank++;
      ];
      If[rank > MinRank,
      условный оператор
       If[PrimeQ[rank],
       ⋯ простое число?
        Print["Порядок точки ", p1, " = ",
        печатать
         rank, " .Кривая y^2==x^3+", a, "*x^2+", b, "*x+", c, "+", i];
        found = True;,
                истина
        x1++;
       ],
       x1++;
      ];,
      (*Точка не принадлежит кривой.*)
      x1++;
    ];
   ];
  ];
  i++;];
```

Порядок точки {35, 109} = 53 .Кривая $y^2 == x^3 + 100*x^2 + 10*x + 1 + 0$